I. J. Computer Network and Information Security, **2023, 4, 84-95**
Published Online on August 8, 2023 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2023.04.08

Deep Learning-based Resource Prediction and Mutated Leader Algorithm Enabled Load Balancing in Fog Computing

Shruthi G.*
Department of Computer Science and Engineering, M S Ramaiah Institute of Technology, VTU, Bengaluru, 590018, India
School of Computer Science and Engineering, REVA University, Bengaluru, 560064, India
E-mail: g.shruthi466@gmail.com
ORCID iD: https://orcid.org/0000-0002-8185-8134
*Corresponding author

Monica R. Mundada
Department of Computer Science and Engineering, M S Ramaiah Institute of Technology, Bengaluru, 560054, India
E-mail: monica@msrit.edu
ORCID iD: https://orcid.org/0000-0001-7262-8997

S. Supreeth
School of Computer Science and Engineering, REVA University, Bengaluru, 560064, India
E-mail: supreeth1588@gmail.com
ORCID iD: https://orcid.org/0000-0002-7097-6733

Bryan Gardiner
School of Computing, Engineering and Intelligent Systems, Ulster University, Northlands Road, Magee Campus, BT48 7JL Londonderry, Northern Ireland, United Kingdom
E-mail: b.gardiner@ulster.ac.uk
ORCID iD: https://orcid.org/0000-0001-5642-6850

Received: 14 September 2022; Revised: 07 November 2022; Accepted: 03 March 2023; Published: 08 August 2023

Abstract: Load balancing plays a major part in improving the performance of fog computing, which has become a requirement in fog layer for distributing all workload in equal manner amongst the current Virtual machines (VMs) in a segment. The distribution of load is a complicated process as it consists of numerous users in fog computing environment. Hence, an effectual technique called Mutated Leader Algorithm (MLA) is proposed for balancing load in fogging environment. Firstly, fog computing is initialized with fog layer, cloud layer and end user layer. Then, task is submitted from end user under fog layer with cluster of nodes. Afterwards, load balancing process is done in each cluster and the resources for each VM are predicted using Deep Residual Network (DRN). The load balancing is accomplished by allocating and reallocating the task from the users to the VMs in the cloud based on the resource constraints optimally using MLA. Here, the load balancing is needed for optimizing resources and objectives. Lastly, if VMs are overloaded and then the jobs are pulled from associated VM and allocated to under loaded VM. Thus the proposed MLA achieved minimum execution time is 1.472ns, cost is $69.448 and load is 0.0003% respectively.

Index Terms: Fog Computing, Mutated Leader Algorithm (MLA), Virtual Machine (VM), Deep Residual Network (DRN), Load Balancing.

1. Introduction

Recently, fog computing is a rising technology that is utilized in business applications, such as smart cities, smart campuses and so on. Fog networking is introduced by CISCO known as fogging that is an expansion of cloud computing [1,2,3,4]. Fog computing contains three major important categories, such as end device, fog nodes, and cloud infrastructure. Fog networking can be organized easily [2,5,6] and is also known as distributed technology. Also, it includes computation, networking, and control to accelerate the speed of data creation while protecting user privacy and the system [2,7,8]. This architecture increases awareness majority for cloud [2,9] and also for simple information. It is assigned middle layer among client as well as cloud layers, where cloud layer has allocated computational tasks to help the bottom layers. Also, fog layer is devised for alleviating cloud load by completing tasks and hence overloaded fog nodes struggled because of heavy demands on services. Here, few nodes are connected externally, and the other nodes remain constant. The intelligent technique is introduced to address the above situation where they reallocate the streamed work with the computational devices to develop a balanced computing environment. Load balancing method [10] is used to assign load to the network components for increasing the network performance and to boost the quality of services (QoS) [11,12]. The fair distribution of client request resources is involved to possess proper resource usages. The error from every processor is fixed and every node in the network distributes the workload which is assigned to them. The important specification is that it creates effective load balancing [12,13], where the load balancing in clouds is done with blade servers and virtual machines (VMs).

Load balancing deliver dynamic workload randomly to every node in a network such that, load balancing in clouds is known as load balancing as service (LBaaS). Though cloud computing handles all diverse data, it also contains enormous challenges in real time processing and load balancing which process huge amount of data [14]. Load balancing plays major role in enhancing the fog computing performance. The load distribution is a difficult mechanism because it contains more number of users that leads to variation in load of fog node [15]. For improving the efficiency, load balancing techniques help end users and service providers. Before implementing corresponding work, load balancing [16] is used to assign the traffic bottleneck and is same as of network traffic control policy [11]. The load balancing algorithm contains two versions namely static and the arbitrary. The static load balancing strategies are for the stable environment along the homogeneous platform, but the dynamic load balancing algorithms much adaptive and efficient among the homogeneous and heterogeneous environment [17]. Mostly it is used for improving system performance, to preserve the system constancy and also to develop the fault tolerance systems at low cost. The load balancing schemes are specified based on the benefits and the challenges that are majorly based on the system capacity for distributing workload [18]. These techniques randomly distribute their load to VMs and give out all workload equally to the current VMs in the section [13]. The core drive of load balancing technique [19] is to provide equal load to servers in cluster to avoid bottleneck [20].

Resource prediction and inappropriate load allocation could lead to under and over utilization while moving the tasks between the nodes, balancing load in fog networking has been a crucial task. So, distribution of tasks equally aims to provide services. This research is mainly designed to develop a deep learning-based resource prediction and MLA enabled load balancing in a fog networking. Initially, fogging is initialized with cloud layer, fog layer, and end user layer. Then task is submitted from end user under fog layer with cluster of nodes. Then, load balancing process is done in each cluster and the resources for each VM are predicted using DRN [21]. The load balancing is performed by allocating and reallocating the task from the users to the VMs in the cloud based on the resource constraints optimally using MLA [22]. Here, the load balancing method is needed to optimize resources and objectives [23]. Finally, if VMs are overloaded then tasks are assigned to other under loaded VM.

The key contribution of the work is given below:

Proposed MLA: An effective model for balancing load in the fog networking is designed using proposed MLA. Here, the resources of VM are predicted using DRN. Moreover, load balancing is conducted by allocating and reallocating a task based upon the resource constraints by optimally using proposed MLA.

