EMPIRICAL ANALYSIS OF MACHINE LEARNING-BASED ENERGY EFFICIENT CLOUD LOAD BALANCING ARCHITECTURES: A QUANTITATIVE PERSPECTIVE

K.R.Singh
Yeshwantrao Chavan College of Engineering,Nagpur, Maharashtra, (India).
Indira Gandhi Delhi Technical University for Women, New Delhi, Delhi, (India).

A.D.Gaikwad
B Yeshwantrao Chavan College of Engineering,Nagpur, Maharashtra, (India).
Indira Gandhi Delhi Technical University for Women, New Delhi, Delhi, (India).

S.D.Kamble
Yeshwantrao Chavan College of Engineering,Nagpur, Maharashtra, (India).
Indira Gandhi Delhi Technical University for Women, New Delhi, Delhi, (India).

Reception: 20/11/2022 **Acceptance**: 05/12/2022 **Publication**: 29/12/2022

Suggested citation:

Singh, K. R., Gaikwad, A. D., y Kamble, S. D. (2022). Empirical analysis of machine learning-based energy efficient cloud load balancing architectures: a quantitative perspective. *3C Empresa. Investigación y pensamiento crítico, 11*(2), 232-248. https://doi.org/10.17993/3cemp.2022.110250.232-248

ABSTRACT

Design of energy efficient load balancing models in cloud environments requires in-depth analysis of the cloud architecture and nature of requests served by the cloud. Depending upon these parameters, machine learning models are designed which aim at assigning best possible resource combination to serve the given tasks. This assignment varies w.r.t. multiple task and cloud parameters; which include task time, virtual machine (VM) performance, task deadline, energy consumption, etc. In order to perform this task, a wide variety of algorithms are developed by researchers cloud designers. Each of these algorithms aim at optimizing certain load balancing related parameters; for instance, a Genetic Algorithm (GA) designed for optimization of VM utilization might not consider task deadline before task allocation to the VMs. While, algorithms aimed at performing deadline aware load balancing might not provide effective cloud-to-task-mapping before allocation of tasks. Thus, it becomes difficult for researchers to select the best possible algorithms for their cloud deployment. In order to reduce this ambiguity, the underlying text compares different energy efficient cloud load balancing algorithms; and evaluates their performance in terms of computational complexity, and relative energy efficiency. This performance evaluation is further extended via inter architecture comparison; in order to evaluate the most optimum load balancer implementation for a given energy efficient application. Thus, after referring this text, researchers and cloud system designers will be able to select optimum algorithmic implementations for their given deployment. This will assist in reducing cloud deployment delay, and improving application specific load balancer performance.

KEYWORDS

Cloud, Load, Balancing, Machine, Learning, Task, Deadline, Energy.

https://doi.org/10.17993/3cemp.2022.110250.232-248

1. INTRODUCTION

Due to the current CoVID-19 pandemic, most businesses are forced to adopt the work-from home (WFH) model. This model has increased dependency of users on cloud-based services, thereby requiring cloud service providers to optimize their load-balancing models. These models are broadly categorized into 2 types; which are hardware load balancing, and elastic load balancing. The former type consists of optimizing performance of hardware components like virtual machines, servers, memory utilization, task optimization, and central processing unit (CPU) optimization.

While, the later type consists of network load balancers, application load balancers and hybrid load balancers. Hierarchical categorization of these algorithms can be observed from figure 1, from where it can be observed that network load balancing depends on VM and Server load balancing; application load balancing depends on memory, task and CPU load balancing; while classic (or hybrid) load balancing depends on all the hardware-based load balancing models. VM load balancing models aim at optimizing virtual machine performance by assigning tasks in such a manner that most VM resources are utilized, thereby improving hardware utilization efficiency.

This model does not take into consideration deadline constraints, memory constraints, etc. while modelling the load balancer. In contrast, memory-based load balancer models only take into consideration memory utilization; and aim at optimizing task storage without considering resource or CPU load values. Server load balancing algorithms assist in optimization server utilization while performing load balancing, while CPU load balancers aim at optimizing CPU utilization while load balancing. Moreover, task load balancing models aim at executing tasks under a given deadline without considering CPU load, memory or virtual machine efficiency values.

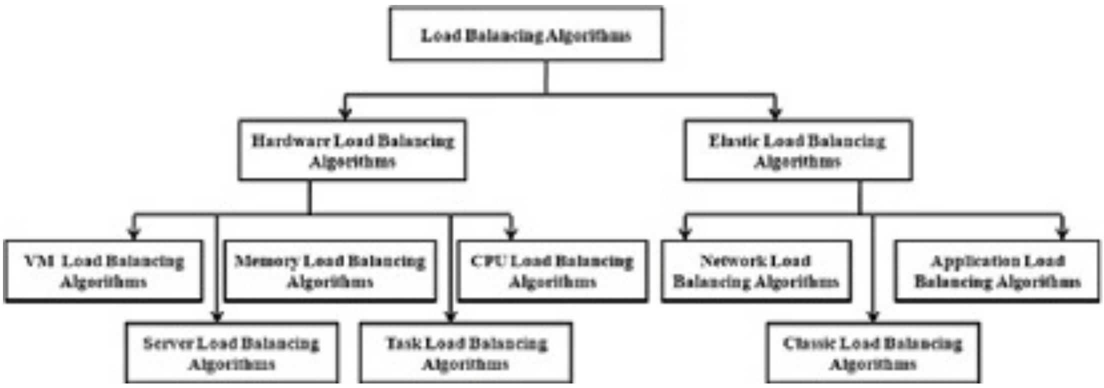


Figure1. Hierarchical categorization of load balancing models.

Elastic load balancers aim at optimizing a group of parameters during load balancing. For instance, network load balancers optimize virtual machine and server parameters; while application load balancers aim at optimization of memory, CPU load and task parameters. A combination of these parameters is optimized by classic load balancers, wherein depending upon the application; one or more cloud task parameters are optimized. An in-depth survey of these optimization models can be referred from the next section; wherein various machine learning models for load balancing are described. This is followed by statistical analysis of these models; their comparative evaluation. The evaluation assists in identification of best suited for models for any given application; which will assist researchers and system designers for high speed and high efficiency system design. Finally, this text concludes with some interesting observations about the reviewed algorithms; and recommends methods to improve them.

2. LITERATURE REVIEW

https://doi.org/10.17993/3cemp.2022.110250.232-248

A wide variety of energy efficient cloud load balancing models have been proposed by researchers and cloud designers. These algorithms aim at improving energy efficiency via application of different machine learning models including but not limited to bio-inspired models, swarm-optimization models, neural networks, etc. The work in (Panda, Moharana, Das, and Mishra, 2019)[1] introduces such an energy efficient model that utilizes virtual machine consolidation in cloud environments.

