International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Fog-Cloud: Efficient Task Scheduling Mechanism for Load Balancing Technique Using KHA – Task Scheduling Algorithm

Archana R.[1], K. Pradeep Mohan Kumar*[2]

Submitted: 09/01/2024 **Revised**: 15/02/2024 **Accepted**: 23/02/2024

Abstract: Fog-Cloud is a computing paradigm. It can able to process, manage and Storage the data virtually. This enhances cloud computing by decentralizing its capabilities to the fog nodes. These fog nodes, positioned near the network's edge, the diverse storage capacity, computation, and processing capabilities. They can analyze requirements and respond to emergencies, processing and executing tasks. The restricted computational power of fog nodes presents a significant challenge in efficiently scheduling tasks among different fog nodes within their deadlines, while minimizing delays, costs, and energy consumption. Task Scheduling is the more crucial issue in fog computing. An Effective Task Scheduling Mechanism will improve the efficiency and Quality of the Service (QoS) in the computing Paradigm. The resource in the Fog Computing is lesser when compare to the Cloud. So it's necessary to enhance the performance of the task scheduler. The proposed the dynamical scheduling task method for dynamically pairing tasks with resources comprises three essential components: a recourse availability check, and a priority method. The batch system effectively utilizes resource capacity and priority methods to match tasks with fog nodes. A notable benefit of this approach is the reduction of the search space achieved through batch processing. This technique has been executed using iFogSim and cloudlet for generating the workload with the current state-of-the-art. The comparative assessment involved various quality parameters, including delay, execution time, energy consumption, etc., revealing an average enhancement of 37.3%.

Keywords: *Task Scheduling, Dynamic scheduling, Offloading, Energy consumption, Latency*

1. Introduction

Recent innovations in technological developments, such as the Internet of Things (IoT), required resource-based processing to run a wide range of real-time systems [1,24]. Due to numerous technological advancements, individuals are utilizing mobile gadgets for a wide variety of purposes in addition to making voice communication. About any task that can be completed on a computer is being envisioned as being able to be done on a device [2]. Cloud services and applications experience undesirable delays when transmitting data to and from end devices due to the proximity of cloud data centres to the network core. Consequently, the cloud proves unsuitable for real-time applications like live monitoring. Furthermore, certain IoT applications often necessitate geographical distribution, support for mobility, and awareness of location. Additionally, the cloud's centralized nature lacks inherent location awareness [3]. A pivotal characteristic of WSNs involves their capability to extend battery life by predominantly operating at low power levels. Actuators [27] play a crucial role in orchestrating measurement operations and influencing behaviour, thereby forming a closed-loop system. These actuators can be conceptualized as fog nodes that execute various actions to regulate end devices equipped with sensors. Such WSNs demand minimal bandwidth, consume lower energy, and operate with very limited processing power [4].

Nonetheless, existing load balancing algorithms may experience diminished efficacy in the dynamic fog environment, necessitating an algorithm compatible with such fluctuations. Thus, this study proposes a decision-making approach rooted in Reinforcement Learning (RL) to discern low-load nodes. To mitigate the likelihood of overload and minimize processing time, the proposed solution empowers fog nodes to offload input tasks by selecting an available neighbouring fog node[5]. Through the application of PPSO, the collective processing time for all incoming loads is decreased. This scheduling algorithm is introduced with the aim of enhancing various parameters related to load balancing within a cloud environment, accomplishing improvements within a shorter timeframe compared to other widely adopted scheduling algorithm [6]. The proposed scheduling model is specifically crafted for executing cloud computing applications and operates through three distinct phases: classification, execution, and minimization, resulting in output parameters such as completion time, average waiting and turnaround time, and render duration.