The remaining portion of paper is arranged in following manner; Literature overview of load balancing equivalent to fog computing with its advantages and disadvantages is depicted in Section 2. Section 3 describes an introduced technique for effective optimization of load balancing and the results obtained from the proposed algorithmic method are elucidated in section 4. Lastly, in section 5 research is concluded.

2. Motivation

The core motive of fog computing is to work to increase the effectiveness and eliminate the redundancy involved with data transmission to cloud for both storage and processing. Load balancing in a fog framework supports the equal distribution of resource demand, intending to offer services continuously if the system component fails that is accomplished through provisioning and de-provisioning instances of requests as well optimal resource use. In order to increase application performance and enable efficient network resource use in a fog system, an effective load balancing technique is required due to the wide range of processing and storage resources in fog nodes. The vital requirement for load balancing emerges nowadays owing to the limitations of network resources. Here, the challenges experienced by reviewed techniques are also elucidated, which motivates the researchers to develop an effectual scheme.

2.1. Literature Survey

Literature survey of traditional schemes of balancing load in fogging is explained as follows: Mandeep Kaur., *et al*. [24] designed Fog Computing Architecture of Load Balancing (FOCALB) for the applications of scientific system.

Implementation of this architecture reduced the consumption of energy at fog nodes, execution time and implementation cost. However, it failed to update security and the privacy of the fog nodes. Naranjo, P.G.V., *et al.* [25] developed Fog Computing Architecture Network (FOCAN) to decrease the traffic overheads because delays are not tolerated in healthcare scenarios. Here, the delay was reduced by the fog servers among the cloud and end users, where the distribution of load tends to the efficient use of the resources. During this process, the consumption of power was high and failed to process its function effectively. Mandeep Kaur., *et al.* [26] introduced energy aware load balancing (EA-LB) method for fog networking to ease consumption of energy and to use resources at fog layer. Such technique reduced latency and it enhanced the quality of services (QoS) but, it entailed high implementation cost, time and energy consumption. Mandeep Kaur., *et al.* [15] developed an approach of PSW-Fog clustering-based load balancing scheme to execute overflow datasets. They had reduction in the time delay of fog nodes and also decreased in their computational cost and energy consumption. But it did not accurately improve the performance parameters during experimentation.

Balancing load in fogging has evolved into difficult job because of rise in IoT devices and demands. Fog networking has capacity to give computing resources to end-to-end systems like Internet-of-Things (IoT). Singh, S.P., *et al*. [27] developed Energy-efficient algorithm for load balancing that distributes jobs to optimal Virtual Machines and delivers them as quickly as possible. Jagdeep Singh., *et al.* [28] boost the full-utilization of resources in an SDN-enabled fog system, a safe and energy-conscious fog computing framework was presented, and also load-balancing strategy was also created. Fuzzy Golden Eagle Load Balancing (FGELB), an efficient load balancing strategy, was suggested by Simar Preet Singh., *et al*. [29]. It consisted of 3 stages: setting job priorities, allocating resources according to schedules, and managing power. This scheme is compared with terms of usage of energy, failure level, computational expense, network latency, and average turnaround time, and waiting time and achieves higher performance than other approaches.

To increase the performance of the fogging platform, a load-balancing strategy is used. A. Abuhamdah [30] suggested hybrid method that takes advantage of Optimizing Processing Time (OPT) strategies for minimizing processing time. Proposed algorithm has been compared with existing algorithms with respect to user demands, and data overall cost centre's as well. Results indicates that using the proposed optimize processing time algorithm has superior response and processing time. Mona Albalawi, *et al.,* [31] proposed a fog networking load balancing model using support vector regression (SVR) and the many-objective particle swarm optimization (PSO) technique. Response time, energy usage, resource utilization, and throughput were four variables that the suggested model took into account in order to maximize them while dispersing the load. The results improve the performance of PSO efficiently that balances the load. A. J. Kadhim [32] merged using a parked vehicle as an aid fog computing in a Software Defined Network (SDN) using IoT. This proposed system was more efficient than VANET-Fog-Cloud in terms of average response time, bandwidth consumption, achieving the deadline, and resource usage. The local fog managers and SDN controller constantly try to balance the load both locally and globally. Diverse sample IoT user apps are joined to the fog nodes, when the load assessment problem emerges. Baburao, D., *et al.* [33] proposed an, a particle swarm optimization-focused Enhanced Dynamic Resource Allocation Method (EDRAM) to efficiently handle the load. This proposed method lessened task waiting time, latency and network bandwidth consumption and advanced the Quality of Experience (QoE). G.Shruthi., *et al.* [34] proposed Weighted Greedy Knapsack method for resource allocation but did not consider the balancing load among the nodes.

2.2. Major Challenges

Some of the major issues encountered by traditional approaches of balancing load in fog networking are deliberated as follows:

- Method developed in [35] was not feasible at unplanned functions because of higher propagation time at the time of congestion on network nodes as it depends on the bandwidth, data rate and the circumstances of the network.
- In [25], the technique developed failed to add real-time data processing solutions along with methods of mobile edge computing utilization for making robust models.
- PSW- fog clustering based balancing strategy introduced in [15] failed to progress multi-objective load scheduling problem in a fog cloud networking which was a major drawback.
- The data protection accessed only by the authorized personnel to make safety measures against assaults is the prime challenge in fog computing and also implementing load balancing and scheduling in a real-time context is necessary.

3. Proposed MLA for Load Balancing in Fog Computing

The goal of this approach is to develop a deep learning-based resource prediction and optimization enabled load balancing in fog computing. At first, initialize the fog computing environment with diverse layers namely end user layer, fog layer, and cloud layer. After initializing the fog computing environment, the tasks are submitted from an end user to fog layer, which contains fog nodes that have been divided into diverse clusters. At each cluster, the load balancing process is evaluated and also for every VM the resources are predicted using DRN. However, it depends upon the workload in the network and the load balancing is performed by allocating and reallocating the task from the users to the VMs in the cloud based on the resources constraints optimally using MLA [37-40], such that it required balancing load strategy for the optimization of resources by considering the objectives like predicted resources, energy consumption, cost, and execution time. Lastly, if VMs are overloaded, then the tasks are allocated from the consistent VM and assigned to underloaded VM. Fig. 1 reveals a diagrammatic view of proposed MLA for balancing load in fog networking.