The model aims at minimizing energy consumption during virtual machine (VM) migration process via threshold-based sleep scheduling. Here, virtual machines with lower load levels are put to sleep for specified clock cycles, thereby assisting in energy reduction. Due to this sleep scheduling, VMs with high loads are easily identified. This identification assists in assigning tasks to the sleep mode VMs, thereby reducing the probability of load imbalance. Description of the model can be observed from figure 2, wherein load balancing and VM migration processes are described.

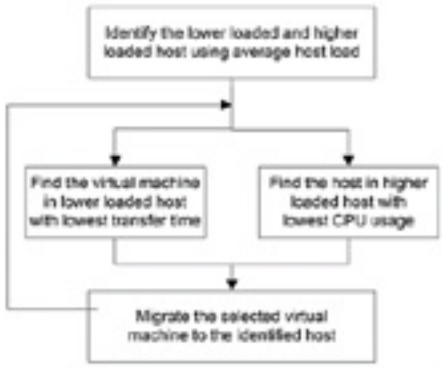


Figure2. Energy efficient VM migration and load balancing.

Figure 2. Energy efficient VM migration and load balancing (Panda et al., 2019)[1] Due to efficient migration of VMs, and proper load division, the model is able to reduce load imbalance levels by 15% energy consumption by 20% when compared with non-energy aware models. A similar model that uses dynamic energy efficient resource (DEER) allocation can be observed from (Rehman, Ahmad, Jehangiri, Ala'Anzy, Othman, Umar, and Ahmad, 2020)[2], wherein resource with minimum utilization is used for load balancing. This model also performs intelligent sleep scheduling, due to which an energy efficiency of 15% is achieved when compared with Dynamic Resource Allocation Strategy (DRAM) model. This model also reduces computational cost by 10% when compared with DRAM, thereby making is useful for real time load balancing applications. A load-based model that uses self-organizing maps (SOMs) can be observed from (Malshetty and Mathapati, 2019)[3], wherein cluster heads are created, and all load requests are handled by them.

The selected cluster heads are able to reduce computational load on centralized server, and select machines with minimum power consumption, thereby improving overall energy efficiency of the system. It is observed that the proposed model is able to reduce energy consumption by 18% and delay of processing by 10% when compared with Low-energy adaptive clustering hierarchy (LEACH) model. This approach can be compared with other models, a survey of these models can be observed from (Ala'anzy and Othman, 2019)[4], wherein effects load balancing & server consolidation are studied. The work compares algorithms like load-aware Global resource affinity management, advanced prediction-based minimization of load migration, multidimensional hierarchical VM migration, extended first fit decreasing algorithm, locusts inspired scheduling algorithm, etc.

It is observed that soft computing models outperform linear models in terms of energy efficiency & load balancing performance. An example of such a soft computing model can be observed from (Salem Alatawi and Abdullah Sharaf, 2020)[5], wherein Honey Bee optimization is combied with fuzzy logic for improved energy efficiency during load balancing. The model uses a fuzzy approach for host & VM selection, and then deploys a Honey Bee optimization model for load scheduling

https://doi.org/10.17993/3cemp.2022.110250.232-248

between these components. Due to incorporation of VM energy levels, and task length in the fitness function (described in equation 1), the model is able to schedule tasks with good energy efficiency.

$$Fitness = \frac{T_{length} * T_{deadline}}{E_{vm}} \ldots (1)$$

Where, T length T deadline are task length, and task deadline; while E vm is the per task execution energy of the VM. The model aims are reducing this fitness value in order to improve the energy efficiency, and execute tasks of the given length under the given deadline. The model is able to reduce energy consumption by up to 18% when compared with only fuzzy model, and up to 15% when compared with only the Honey Bee optimization model, thereby making it applicable for real time use. Execution of tasks with high energy efficiency must be accompanied with effective placement of VM services. This placement allows schedulers to select nearby VMs in order to execute tasks with high efficiency and low energy consumption. Example of such an architecture can be observed from (Alharbi, El-Gorashi, and Elmirghani, 2019)[6], wherein researchers have showcased the use of cloud-to-fog load balancing reduces energy consumption by 75% when compared with only-cloud load balancing architecture. This architecture can be observed from figure 3, wherein data from cloud is offloaded to fog nodes for efficient balancing.

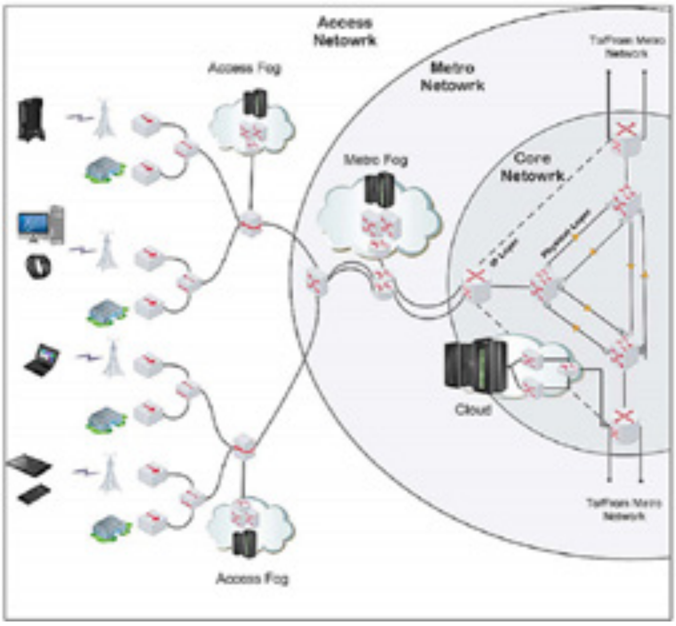


Figure3. Offloading data from single cloud to fog nodes for energy efficient load balancing [6].

Figure 3. Offloading data from single cloud to fog nodes for energy efficient load balancing [6] Due to offloading of tasks on fog nodes, the computational load is shared between different nodes, thereby assisting in faster task execution, and better energy efficiency. This efficiency can be further improved via addition of soft computing models. Such a model is defined in (Dong, Xu, Ding, Meng, and Zhao, 2019)[7], wherein glow worm swarm optimization (GSO) is used. The model combines clustering for extraction of large task resources, with sine cosine analysis (SCA) in order to modify the step size of GSO for energy adaptive scheduling. This modification in step size allows the GSO model to select load & energy optimized cloud resources for the given task.