The execution times of tasks in cloud computing applications are generated within the task scheduling model using both exponential and normal distributions. Task

[1] *S.R.M Institute of Science &Tech, Department of Computing Technologies, Kattankulathur– 603202, Chennai.*
[2] *S.R.M Institute of Science &Tech, Department of Computing Technologies, Kattankulathur– 603202, Chennai.*
** Corresponding Author Email: ar6036@srmist.edu.in*

prioritization is influenced by the shortest strategy for the First Job, and the outcomes are benchmarked against alternative ranking methods such as "Largest Processing Time" and "First Come First Serve." The suggested scheduling model consistently yields more favourable results across the specified performance parameters [7]. The task of highest priority is selected from the waiting queue, following the principles of the Fibonacci heap. We have introduced a parallel algorithm for task scheduling, wherein task priority assignment and heap construction are concurrently executed, addressing both pre-emptive and non-pre-emptive scheduling approaches [8]. The inclusion of priority is recognized as a pivotal aspect in the task scheduling process. Priority is calculated based on key parameters essential for meeting user requirements and enhancing the resource utilization in the fog environment. We present a novel Dynamic Priority-Queue (DPQ) method employing a hybrid multi-criteria decision-making (MCDM) approach, specifically ELECTRE III and Differential Evolution (DE). Additionally, for task scheduling, we introduce a hybrid metaheuristic algorithm incorporating Particle Swarm Optimization (PSO) and Simulated Annealing (SA). The proposed DEELDPQ-SAPSO approach is validated using the Cloud Sim simulator [1]. Experimental results demonstrate the effectiveness of the proposed approach in achieving strong performance, user prioritization, load balancing, and enhanced resource utilization [8].

The PR_BTSKMC approach, as proposed, prioritizes and ranks tasks by treating task length as the sole attribute, utilizing the K-Means Clustering algorithm. Following the prioritization and ranking of tasks, they are allocated to the most suitable Virtual Machines (VMs) based on the CPU capacity of the VMs. Consequently, PR_BTSKMC generates a practical schedule that enhances various scheduling metrics when compared to the existing OPFT scheduling method. Grounded in the belief that K-Means clustering can form efficient clusters and is simpler than alternative clustering algorithms, it is applied to cluster tasks based on a singular attribute, specifically the task length. The proposed methods outperform the existing OPFT algorithm in fulfilling scheduling objectives [9]. Initially, a rule-based reactive approach is used to create a training dataset and auto-scale. [2,10]. By thoroughly examine these research works still latency, scheduling the task, energy efficiency need to be make effective. These issues need to be addressed in order to state-of-the art in the proposed method. Thus the research endeavours is aimed to improve the efficiency of scheduling the task in the fog computing environment.

The main contribution of this proposed works follows:

- Intelligent-based task scheduling methods have been suggested to perform load balancing by considering the task priority and the resource capacity.
- Designed an central controller to monitor the resource capacity of the each node dynamically using optimization algorithm.
- Analysing the each task criticality to find the priority task based on the time critical task.
- Discussed the several issues and challenges and demonstrate how this work improved the efficiency than traditional method.

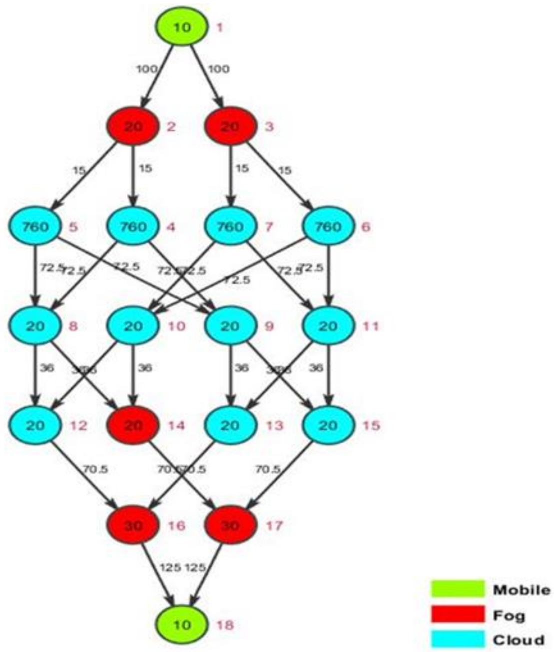

Fig 1: System Architecture of the three level environments associated with each other.

2. Literature Survey

M.K. Hussain [11] presented a IoT based application in Fog environment by using the method called Ant-Colony Algorithm to offload the task efficiently. This technique is used to improve the Quality of service by reducing the latency. Based on the parameters of communication cost and Response time the experimental results are analyzed. Additionally, effectiveness of the various algorithms are compared such as Round robin and Particle swam optimizer with Ant Colony Algorithm.