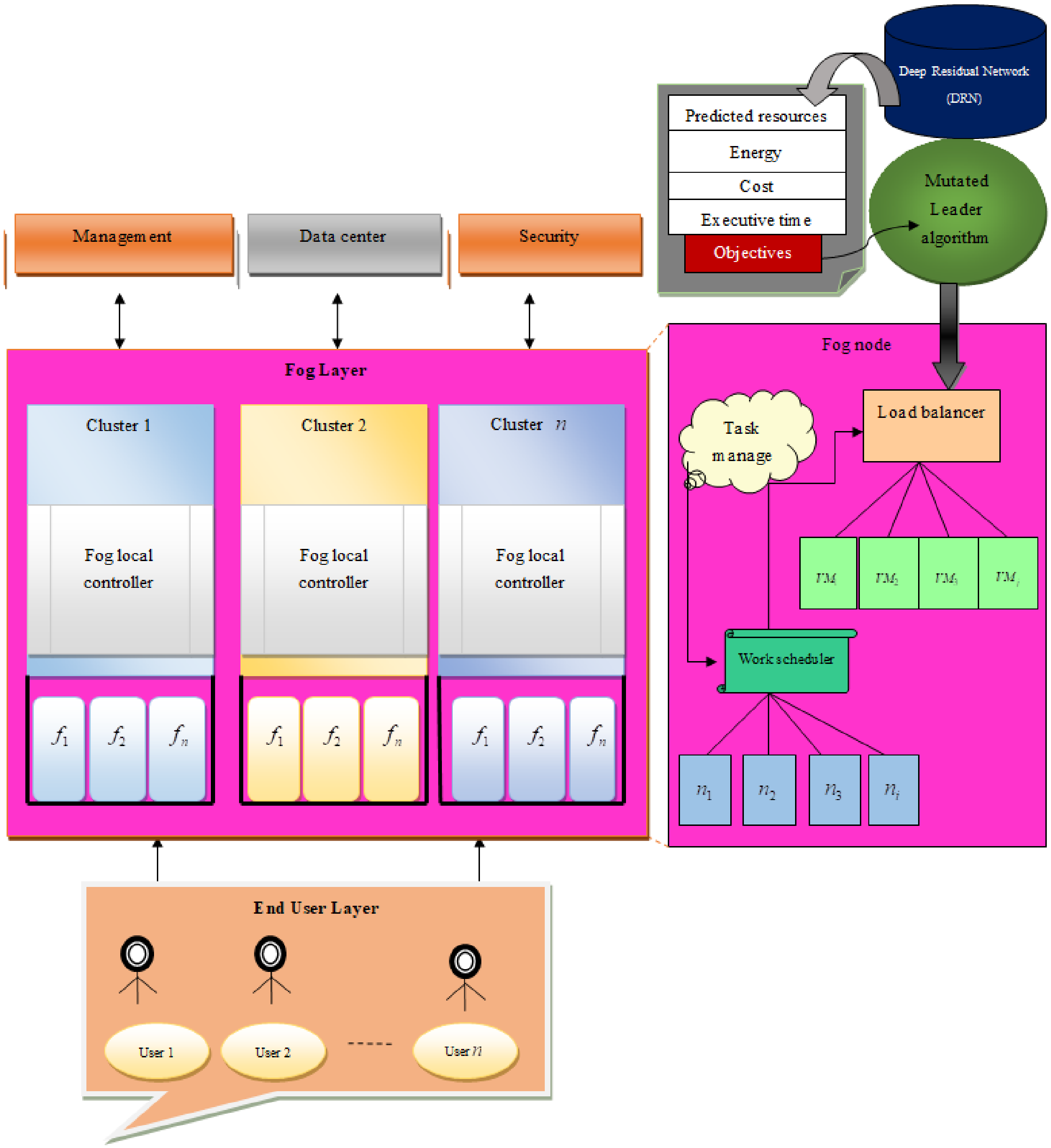


Fig.1. Schematic Block diagram of proposed ML

3.1. Resource Prediction Using DRN

Resource prediction is carried out for every VM, which depends on the workload in a network utilizing DRN.

Architecture of DRN

Architecture of single residual unit is not complicated [36], whereas in residual unit the output is given by $R(i)$, where i indicates an input matrix of residual unit structure. Fig. 2 represents architecture of DRN. $D_q$is the input given to DRN. Theoretically, the fundamental mappings are stacked by means of numerous hidden layers that can fit to any functions, so $F(i) = R(i) - i$ can be fitted in natural manner. Formally, $R(i)$ indicates the desired underlying mapping and the non-linear layers that are stacked fits another mapping of $F(i) = R(i) - i$. The actual mapping is re-cast to $R(i) + i$. Each of the residual unit has two layers, i represents input whereas T_1 and T_2 are the weight matrices.

An expression of residual unit is given beneath whereas α denotes the ReLU activation function.

$$F(i) = T_2\alpha(T_1 i) \tag{1}$$

Afterwards another ReLU activation and shortcut connection function, the residual unit output obtained is represented by,

$$u = F(i, T_x) + i, \tag{2}$$

Where, $F(i,T_x)$ indicates the residual function, which requires to be tuned whereas T_x is x^{th} weight matrix in the stacked hidden layers. T_x can be T_1 and T_2. The linear mapping $T_s i$ could be added to the shortcut connection for making a count of input channels equivalent to an output, where, T_s denotes weight matrix in the shortcut connection. A definite formulation is presented by,

$$R(i) = F(i, T_x) + T_s i \tag{3}$$

Where, $R(i)$ is the output of the residual unit. Thus, predicted output attained from DRN is represented by P_q.