The clustered input tasks are divided into edge group & cloud group; wherein each group has task elements based on CPU utilization and memory consumption. For instance, the edge group retains

https://doi.org/10.17993/3cemp.2022.110250.232-248

tasks with high CPU utilization but low memory consumption, while the cloud group retains tasks with high memory consumption. This task division is governed using the following equation,

$$L\_group - \partial * L\_cpu + \emptyset * L\_memory \ldots (2)$$

Where, L group is resource requirement for the given group, L cpu is resource requirement w.r.t CPU utilization, L memory is resource requirement w.r.t. memory utilization, while and are cloud and edge constants. The model is compared with First Fit Decreasing (FFD) & OTS models and it is observed that the GSO model has 25% better energy efficiency, 15% better throughput, and 18% better load balancing degree when compared with these algorithms. This efficiency can be further improved by performing computations on fog devices as observed from (Bhuvaneswari and Akila, 2019)[8], wherein comparison of different fog-based load balancing algorithms w.r.t. their energy performance is studied. Algorithms like ant colony optimization (ACO), max-min algorithm, active monitoring for load balancing (AMLB), and round robin (RR) are compared. It is observed that the ACO based soft computing model has 15% better efficiency when compared with RR, while AMLB when combined with ACO provides 25% better energy efficiency than individual algorithms. Another hybrid combination that uses ACO with support vector machines (SVM) can be observed from (Junaid, Sohail, Ahmed, Baz, Khan, and Alhakami, 2020)[9], wherein file type formatting (FTF) is used for load classification. Input requests are given to SVM model and depending upon file type at input, the model classifies load types into low power, medium power and high power. After this, the classified load is given to ACO, wherein VM to load mapping is performed using greedy heuristics. Flow of the model can be observed from figure 4, wherein training phases and testing phases can be seen with the final load balancing process. It is observed that the model performs SVM training on file formats, and provides the classified results to ACO. The ACO model internally maps suitable VMs to tasks for high energy efficiency. Due to this, the proposed SVMFTF model provides 8% better energy efficiency than random forest (RF), 6% than Na¨ıve Bayes, 9% than k-nearest neighbours (kNN), and 4% than convolutional neural network (CNN) models.

https://doi.org/10.17993/3cemp.2022.110250.232-248

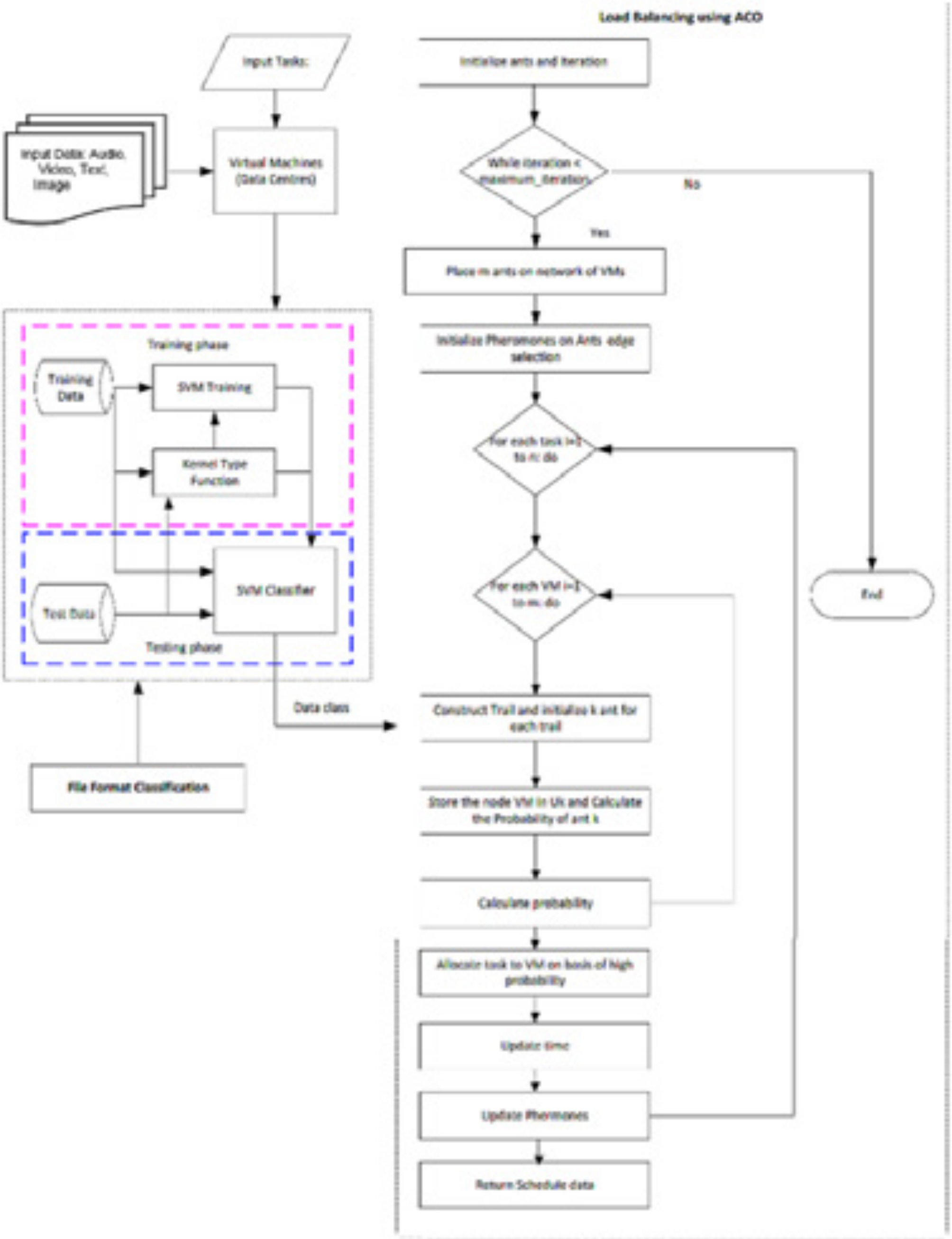


Figure4.SVMFTF model for energy efficient load balancing [9].

Figure 4. SVMFTF model for energy efficient load balancing [9] The model also showcases better throughput, and migration performance when compared with RF, Na¨ıve Bayes, kNN, CNN models. Specifically, the model provides 8% better throughput than RF, 3% better than Na¨ıve Bayes, 15% better than kNN, and 5% better than CNN; while it provides 10% better migration performance than RF, 4% better than Na¨ıve Bayes, 8% better than kNN, and 5% better than CNN models when compared on the same task set and cloud configurations . Comparison of this model can be done with other standard load balancing models as suggested in (Dey and Gunasekhar, 2019)[10], wherein models like first in first out (FIFO), fair scheduling, capacity scheduling, hybrid scheduling, longest approximate time to end scheduling, self-adaptive mapreduce scheduling, and context-aware scheduling for Hadoop are discussed. Each of these models are compared on parameters like Job Characteristics, Responsiveness, Resource Pool Configuration, Queue Characteristics, Parallelization of Tasks, Queue Responsiveness, Dynamic Priority, Locality Management, Remaining Burst Time, Task Priority, Context and Energy Efficiency. All these parameters are evaluated on the Planet Lab dataset, wherein it is observed that Robust Local Regression (RLR) methods like self-adaptive map-reduce scheduling, and context-aware scheduling outperform other methods by over 10% in terms of

energy efficiency. The same trend is observed for other parameters, due to the sophistication of the RLR methods, and in-depth analysis for the given tasks.