Kishore [12]Proposed a meta-heuristic scheduler with smart Ant-colony algorithm to improve the response time while offloading the task in a Edge Computing Environment by modified particle swarm optimization algorithm and numerical results are compared with the existing techniques.

Abdel [13] presents the multi-objective approach for scheduling the task in the Fog Computing Environment. Polynomial mutation mechanism is integrated with marine predator's algorithm. Based on the optimization process the results are calculated with the performance metrics are makespan, energy, corbon emission and Flow time.

Vishalika[14] It entailed giving jobs to the selection methods, or the virtual machine with the least amount of demand, to ensure effective resource consumption. To test the load balancing effectiveness of the proposed method by making the best use of the available resources, simulations were run to modify the number of virtual machines. In the test results, the proposed strategy chose the virtual machines with the lowest load in all settings and enhanced resource utilisation. The proposed approach had demonstrated its suitability in VM load balancing to prevent cloud systems from becoming overloaded.

A.J. Kadhim [15] proposed a system operates proactively to achieve load balancing at both local and global levels, controlled, in turn, by an SDN controller and local fog managers. Simulation experiments show that the suggested system outperforms the VANET-Fog-Cloud and IoV-Fog-Cloud frameworks in terms of efficiency in a number of parameters, such as average response time, bandwidth consumption %, fulfilling deadlines, and resource utilisation.

Fatma M. Talaat [16] proposed the Load Balancing Algorithm (LBA) serves as software responsible for determining the fog server (FS) tasked with processing incoming requests. The Resource Allocation (RA) module utilizes a Reinforcement Learning (RL) algorithm to attain optimal load balancing in the fog environment. The selection priority for a specific FS is contingent upon the adaptive weight (AW) value.

Hoa Tran-Dang [18] proposed a perspective suggests significant potential for the application of Reinforcement Learning (RL) in the realm of fog computing, specifically concerning resource allocation for task offloading and execution to enhance overall performance. This study provides a comprehensive overview of RL applications designed to address resource allocation-related challenges within the fog computing environment[3].

3. Research Gap and Motivation

According to these research articles and authors finding the most of the techniques are focused on the load balancing strategy in the system is allocating the task to the nodes by the way balancing the load. (e.g., [18,23,19,20 ,22,21]) on the other hand scheduling the task among the node based on Decision – making process by the central controller. So analyzing the existing task in the node and also necessary information were analyze to schedule the task in that major necessary things to know is size of the task (e.g[24]) The number of running task/workload of the node (eg.,[25,19]). The computation capacity of each node eg.,[26]), etc., In this proposed work were the additional parameters are evaluated. The central controller keep on monitoring the fog nodes about the workload, no of task completed per cycle and also considering the time sensitive task has given the first priority.

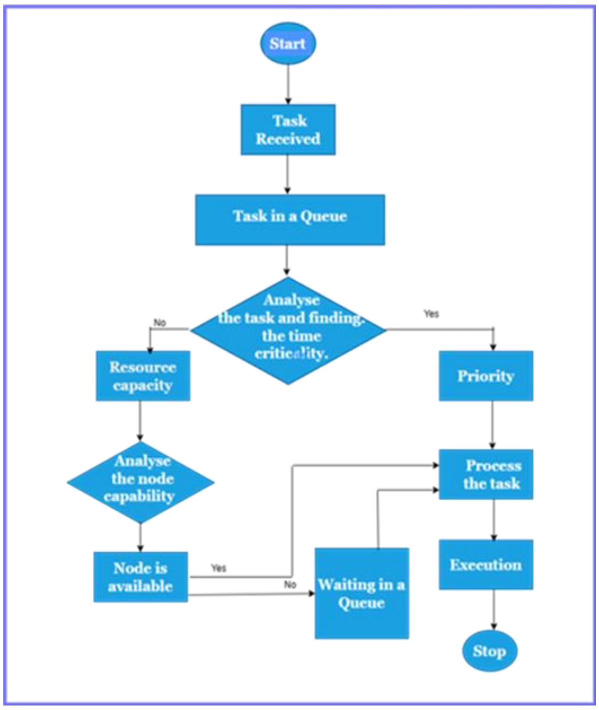

Fig 2: Flow diagram of the overall task scheduling system.