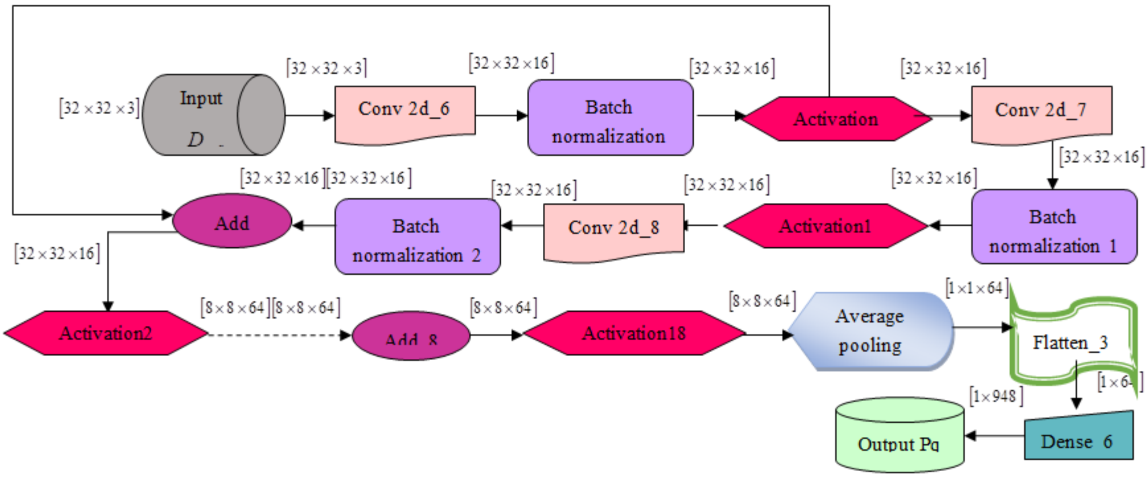


Fig.2. Architecture of DRN

3.2. *Task Assignment in Fog Nodes*

In each cluster, load balancing is acted upon and the fog cluster local controller monitor load distribution in VM. Depending on the load, the tasks are allocated and reallocated optimally utilizing the optimization.

A. *Solution Encoding*

In solution encoding, the tasks performed by VM are evaluated and then under loaded VMs are assigned to the overloaded VM. The solution encoding is delineated in Fig. 3.

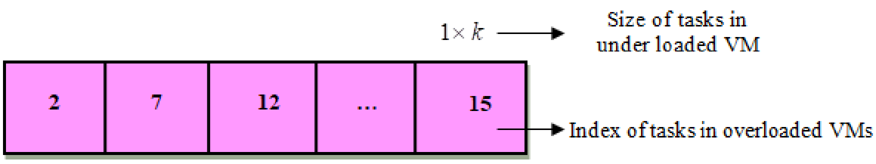


Fig.3. Solution encoding

B. *Fitness Function*

It is utilized for assessing the best solution that is computed based upon execution time, cost, predicted output and energy. The fitness function is formulated by,

$$\eta = T + C + P_q + E \tag{4}$$

Where, T is the execution time, C denotes the cost and E signifies the energy. An execution time is given by,

$$T = \sum_{z=1}^{n} T_t + \sum_{z=1}^{n} T_p + \sum_{z=1}^{n} T_w \tag{5}$$

Here, T_t represents the receiving time of task whereas T_p and T_w are the processing time and waiting time of task. The cost can be calculated as follows.

$$C = \frac{M_f + H_f}{2} \tag{6}$$

Where, M_f and H_f symbolizes movement factor and cost factor. The movement factor can be computed by,

$$M_f = \frac{1}{N_o} \sum_{z=1}^{n} \frac{N_m}{N_v} \tag{7}$$

Where,N_o, N_m and $N_v$denotes number of frosts in datacenter, migrations and VM utilized. The cost factor is formulated as given below,

$$H_f = \sum_{z=1}^{n} \frac{P * M}{VM * A} \tag{8}$$

Here, P indicates the cost to process, M represents the memory of task, A symbolizes datacenter and n is the overall number of VM in a setup. The Energy can be represented as,

$$E = \sum_{z=1}^{n} \left(T + M_f + H_f\right) \tag{9}$$

3.3. Proposed MLA for Load Balancing

MLA is the population-based technique, which is a repetition based procedure capable in suggesting suitable quasi-optimum solution to optimization issues. The solutions are updated underneath a leadership of mutated leaders in the search space.

A. Initialization

Initialize members of MLA population, where each of the members in population is vector and the values represent problem variable values.

$$K = \{K_1, K_2, \dots, K_r, \dots, K_h\} \tag{10}$$

Where K_r represents the r^{th} solution whereas h is the overall count of solutions.

B. Evaluate Objective Function

The finest solution is evaluated for balancing load in a fog networking and is computed utilizing Eq. (4).

C. Create Mutated Leader

After calculation of objective function, the best as well as worst members of population are founded. The solutions in mutated leader are selected randomly from a finest member of population, worst member of population and also, ordinary members of population. A mutated leader is framed for individual members of population based on the equation represented below.

$$G_r : g_{r,y} = \begin{cases} a_{best_y}, b \le B_{best}; \\ a_{worst_y}, B_{best} < b \le B_{best} + B_{worst}; \\ a_{j,y}, else, \end{cases} \tag{11}$$

Here, G represents the mutated leader for guiding r^{th} population member in the search space and $g_{r,y}$ indicates the y^{th} dimension of it. a_{best} and a_{worst} are the best as well as worst population members. $a_{j,y}$ symbolizes the y^{th} dimension of j^{th} population member that is chosen randomly. b denotes the random number whereas B_{best} and B_{worst} are the probability of best and worst member that is being elected.

D. Calculate the New Status

A mutated leader is generated for individual member of the population for each iteration. Later devising mutated leader, every member of population is updated in the search space based upon own leader supervision. The expression is illustrated by,

$$K_r{}^{new}: a_{r,y}{}^{new} = \begin{cases} a_{r,y} + \gamma \times \left(g_{r,y} - X \times a_{r,y}\right), \eta_r{}^G < \eta_r; \\ a_{r,y} + \gamma \times \left(a_{r,y} - X \times g_{r,y}\right), else \end{cases} \tag{12}$$

$$K_r = \begin{cases} K_r{}^{new}, \eta_r{}^{new} < \eta_r; \\ K_{r,} else, \end{cases} \tag{13}$$

$$X = Round(1 + rand) \tag{14}$$

Here, $K_r{}^{new}$ represents the new status of r^{th} population member, $a_{r,y}{}^{new}$ indicates the y^{th} dimension of it. $\eta_r{}^{new}$ signifies the objective function whereas γ is the random number ranging from 0 to 1. $\eta_r{}^G$ symbolizes mutated leader objective function value for r^{th} population member and X denotes a random number that may be 1 or 2.