A similar study like [10] can be observed in (Kulshrestha and Patel, 2019)[11], wherein models like ALB, transport layer load balancer (TLLB), network layer load balancer (NLLB), VM provisioning on host, consolidation of VMs on host, and VM-level task scheduling are described. Out of these algorithm ALB outperforms other models in terms of energy efficiency by providing 8% better performance than TLLB, 5% better performance than VM provisioning, and 15% better performance than VM consolidation. An example the ALB scheme can be observed in (Zhang, Jia, Gu, and Guo, 2019)[12], wherein Matrix sparseness with normalizedWater-Filling (MSNWF) is described. This model is compared with Heterogeneous Network (HETNET), and OPT models, and it is observed that MSNWF provides 15% better energy efficiency than HETNET and 5% better energy efficiency than OPT. Thus, the MSNWF model can be used for high performance cloud load balancing applications, wherein along with efficiency of task scheduling, energy efficiency is also improved. This model performance can be further improved by integration of broker service policy for software as a service (SaaS) application. Modelling of such architectures requires high efficiency broker design, wherein any incoming task is first given to a broker for estimation of approximate processing site. This estimation allows the cloud VMs to pre-allocate resources for the task, thereby improving the task execution efficiency. In order to model such brokers, architectures like shortest job scheduling, Min-min, Max-min, Two-phase (OLB + LBMM), Modified active monitoring, Throttled Load Balancer, Genetic Algorithm, Honey Bee foraging algorithm, ACO, etc. are available (Jyoti, Shrimali, Tiwari, and Singh, 2020)[13]. It is observed that ACO and other soft computing models when utilizing fog and cloud computing, outperform other models in terms of energy efficiency. An example of such a model can be observed from (Lin, Peng, Bian, Xu, Chang, and Li, 2019)[14], wherein the soft computing models are deployed on cloud. The results of these models are VM-to-task mapping, which are executed either on the cloud infrastructure or offloaded to the fog device for better load balancing capabilities. The model for this architecture can be observed from figure 5, wherein offloading process is performed using different wireless standards.

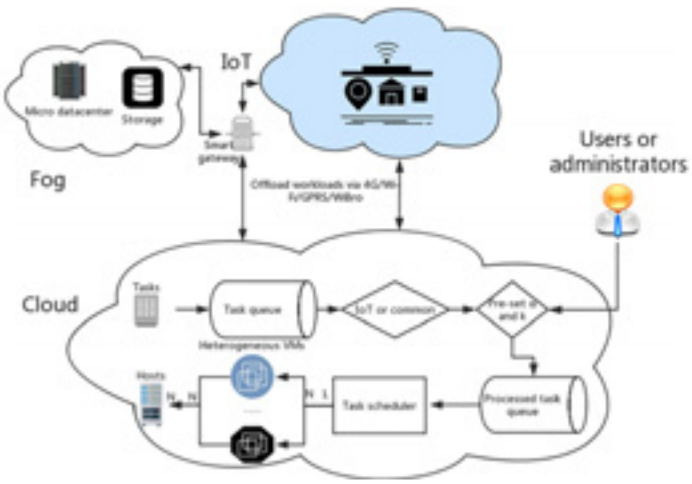


Figure5. Fog-cloud load balancing model with soft computing for efficient task mapping [14].

Figure 5. Fog-cloud load balancing model with soft computing for efficient task mapping [14] The model utilizes total amount of resource in host, total amount of resource in VM, estimated execution time of task, normalization of task average demand for resource, normal resource load of task, relative load of task for resource upon VM, task length, and task priority for scheduling. It is observed that the proposed soft computing model outperforms first come first serve by 15%, random assignment by 25%, trade-off by 8%, main resource load & time balancing by 16%, and main resource task balance by 9% in terms of energy efficiency. This is due to design of energy & task length aware fitness function design, which suggests that soft computing must be used for any kind of load balancing models. A similar model is proposed in (Mandal, Mondal, Banerjee, and Biswas, 2020)[15], wherein

https://doi.org/10.17993/3cemp.2022.110250.232-248

service level agreement (SLA) is used for detection of task overload at different hosts. It uses a mapping ratio that consists of VM utilization and allocated resource characteristics in order to assign tasks to non-overloaded VMs. The value of this mapping ratio (Mapping r) can be observed from equation 3, wherein both the parameters are split into task & resource related characteristics.

$$Mapping_r = \frac{VM\_mips + VM\_RAM + VM\_BW}{T_{length} + T_{deadline}} \ldots (3)$$

where in, VM mips, VM RAM & VM BW are VM specific capacity, available RAM & bandwidth while Tlength & Tdeadline are task related length & deadline parameters. It is observed that the proposed model outperforms minimum migration time MMT by 45%, maximum correlation MaxCorr by 35%, minimum utilization (MU) by 46%, and random selection RS by 34%, thereby making it highly useful for real time cloud deployments. Context-aware load balancing models have better efficiency than energy-aware, or task-length aware models because these models adaptively modify their internal rules depending upon the context of given task and condition of the VMs. Such a model that utilizes context information for energy efficient load balancing is described in (Royaee, Mirvaziri, and Khatibi Bardsiri, 2021)16], wherein automata ant colony based multiple recursive routing protocol (AMRRPL) is used. The model solves issues like bottlenecking, efficient parameter selection, effect of upstream nodes, and congestion which are inherent with load balancing. The model uses destination oriented directed acyclic graph (DODAG) in order to perform load balancing via laying out all possible VM-to-Task combinations on an acyclic graph. Due to use of DODAG the model is able to achieve an energy efficiency of 8% when compared with ERPL (enhanced RPL), and 45% when compared with HECRPL (hybrid energy efficient RPL) and its configurations. This model can be applied to various applications including software defined network (SDN), content delivery network (CDN), cost-based distribution (CBD) networks, etc. for highly energy efficient load balancing. An example of this application for CDN can be observed in (Gupta, Goyal, and Gupta, 2015)[17], wherein a reliability aware load balancer model is applied.

The model uses a modified version of Genetic Algorithm (GA) for task scheduling, and is able to obtain 15% better energy efficiency when compared with queue length-based load balancing (QLBLB) model. Another low power model that uses first of maximum loss scheduling algorithm (FOML) is described in (Liang, Dong, Wang, and Zhang, 2020)[18], wherein relationship between energy utilization & average completion time is used. The model selects VM with maximum energy utilization and assigns it to a task that has average completion time (when compared to all tasks in queue). This task is then deleted, and a new average completion task is evaluated and assigned to the next maximum energy utilization VM. This process makes sure that all the high energy consuming VMs are assigned to moderate sized tasks, while other VMs are assigned to large & small sized tasks. Flow of this model can be observed from figure 6, wherein ETC (extended time of completion) and ACT (average completion time) matrices are evaluated for the given set of tasks.

https://doi.org/10.17993/3cemp.2022.110250.232-248

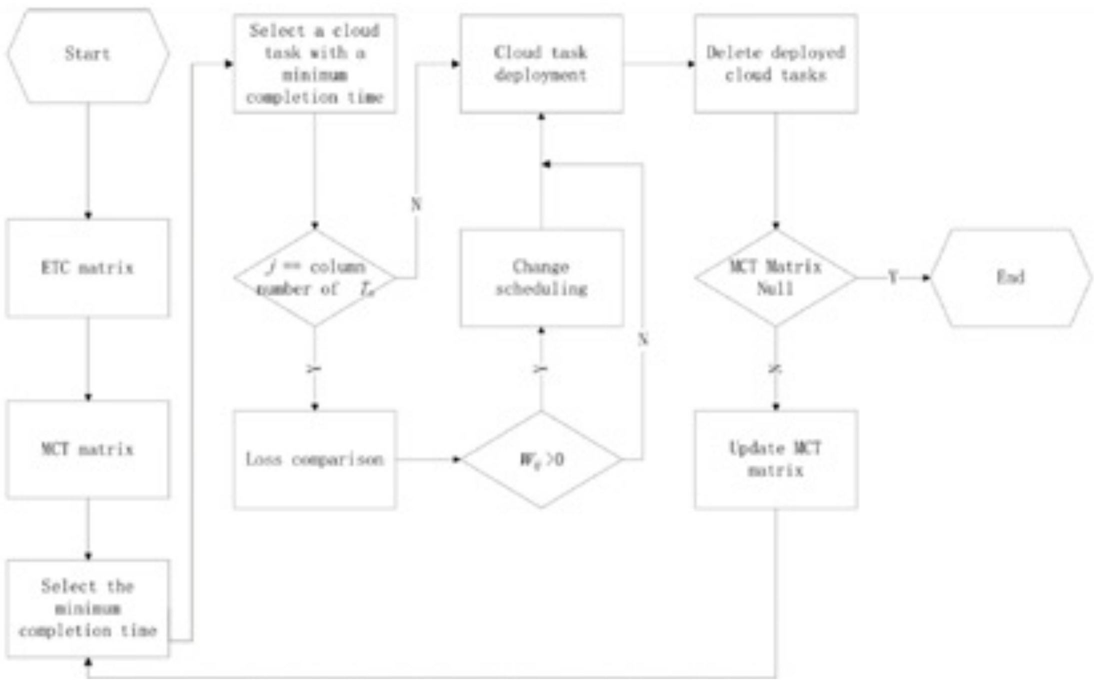


Figure6.Maximum energy utilization model for efficient task scheduling [18].

Figure 6. Maximum energy utilization model for efficient task scheduling [18] The FOML model is able to obtain 10% better energy utilization when compared with min-min model, 8% better utilization when compared with max-min model, 12% when compared with suffrage model, and 15% better than E-HEFT model, thereby making it highly effective for real time deployments.

Similar models are described in (Hadikhani, Eslaminejad, Yari, and Ashoor Mahani, 2020)[19], and (Sadeghi and Avokh, 2020)[20], wherein geographic information and Two-hop Routing Tree with Cuckoo search (CCTRT) models are defined. These models able to achieve 5% and 12% better energy efficiency when compared with E-HEFT model, thereby suggesting that the use of soft computing models is fundamental to design of energy efficient load balancing algorithms. The efficiency of these models can be further improved via use of predictive workload balancing, wherein the system is able to predict workloads depending upon task patterns, and pre allocate cloud & fog VMs for efficient execution. Architecture for such a system can be observed in (Jodayree, Abaza, and Tan, 2019)[21], wherein rule-based workload prediction is defined. It uses a combination of historical data analysis and random workload assignment in order to speed up workload balancing. Due to predictive analysis, the model is able perform host reduction and thereby reduce energy consumption by 10% when compared with random assignment algorithm.

This model can be further extended via use of deadline constrained task scheduling as suggested in (Ben Alla, Ben Alla, Touhafi, and Ezzati, 2019)[22], wherein a dynamic classifier is used to divide incoming tasks into priority queues, and each of these queues is processed using Fuzzy Logic and Particle Swarm Optimization model (FLPSO). The FLPSO model used in this approach is able to reduce energy consumption by 60% when compared with FCFS (first come first serve), 25% when compared with EDF (earliest deadline first), and 15% when compared with Differential Evolution (DE) with Multiple Criteria Decision Making (MCDM) algorithms. This comparison appears to be true for different VM and task combinations, thereby assisting in deploying the FLPSO model for a wide variety of cloud infrastructures.

This approach must be compared with other models like the ones mentioned in (Pourghebleh and Hayyolalam, 2020)[23] in order to evaluate its real time applicability and deployment capabilities.

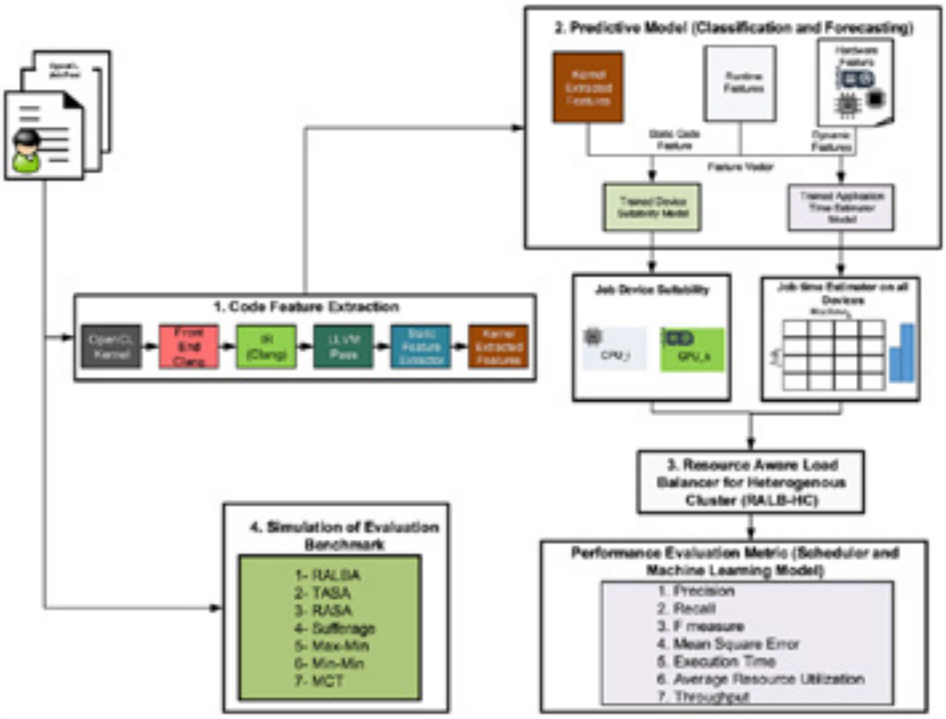
Similar energy efficient models are discussed in (Rashid, Tripathi, Prakash, and Tripathi, 2019)[24], (Singh and Kumar, 2019a)[25] and (Kansal and Chana, 2018)[26] wherein load based energy efficiency, security aware energy efficiency, and migration aware energy efficiency models are