E. Termination

To achieve finest solution above steps are done continuously. Algorithm 1, describes the pseudocode of MLA

Algorithm 1: MLA

Input: MLA population $K = \{K_1, K_2, \dots, K_r, \dots, K_h\}$
Output: Best solution

```
Set h
Initialization of population
Evaluating objective function utilizing Eq. (4)
Update a_best and a_worst
for r = 1: h
      Create mutated leader G_r Eq. (11)
      Calculate K_r^new Eq. (12) and (14)
      Update K_r utilizing Eq.(13)
End for
Best attained solution
End
```

4. Results and Discussions

This portion expounds the outcomes and discussion of newly devised MLA for load balancing in fog computing is done.

4.1. Experimental Setup

An experimentation setup of MLA is conducted in Python tool with iFogSim simulator in a PC comprising of Intel core-i3 processor, windows 10 OS and 4 GB RAM.

4.2. Evaluation Measures

The performance of MLA is explored employing an evaluation measures like execution time, cost and load.

A. Execution Time

The overall time required to execute a task for balancing load in fog networking is known as execution time. Formula for execution time is already illustrated in Eq. (5)

B. Cost

It is defined as an overall time necessary to finish a whole operation for balancing load in a fog networking. It can be calculated utilizing Eq. (6)

C. Load

It is referred to as an amount of load carried by the VM, which can be formulated utilizing the equation given below.

$$L = \frac{1}{N} \sum_{z=1}^{n} \left[\frac{V - (V_t - U) + S}{V}\right] \tag{15}$$

Here, V indicates the size of VM, V_t denotes the total dimension of VM and U indicates the free space of VM.

N symbolizes a count of task and S is the size of task.

4.3. Comparative Techniques

An evaluation of MLA considering the performance measures by comparing with classical techniques like FOCALB [24], FOCAN [25], EA-LB [26], PSW-fog [15] clustering with proposed MLA is done.

4.4. Comparative Analysis

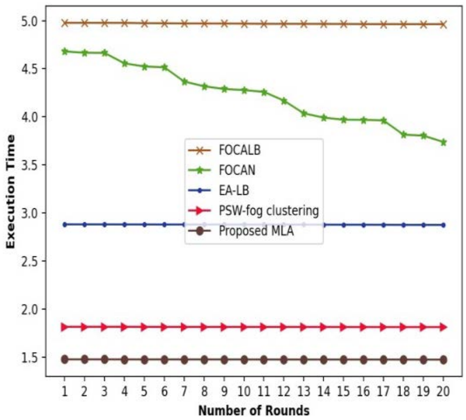
A comparable evaluation of proposed MLA is performed with respect to measures of performance by varying the number of rounds, utilizing number of tasks 100 and 200.

A. Analysis Based Upon Number of Tasks=100

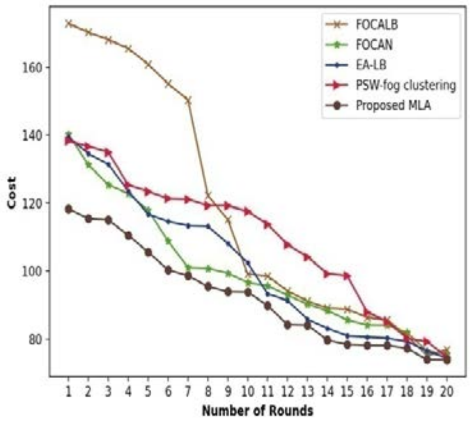
Fig. 4. Illustrates a comparable estimation of proposed MLA with respective to performance metrics by altering number of rounds for the number of tasks=100. An assessment of MLA based upon execution time is elucidated in Fig. 4 a). Execution time calibrated for round=20 by MLA is 1.472 whereas FOCALB is 4.964, FOCAN is 3.738, EA-LB is 2.874 and PSW-fog clustering is 1.811 respectively. Fig. 4 b) demonstrates an evaluation of proposed MLA based upon cost. For round=20, the cost calibrated by proposed MLA is 73.875 while other methods like FOCALB, FOCAN. EA-LB and PSW-fog clustering are76.798, 75.482, 74.420 and 74.665respectively. An evaluation of load is depicted in Fig. 4 c). MLA achieved the load of 0.0126 while considering round=20. Likewise, existing techniques like FOCALB, FOCAN. EA-LB and PSW-fog clustering obtained values of 0.490, 0.398, 0.043 and 0.042 respectively.

B. Analysis Based upon Number of Tasks=200

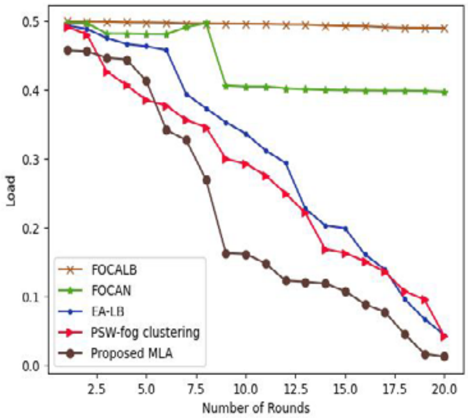
Fig. 5 depicts a comparable estimation of proposed MLA with respective to performance metrics by altering number of rounds for the number of tasks=200. An assessment of MLA based upon execution time is elucidated in Fig. 5 a). Execution time calibrated for round=20 by MLA is 1.472 whereas FOCALB is 4.965, FOCAN is 3.761, EA-LB is 2.874 and PSW-fog clustering is 1.811 respectively. Fig. 5 b) show an evaluation of proposed MLA based upon cost. For round=20, the cost calibrated by proposed MLA is 69.448 while other methods like FOCALB, FOCAN. EA-LB and PSW-fog clustering are91.952, 72.562, 74.822 and 72.571 respectively. An evaluation of load is depicted in Fig. 5 c). MLA achieved the load of 0.0003 while considering round=20. Likewise, existing techniques like FOCALB, FOCAN, EA-LB and PSW-fog clustering obtained 0.4030, 0.1988, 0.0150 and 0.0080 respectively.