https://doi.org/10.17993/3cemp.2022.110250.232-248

described. Each of these models utilize soft computing techniques like ACO, PSO, GA, and GSO in order to achieve high energy efficiency. A resource aware load balancing model can be observed in (Ahmed, Aleem, Noman Khalid, Arshad Islam, and Azhar Iqbal, 2021)[27], wherein heterogenous clustering is used in order to perform resource-based task mapping. The model performs job to resource mapping depending upon resource availability, and resource aware load balancing for obtaining higher utilization ratio. It uses a predictive model for classification and forecasting job device suitability & job time estimation matrix as observed from figure 7, wherein the overall model is described. The model extracts features including front end clang (percentage of tasks remaining), kernel features ratio of task length to current machine configuration), and static features (initial performance of machines and number of tasks) from tasks and provides them to Resource aware load balancing and hierarchical clustering (RALBHC) model for improvement of resource utilization.

The model is able to achieve an energy efficiency of 25% when compared with Max-min algorithm, 15% when compared with Minimum Completion Time, 8% when compared with Resource-Aware Scheduling Algorithm (RASA), and 10% when compared with Task-Aware Scheduling Algorithm (TASA). Another energy efficient model that uses equal load distribution for fog-to-cloud & cloudto-fog migration (EDCW) is described in (Kaur and Aron, 2020)[28], wherein linear programming (LP) is used. The use of LP results into equal distribution of tasks between fog node and cloud node, thereby assisting in improved load balancer performance. The model is able to achieve 15% better energy efficiency when compared with Round Robin model, and 8% better efficiency when compared with throttled model, thereby making it useful for low energy load balancing applications.

Similar energy-efficient models are proposed in (Escobar, Ortega, D´ıaz, Gonz´alez, and Damas, 2019)[29], (?, ?)[30], (Taboada, Aalto, Lassila, and Liberal, 2017)[31],(Kumar, Singh, and Mohan, 2021)[32], and (Singh and Kumar, 2019b)[33], where in parallel evolutionary algorithms, context-based load balancing, energy-aware load balancing, resource-efficient load-balancing, and secure load balancing models are described. These models make use of different soft computing methods in order to perform task-based & resource-based load balancing. The underlying models are able to reduce energy consumption via optimization of the fitness function, wherein resource energy, task length, task deadline, and resource performance parameters are used. A quantitative analysis of these models is described in the next section, wherein the underlying models are compared in terms of relative energy efficiency values, thereby assisting cloud system designers to identify energy-efficient load-balancing models for their deployment.



https://doi.org/10.17993/3cemp.2022.110250.232-248

Figure7.Maximum energy utilization model for efficient task scheduling [18].

3. QUANTITATIVE ANALYSIS

From the literature, it is observed that the energy efficiency of different cloud load balancing algorithms is estimated using their relative percentages. This limits the quantitative comparison capability of these algorithms. In order to resolve this drawback, this section evaluates absolute percentage energy efficiency when compared with the basic first come first serve (FCFS) algorithm.

This process will allow researchers to estimate energy performance of reviewed algorithms, along with their computational complexity. The computational complexity is divided into fuzzy ranges of low (L), medium (M), high (H) and very high (VH). Maintaining a balance between energy efficiency and computational complexity is a must while designing load balancing models for cloud. The quantitative results are tabulated in table 1, wherein the aforementioned parameters are compared across different algorithms.Based on this analysis it can be observed that the FLPSO (Ben Alla et al., 2019)[22], AMRRPL (Royaee et al., 2021)[16], MU (Mandal et al., 2020)[15], SLA based model (Mandal et al., 2020)[15], EDF(?, ?)[22], and GSO SCA(Dong et al., 2019)[7] model outperform other models in terms of relative energy efficiency.This performance evaluation can also be observed from the visualization in figure 8, wherein different algorithms and their accuracies are compared. It can also be observed that CNN and other deep learning models are not used for energy efficient load balancing, because training of these models for dynamic loads is resource intensive, thereby requires large amount of power.

https://doi.org/10.17993/3cemp.2022.110250.232-248

Model Used	Energy efficiency %	Computational Complexity
Threshold based sleep scheduling [1]	15	M
DEER [2]	12	M
DRAM [2]	8	M
Load based SOM [3]	22	H
Honey bee optimization [5]	25	H
Fuzzy logic [5]	15	M
Cloud to fog migration [6]	34	H
Single cloud [6]	18	M
GSO SCA [7]	33	M
FFD [7]	23	M
OTS [7]	15	L
ACO [8]	18	M
AMLB [8]	8	L
RR [8]	3	L
ACO FTF SVM [9]	12	H
RF [9]	5	M
Naive Bayes [9]	8	M
kNN [9]	2	L
CNN [9]	0	VH
RLR [10]	15	H
NLLB [11]	10	M
MSNWF [12]	25	VH
HETNET [12]	15	H
Fog-cloud model [14]	15	H
Random assignment [14]	22	M
SLA based model [15]	48	H
MMT [15]	40	M
Max Corr. [15]	30	M
MU [15]	41	M
RS [15]	31	M
AMRRPL [16]	48	H
ERPL [16]	16	M
HECRPL [16]	5	M
FOML [18]	15	M
Min to Min [18]	6	L
E-HEFT [18]	23	M
CCTRT [20]	18	H
Rule based prediction [21]	15	M
FLPSO [22]	60	H
EDF [22]	48	M
DE [22]	26	M
MCDM [22]	31	M
RALBHC [27]	35	H
Max to Min [28]	18	M
RASA [28]	2	M
TASA [28]	25	M

Moreover, standard CNN models are also not available for this purpose, therefore it is a necessity that researchers should develop such models that aim towards energy efficiency. These models can then be extended via transfer learning or recurrent networks in order to incrementally tune their performance. Neural network models have very low energy consumption during evaluation, thus pre-training of models is further recommended.