(a)

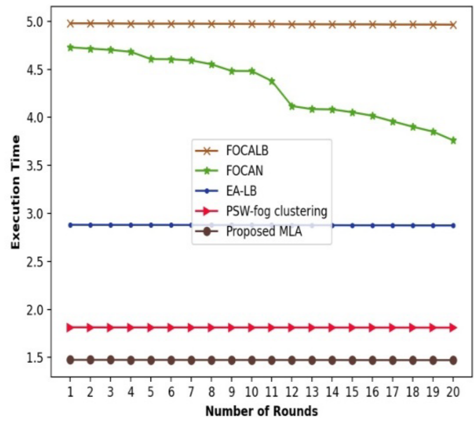


(b)

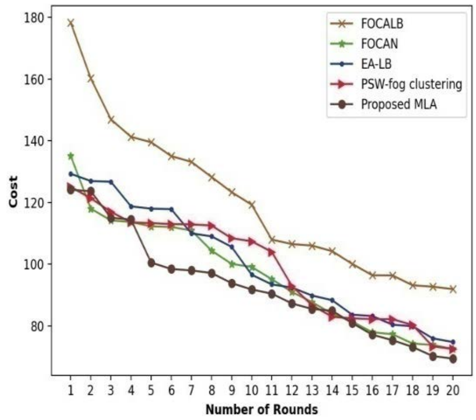


(c)

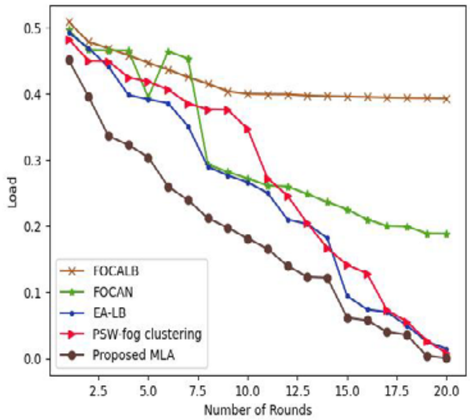
Fig.4. Comparative estimation of proposed MLA a) Execution time, b) Cost, c) Load



(a)



(b)



(c)

Fig.5. Comparative estimation of proposed MLA a) Execution time, b) Cost, c) Load

4.5. *Comparative Discussion*

Comparing of deliberation of introduced MLA is expounded in Table 1. From below shown table, it is acceptable that the proposed MLA calibrated minimum execution time is 1.472, minimum cost is 69.448 and minimum load is 0.0003 for number of rounds is 20 with number of tasks=200.

Table 1. Comparative discussion of proposed MLA

Number of tasks	Metrics/Methods	FOCALB	FOCAN	EA-LB	PSW-fog clustering	Proposed MLA
100	Execution time(ns)	4.964	3.738	2.874	1.811	**1.472**
	Cost($)	76.798	75.482	74.420	74.665	73.875
	Load(%)	0.490	0.398	0.043	0.042	0.0126
200	Execution time(ns)	4.965	3.761	2.874	1.811	1.472
	Cost($)	91.952	72.562	74.822	72.571	**69.448**
	Load(%)	0.4030	0.1988	0.0150	0.0080	**0.0003**

5. Conclusions

In fog computing, the load balancing dispenses much across obtainable resources every node. It is utilized for getting highly satisfied users and high resource usage, guarantees that no individual node is overwhelmed, thereby improving the total performance of system. Therefore, proposed MLA is introduced for balancing load in a fog networking. Firstly, fogging is initialized with fog layer, cloud layer and end user layer, then, the task is submitted from end user under fog layer with cluster of nodes. Thereafter, load balancing process is conducted, where cluster and the resources for individual VM is predicted utilizing DRN. Depending on the workload in cloud, the load balancing is conducted by allocating and reallocating the task from users to VMs in a cloud on basis of resource constraints optimally utilizing Mutated Leader Algorithm (MLA). Here, the load balancing method is essential for optimizing resources and the objectives. Lastly, if VMs are overloaded, then tasks are obtained from equivalent VM and allocated to an under loaded VM. Proposed approach has been reduced in the range of 18.71% to 70.34% in execution time for 100 tasks and 200 tasks when it compared to existing strategies. Execution cost has been reduced in the range of 1.05%

to 3.80% for 100 tasks and 4.30% to 24.47% for 200 tasks when it compared to existing strategies. Balancing the load has been improved in the range of 70.00% to 97.42% for 100 tasks and 98.00% to 99.92% for 200 tasks when it compared to existing strategies. Hence, the proposed MLA achieved minimal execution time of 1.472ns, cost of $69.448 and load 0.0003% respectively. In prospective, the finest load balancing technique will be implemented for decreasing cost and consumption of energy in fog environment.

References

[1] Devagnanam J,Elango N M, "Optimal Resource Allocation of Cluster using Hybrid Grey Wolf and Cuckoo Search Algorithm in Cloud Computing", Journal of Networking and Communication Systems, vol.3, no.1, pp.31-40, 2020.

[2] Asif Ali Laghari, Awais Khan Jumani and Rashid Ali Laghari,R "Review and state of art of fog computing", Archives of Computational Methods in Engineering, vol.28, no.5, pp.3631-3643, 2021.

[3] Xavi Masip-Bruin, Eva Marín-Tordera, Ghazal Tashakor, Admela Jukan and Guang-Jie Ren, "Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems", IEEE Wireless Communications, vol.23, no.5, pp.120-128, 2016.

[4] Vishal Kumar, Asif Ali Laghari, Shahid Karim, Muhammad Shakir and Ali Anwar Brohi, "Comparison of fog computing & cloud computing", International Journal of Mathematical Sciences and Computing, vol.1, pp.31-41, 2019.

[5] Valeria Cardellini, Gabriele Mencagli, Domenico Talia and Massimo Torquati, "New landscapes of the data stream processing in the era of fog computing", Future Generation Computer Systems, vol.99, pp.646-650, 2019.

[6] Cheng Huang, Rongxing Lu, and Kim-Kwang Raymond Choo, "Vehicular fog computing: architecture, use case, and security and forensic challenges", IEEE Communications Magazine, vol.55, no.11, pp.105-111, 2017.

[7] Michaela Iorga, Larry Feldman, Robert Barton, Michael J. Martin, Nedim Goren and f Mahmoudi, "Fog computing conceptual model", National Institute of Standards and Technology Special Publication, 2018.

[8] Mohammad Wazid, Ashok Kumar Das,Rasheed Hussain, Giancarlo Succi and Joel J. P. C. Rodrigues, "Authentication in cloud-driven IoT-based big data environment: Survey and outlook", Journal of systems architecture, vol. 97, pp.185-196, 2019.

[9] Yang Liu, Jonathan E.Fieldsend and Geyong Min, G. "A framework of fog computing: Architecture, challenges, and optimization", IEEE Access, vol.5, pp.25445-25454, 2017.

[10] Ashok Kumar C,Vimala R, "Load Balancing in Cloud Environment Exploiting Hybridization of Chicken Swarm and Enhanced Raven Roosting Optimization Algorithm", Multimedia Research, vol.3, no.1, pp.45-55, 2020.

[11] Mohammad Riyaz Belgaum, Shahrulniza Musa, Muhammad Mansoor Alam and mazliham Mohd Suud, "A systematic review of load balancing techniques in software-defined networking" IEEE Access, vol. 8, pp.98612-98636, 2020.

[12] Changlong Li, Hang Zhuang, Qingfeng Wang and Xuehai Zhou, "SSLB: self-similarity-based load balancing for large-scale fog computing", Arabian Journal for Science and Engineering, vol.43, no.12, pp.7487-7498, 2018.

[13] Abuhamdah, A and Al-Shabi, M, "Hybrid Load Balancing Algorithm For Fog Computing Environment". International Journal of Software Engineering and Computer Systems, vol. 8, no.1, pp.11-21, 2022.

[14] Muhammad Junaid, Adnan Sohail, Rao naveed Bin Rais, Adeel Ahmed, Osman Khalid, Imran Ali Khan, Syed Sajid Hussain and Naveed Ejaz, "Modeling an optimized approach for load balancing in cloud", IEEE Access, vol.8, pp.173208-173226, 2020.

[15] Mandeep Kaur1 and Rajni Aron, "An Energy-Efficient Load Balancing Approach for Scientific Workflows in Fog Computing", Wireless Personal Communications, pp.1-25, 2022.

[16] Daniel E. Eisenbud, Cheng Yi, Carlo Contavalli, Cody Smith, Roman Kononov, Eric Mann-Hielscher, Ardas Cilingiroglu, Bin Cheyney,Wentao Shang and Jinnah Dylan Hosein, "Maglev: A fast and reliable software network load balancer", In proceedings of 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16), pp. 523-535, 2016.

[17] Sambit Kumar Mishra, Bibhudatta Sahoo and Priti Paramita Parida, "Load balancing in cloud computing: a big picture", Journal of King Saud University-Computer and Information Sciences, vol.32, no.2, pp.149-158, 2020.

[18] Hamza Sulimani, Wael Y.Alghamdi, Tony Jan, Gnana Bharathy and Mukesh Prasad, "Sustainability of Load Balancing Techniques in Fog Computing Environment", Procedia Computer Science, vol.191, pp.93-101, 2021.

[19] Ibrahim Mahmood Ibrahim, Siddeeq Y. Ameen, Hajar Maseeh Yasin, Naaman Omar, Shakir Fattah Kak, Zryan Najat Rashid, Azar Abid Salih, Nareen O. M. Salim and Dindar Mikaeel Ahmed, "Web server performance improvement using dynamic load balancing techniques: A review", system, vol.19, p.21, 2021.

[20] Gopala Krishna Sriram, "Challenges of cloud compute load balancing algorithms. International Research Journal of Modernization in Engineering Technology and Science, vol.4, no.1, pp.1186-1190, 2022.

[21] Zhicong Chen, Yixiang Chen, Lijun Wu, Shuying Cheng and Peijie Lin, "Deep residual network based fault detection and diagnosis of photovoltaic arrays using current-voltage curves and ambient conditions", Energy Conversion and Management, vol.198, pp.111793, 2019.

[22] Fatemeh Ahmadi Zeidabadi, Sajjad Amiri Doumari, Mohammad Dehghani, Zeinab Montazeri, Pavel Trojovský and Gaurav Dhiman, "MLA: a new mutated leader algorithm for solving optimization problems", Computers, Materials & Continua, vol.70, no.3, pp.5631-5649, 2022.

[23] G. Shruthi, Monica R. Mundada, B. J. Sowmya, S. Supreeth, "Mayfly Taylor Optimisation-Based Scheduling Algorithm with Deep Reinforcement Learning for Dynamic Scheduling in Fog-Cloud Computing", Applied Computational Intelligence and Soft Computing, vol. 2022, Article ID 2131699, 17 pages, 2022. https://doi.org/10.1155/2022/2131699.

[24] Mandeep Kaur and Rajni Aron, "FOCALB: Fog Computing Architecture of Load Balancing for Scientific Workflow Applications", Journal of Grid Computing, vol.19, no.4, pp.1-22, 2021.

[25] Naranjo, P.G.V., Pooranian, Z., Shojafar, M., Conti, M. and Buyya, R., 2019. FOCAN: A Fog-supported smart city network architecture for management of applications in the Internet of Everything environments. Journal of parallel and distributed computing, 132, pp.274-283.

[26] Mandeep Kaur and Rajni Aron "Energy-aware load balancing in fog cloud computing", Materials Today: Proceedings, 2020.

[27] Singh, S.P., Kumar, R., Sharma, A. et al. Energy efficient load balancing hybrid priority assigned laxity algorithm in fog computing. Cluster Comput 25, 3325–3342 (2022). https://doi.org/10.1007/s10586-022-03554-x.

[28] Singh, Jagdeep, Parminder Singh, El Mehdi Amhoud, and Mustapha Hedabou. 2022. "Energy-Efficient and Secure Load Balancing Technique for SDN-Enabled Fog Computing" *Sustainability* 14, no. 19: 12951. https://doi.org/10.3390/su141912951.