https://doi.org/10.17993/3cemp.2022.110250.232-248

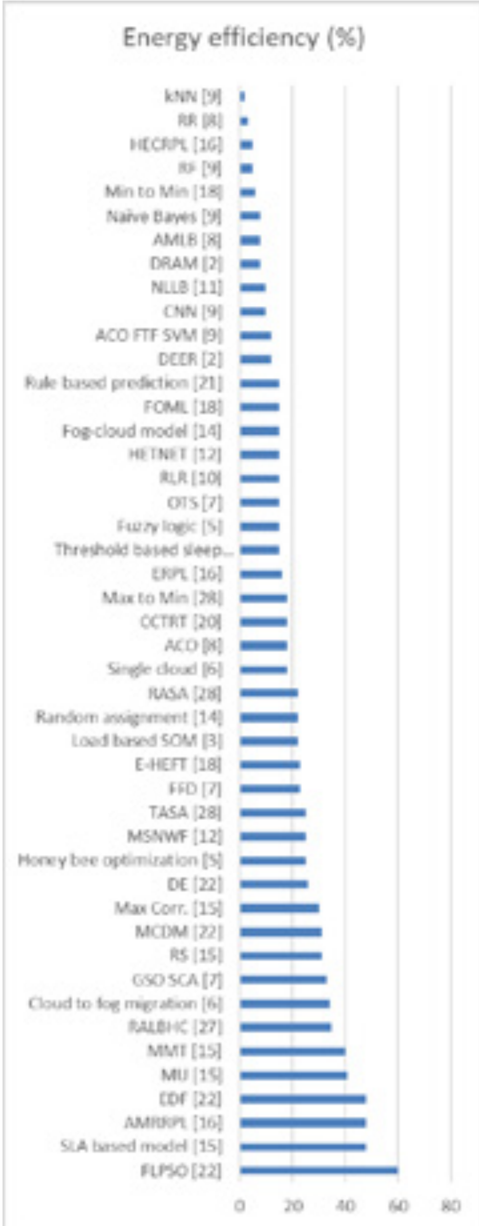


Figure8.Energy efficiency of different load balancing models.

4. CONCLUSION AND FUTURE SCOPE

The comparative quantitative analysis indicates that FLPSO (Ben Alla et al., 2019)[22], SLA based model (Mandal et al., 2020)[15], AMRRPL (Royaee et al., 2021)[16], EDF(Ben Alla et al., 2019)[22], MU (Mandal et al., 2020)[15], MMT [15], RALBHC (Ahmed et al., 2021)[27], Cloud to fog migration (Alharbi et al., 2019)[6], GSO SCA (Dong et al., 2019)[7], RS (Mandal et al., 2020)[15], MCDM (Ben Alla et al., 2019)[22], and Max Corr. (Mandal et al., 2020)[15] outperform linear models like kNN (Junaid et al., 2020)[9], HECRPL (Royaee et al., 2021)[16], CNN (Junaid et al., 2020)[9], and Rule based prediction (Jodayree et al., 2019)[21] in terms of energy efficiency.

Energy efficient models utilize soft computing techniques like PSO, ACO, GA, GSO, and Honey Bee Optimization in order to achieve this task via energy aware fitness function design. Deep learning models are not used for this purpose due to their energy intensive training process.

This limitation can be removed via using pre-trained CNN models that are optimized for energy efficient load balancing. Furthermore, existing models like SLA based model [15], AMRRPL (Royaee et al., 2021)16], EDF (Ben Alla et al., 2019)[22], MU (Mandal et al., 2020)[15], MMT [15], RALBHC (Ahmed et al., 2021)[27], etc. can be further improved via addition of soft computing for optimization

https://doi.org/10.17993/3cemp.2022.110250.232-248

of energy consumption. These additions will enhance system performance and help the models to be tuned with high energy efficiency.

REFERENCES

[1] Ahmed, U., Aleem, M., Noman Khalid, Y., Arshad Islam, M., and Azhar Iqbal, M. 2021. RALB-HC: A resource-aware load balancer for heterogeneous cluster. Concurr. Comput. 33, 14 (July).

[2] Ala'anzy, M. and Othman, M. 2019. Load balancing and server consolidation in cloud computing environments: A meta-study. IEEE Access 7, 141868–141887.

[3] Alharbi, H. A., El-Gorashi, T. E. H., and Elmirghani, J. M. H. 2019. Energy efficient virtual machine services placement in cloud-fog architecture. In 2019 21st International Conference on Transparent Optical Networks (ICTON). IEEE.

[4] Ben Alla, S., Ben Alla, H., Touhafi, A., and Ezzati, A. 2019. An efficient energy-aware tasks scheduling with deadline-constrained in cloud computing. Computers 8, 2 (June), 46.

[5] Bhuvaneswari, D. and Akila, A. 2019. An energy efficient management in various fog platforms. In 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon). IEEE.

[6] Dey, N. S. and Gunasekhar, T. 2019. A comprehensive survey of load balancing strategies using hadoop queue scheduling and virtual machine migration. IEEE Access 7, 92259– 92284.

[7] Dong, Y., Xu, G., Ding, Y., Meng, X., and Zhao, J. 2019. A 'joint-me' task deployment strategy for load balancing in edge computing. IEEE Access 7, 99658–99669.

[8] Escobar, J. J., Ortega, J., D´ıaz, A. F., Gonza´lez, J., and Damas, M. 2019. Energyaware load balancing of parallel evolutionary algorithms with heavy fitness functions in heterogeneous CPU-GPU architectures. Concurr. Comput. 31, 6 (Mar.), e4688.

[9] Gupta, P., Goyal, M. K., and Gupta, N. 2015. Reliability aware load balancing algorithm for content delivery network. In Advances in Intelligent Systems and Computing. Advances in intelligent systems and computing. Springer International Publishing, Cham, 427–434.

[10] Hadikhani, P., Eslaminejad, M., Yari, M., and Ashoor Mahani, E. 2020. An energyaware and load balanced distributed geographic routing algorithm for wireless sensor networks with dynamic hole. Wirel. netw. 26, 1 (Jan.), 507–519.

[11] Jodayree, M., Abaza, M., and Tan, Q. 2019. A predictive workload balancing algorithm in cloud services. Procedia Comput. Sci. 159, 902–912.

[12] Junaid, M., Sohail, A., Ahmed, A., Baz, A., Khan, I. A., and Alhakami, H. 2020. A hybrid model for load balancing in cloud using file type formatting. IEEE Access 8, 118135–118155.

[13] Jyoti, A., Shrimali, M., Tiwari, S., and Singh, H. P. 2020. Cloud computing using load balancing and service broker policy for IT service: a taxonomy and survey. J. Ambient Intell. Humaniz. Comput. 11, 11 (Nov.), 4785–4814.