[29] Simar Preet Singh, "Effective load balancing strategy using fuzzy golden eagle optimization in fog computing environment", Sustainable Computing: Informatics and Systems, Volume 35, 2022, 100766, ISSN 2210-5379, https://doi.org/10.1016/j.suscom.2022.100766.

[30] Abuhamdah, A., & Al-Shabi, M. (2022). HYBRID LOAD BALANCING ALGORITHM FOR FOG COMPUTING ENVIRONMENT. International Journal of Software Engineering and Computer Systems, 8(1), 11–21. https://doi.org/10.15282/ijsecs.8.1.2022.2.0092.

[31] Albalawi, Mona, Entisar Alkayal, Ahmed Barnawi, and Mehrez Boulares. "Load Balancing Based on Many-objective Particle Swarm Optimization Algorithm with Support Vector Regression in Fog Computing." Journal of Engineering and Applied Sciences Technology. SRC/JEAST-170. DOI: doi. org/10.47363/JEAST/2022 (4) 138 (2022).

[32] A. J. Kadhim and J. I. Naser, "Proactive load balancing mechanism for fog computing supported by parked vehicles in IoV-SDN," in China Communications, vol. 18, no. 2, pp. 271-289, Feb. 2021, doi: 10.23919/JCC.2021.02.019.

[33] Baburao, D., Pavankumar, T. & Prabhu, C.S.R. Load balancing in the fog nodes using particle swarm optimization-based enhanced dynamic resource allocation method. Appl Nanosci (2021). https://doi.org/10.1007/s13204-021-01970-w.

[34] G., Shruthi, Monica R. Mundada, and Supreeth S. 2022. "The Resource Allocation Using Weighted Greedy Knapsack Based Algorithm in an Educational Fog Computing Environment". International Journal of Emerging Technologies in Learning (iJET) 17 (18): pp. 261-274. https://doi.org/10.3991/ijet.v17i18.32363.

[35] Hasan Ali Khattak, Hafsa Arshad, Saif ul Islam, Ghufran Ahmed, Sohail Jabbar, Abdullahi Mohamud Sharif and Shehzad Khalid, "Utilization and load balancing in fog servers for health applications", EURASIP Journal on Wireless Communications and Networking, vol.2019, no.1, pp.1-12, 2019.

[36] Min Xia, Wanan Liu, Bicheng Shi, Liguo Weng and Jia Liu, "Cloud/snow recognition for multispectral satellite imagery based on a multidimensional deep residual network", International journal of remote sensing, vol.40, no.1, pp.156-170, 2019.

[37] S. Supreeth, K. Patil, S. D. Patil, S. Rohith, Y. Vishwanath, and K. S. V. Prasad, "An Efficient Policy-Based Scheduling and Allocation of Virtual Machines in Cloud Computing Environment," Journal of Electrical and Computer Engineering, vol. 2022. Hindawi Limited, pp. 1–12, Sep. 24, 2022. doi: 10.1155/2022/5889948.

[38] S. S and Kirankumari Patil, "Hybrid Genetic Algorithm and Modified-Particle Swarm Optimization Algorithm (GA-MPSO) for Predicting Scheduling Virtual Machines in Educational Cloud Platforms," International Journal of Emerging Technologies in Learning (iJET), vol. 17, no. 07. International Association of Online Engineering (IAOE), pp. 208–225, Apr. 12, 2022. doi: 10.3991/ijet.v17i07.29223.

[39] S. Supreeth, K. Patil, S. D. Patil, and S. Rohith, "Comparative approach for VM Scheduling using Modified Particle Swarm Optimization and Genetic Algorithm in Cloud Computing," 2022 IEEE International Conference on Data Science and Information System (ICDSIS). IEEE, Jul. 29, 2022. doi: 10.1109/icdsis55133.2022.9915907.

[40] Supreeth S, Kirankumari Patil, "VM Scheduling for Efficient Dynamically Migrated Virtual Machines (VMS-EDMVM) in Cloud Computing Environment," KSII Transactions on Internet and Information Systems, vol. 16, no. 6. Korean Society for Internet Information (KSII), Jun. 30, 2022. doi: 10.3837/tiis.2022.06.007.

Authors' Profiles

Shruthi G. holds M. Tech. degree in Computer Science and Engineering from Visvesvaraya Technological University (VTU), Belgaum, and currently she is pursuing Doctoral program (Ph. D.) in VTU and working as an Assistant Professor in School of Computer Science and Engineering, REVA University, Bengaluru, India. Her areas of interest include Cloud computing, Fog Computing, Machine Learning and Data Science.

Dr. Monica R. Mundada is working as a Professor in Computer Science Department of Ramaiah Institute of Technology. Her areas of interest include theory of computation, wireless sensor networks, computer networks, data communications, and IOT.

Dr. Supreeth S. holds Ph. D degree in Computer Science and Engineering from Visvesvaraya Technological University, Belgaum, India. He has been a technology educator for ten years, teaching various subjects like Theory of Computation, Virtualization and Cloud Computing, Big Data and Data analytics, Image Processing. He has published many papers in international journals and conferences in the domain of Cloud, Fog Computing and in Image Processing. He is a member of IEEE & Life-Time member of CSI, ISTE, IAENG.

Dr. Bryan Gardiner received a first-class honours degree in Electronics and Computer Systems in 2006, and a Ph.D. degree from Ulster University in 2010. A Fellow of the Higher Education Academy, Bryan is currently a Senior Lecturer and Associate Head of School of Computing, Engineering and Intelligent Systems at Ulster University, Northern Ireland. An active member of the Cognitive Robotics team within the Intelligent Systems Research Centre, his research interests are primarily in computer vision, data analytics, and mobile robotics. He has served as a reviewer for numerous international conferences and journals and is a member of the IEEE UKRI society, IEEE Signal Processing Society (SPS), Affiliate member of Computational Imaging SIG, Irish Pattern Recognition & Classification Society, International Association of Pattern Recognition and British Machine Vision Association.

How to cite this paper: Shruthi G., Monica R. Mundada, S. Supreeth, Bryan Gardiner, "Deep Learning-based Resource Prediction and Mutated Leader Algorithm Enabled Load Balancing in Fog Computing", International Journal of Computer Network and Information Security(IJCNIS), Vol.15, No.4, pp.84-95, 2023. DOI:10.5815/ijcnis.2023.04.08