[14] Kansal, N. J. and Chana, I. 2018. An empirical evaluation of energy-aware load balancing technique for cloud data center. Cluster Comput. 21, 2 (June), 1311–1329.

[15] Kaur, M. and Aron, R. 2020. Equal distribution based load balancing technique for fog-based cloud computing. In Algorithms for Intelligent Systems. Springer Singapore, Singapore, 189–198.

[16] Kulshrestha, S. and Patel, S. 2019. A study on energy efficient resource allocation for cloud data center. In 2019 Twelfth International Conference on Contemporary Computing (IC3). IEEE.

[17] Kumar, J., Singh, A. K., and Mohan, A. 2021. Resource-efficient load-balancing framework for cloud data center networks. ETRI J. 43, 1 (Feb.), 53–63.

[18] Liang, B., Dong, X.,Wang, Y., and Zhang, X. 2020. A low-power task scheduling algorithm for heterogeneous cloud computing. J. Supercomput. 76, 9 (Sept.), 7290–7314.

[19] Lin, W., Peng, G., Bian, X., Xu, S., Chang, V., and Li, Y. 2019. Scheduling algorithms for heterogeneous cloud environment: Main resource load balancing algorithm and time balancing algorithm. J. Grid Comput. 17, 4 (Dec.), 699–726.

[20] Malshetty, G. and Mathapati, B. 2019. Efficient clustering in WSN-Cloud using LBSO (load based Self-Organized) technique. In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI). IEEE.

[21] Mandal, R., Mondal, M. K., Banerjee, S., and Biswas, U. 2020. An approach toward design and development of an energy-aware VM selection policy with improved SLA violation in the domain of green cloud computing. J. Supercomput. 76, 9 (Sept.), 7374–7393.

[22] Panda, B., Moharana, S. C., Das, H., and Mishra, M. K. 2019. Energy aware virtual machine consolidation for load balancing in virtualized environment. In 2019 International Conference on Communication and Electronics Systems (ICCES). IEEE.

[23] Pourghebleh, B. and Hayyolalam, V. 2020. A comprehensive and systematic review of the load balancing mechanisms in the internet of things. Cluster Comput. 23, 2 (June), 641–661.

[24] Rashid, A., Tripathi, Y., Prakash, A., and Tripathi, R. 2019. Load aware energy-balanced data gathering approach in CRSNs. IET Wirel. Sens. Syst. 9, 3 (June), 143–150.

[25] Rehman, A. U., Ahmad, Z., Jehangiri, A. I., Ala'Anzy, M. A., Othman, M., Umar,

[26] A. I., and Ahmad, J. 2020. Dynamic energy efficient resource allocation strategy for load balancing in fog environment. IEEE Access 8, 199829–199839.

[27] Royaee, Z., Mirvaziri, H., and Khatibi Bardsiri, A. 2021. Designing a context-aware model for RPL load balancing of low power and lossy networks in the internet of things. J. Ambient Intell. Humaniz. Comput. 12, 2 (Feb.), 2449–2468.

[28] Sadeghi, F. and Avokh, A. 2020. Load-balanced data gathering in internet of things using an energy-aware cuckoo-search algorithm. Int. J. Commun. Syst. 33, 9 (June), e4385.

[29] Salem Alatawi, H. and Abdullah Sharaf, S. 2020. Toward efficient cloud services: An energy-aware hybrid load balancing approach. In 2020 International Conference on Computing and Information Technology (ICCIT-1441). IEEE.

[30] Singh, A. K. and Kumar, J. 2019a. Secure and energy aware load balancing framework for cloud data centre networks. Electron. Lett. 55, 9 (May), 540–541.

[31] Singh, A. K. and Kumar, J. 2019b. Secure and energy aware load balancing framework for cloud data centre networks. Electron. Lett. 55, 9 (May), 540–541.

[32] Taboada, I., Aalto, S., Lassila, P., and Liberal, F. 2017. Delay- and energy-aware load balancing in ultra-dense heterogeneous 5G networks. Trans. emerg. telecommun. technol. 28, 9 (Sept.), e3170.

[33] Zhang, X., Jia, M., Gu, X., and Guo, Q. 2019. An energy efficient resource allocation scheme based on cloud-computing in H-CRAN. IEEE Internet Things J. 6, 3 (June), 4968–4976.

https://doi.org/10.17993/3cemp.2022.110250.232-248

AUTORS BIOGRAPHY

Dr. Kavita Singh has completed her doctoral degree from SVNIT, Surat in 2014 in biometrics. Her areas of expertise are Machine Learning, soft computing, computer vision, latex and Image Pro-cessing. She has more than 50 international paper publications in reputed conferences, book chapters and journals. She worked as reviewer, session chair for many international journals and conferences. She has delivered 20 talks on different topics in-cluding her expertise area. She worked as Principal Investigator for projects funded by AICTE under RPS Scheme. She has also a strong association with industries like IBM, Nvidia etc. She is currently working as Associate Prof. at Department of Computer Technology, YCCE, Nagpur.

Mr. Amol D Gaikwad has received his B.Tech degree in Computer Science and Engineering from RTMNU Nagpur University, Nag-pur and MTech. degree in Computer Science and Engineering Master of Technology degree from Yeshwantrao Chavan College of Engineering, Engineering (An Autonomous Institution Affili-ated to Rashtrasant Tukadoji Maharaj Nagpur University), Nag-pur, India. He is currently pursuing Ph.D. from from Yeshwan-trao Chavan College of Engineering, (An Autonomous Institu-tion Affiliated to Rashtrasant Tukadoji Maharaj Nagpur Univer-sity), Nagpur, India. His predominant research areas include High Performance Computing, Cloud Computing. He has pub-lished research papers in various reputed international confer-ences and journals. He is the life member of ISTE, India, IE, India.

Shailesh D. Kamble received Bachelor of Engineering degree in Computer Technology from Yeshwantrao Chavan College of Engineering, Nagpur, India under the Rashtrasant Tukadoji Ma-haraj Nagpur University, Nagpur, India. He received Master of Engineering degree from Prof Ram Meghe Institute of Technolo-gy and Research, formerly known as College of Engineering, Bandera under Sant Gadge Baba Amravati University, Amravati, India. He completed his doctoral degree from G.H. Raisoni Col-lege of Engineering (An Autonomous Institution Affiliated to Rashtrasant Tukadoji Maharaj Nagpur University), Nagpur, In-dia. He is working as the Associate Professor in Department of Computer Technology at Yeshwantrao Chavan College of Engi-neering, Nagpur, India. He is having 20 years of teaching and re-search experience. His current research interests include image processing, video processing and language processing. He is the author or co-author of more than 20 scientific publications in International Journal, International Conferences, and National Conferences. He is the life member of ISTE, India.

https://doi.org/10.17993/3cemp.2022.110250.232-248