4. Methodology

Here considered Fog node as FN={F1,F2,F3…..Fn}, Task as {T1,T2,T3…Tn} and IoT devices for to generating the task to the fog node IF={I1,I2,I3…In}. To schedule the task based on the tow criteria one is capacity of the node and another one is priority if the task. Based on these two strategies the task scheduling is process effectivelyof the homophones "affect" (usually a verb) and "effect" (usually a noun), "complement" and "compliment," "discreet" and "discrete," "principal" (e.g., "principal investigator") and "principle" (e.g., "principle of measurement"). Do not confuse "imply" and "infer."

Prefixes such as "non," "sub," "micro," "multi," and "ultra" are not independent words; they should be joined to the words they modify, usually without a hyphen. There is no period after the "et" in the Latin abbreviation "*et al.*" (it is also italicized). The abbreviation "i.e.," means "that is," and the abbreviation "e.g.," means "for example" (these abbreviations are not italicized).

4.1. Performance evaluation of the Fog- Node capacity:

In this paper is modeled in order to obtained the load balancing among the fog nodes is carried out with strategy of scheduling the task by considering the capacity of the fog node. Based on the evaluation metrics the model finds the capacity of the each node. To identify the processing power in terms of million instructions per second (MIPS). It is the estimation by considering computational complexity and the instructions executes. This is calculated by

MIPS requirement(α) = Number of instructions / Execution Time (in seconds) * 100 ……...(1)

b) To identify the task requirement which is memory requirements RAM, Communication requirements (bandwidth required for communication between the task /node). This is calculated by

Memory Requirement(β)=Data Size(Ds)+Code Size(Cs)+Temporary Storage(Ts)…..……………................. (2)

c) To find the node capacity to know how much of the node is engaged with the workload by using

Node capacity (Ω) = Total node resources – Scheduled task resource …………………………………………………….(3)

α+β+Ω >=No .of task in queue …………………………….(4)

If the task is waiting in the queue is minimal when compare with these criteria then the model is more efficient than the traditional methods.

4.2. Performance evaluation of the Priority of the Task

Recognizing time-sensitive tasks within a fog computing setting is essential for optimizing task scheduling. These tasks are characterized by strict deadlines and the need for timely execution to fulfill particular performance criteria. The following approaches can be employed to pinpoint and manage time-critical tasks in the context of fog computing.

For that purpose, need to find these things

d) Task Profiling

To examine each task to grasp its attributes, encompassing factors like execution duration, resource needs, and interdependencies. Tasks that operate under stringent time limitations or ecognizing time-sensitive tasks within a fog computing setting is essential for optimizing task scheduling. These tasks are characterized by strict deadlines and the need for timely execution to fulfill particular performance criteria. The following approaches can be employed to pinpoint and manage time-critical tasks in the context of fog computing. For that purpose, need to find these things

have specific deadlines are likely to be considered as time-critical candidates.

ET = 1/Processing rate…………………………….……(5)

Here: ET is the execution time

Processing rate is the rate at which the task is processed. It is often measured in operation per second (OPS) or instruction per second (IPS)

e) Critical Path Analysis (CPA)

It is mainly focus on finding the time critical task for to give more priority that the other jobs/task in the queue which is calculated by using

Earliest Start Time (EST)

The earliest time a task can start/ finish . For the initial tasks/ Final, EST is usually 0.

ESTi = max(EST of all preceding tasks)……………...(6)

Earliest Finish time(EFT)

EFTi = ESTi + Duration of Task……………………..(7)

Latest Finish Time(LFT)

It should finish without delay

LFTi = min(LFT of all succeeding tasks) - Duration of Task …………………………………………………………….(8)

Latest Start Time (LST)

start the task without delay

LSTi= LFTi – Duration of task…………………….…..(9)

- i represents the task in question.
- "Preceding tasks" are tasks that must be completed before the current task can start.
- "Succeeding tasks" are tasks that depend on the completion of the current task.

5. Result and Discussion

In the case of uneven distributed of the workload in a fog computing environment is leads to the delay in task execution, energy consumption, Network congestion etc., The proposed work is to reduce the latency and improve the execution of the task ratio and most importantly execute the time critical task within the deadline period.

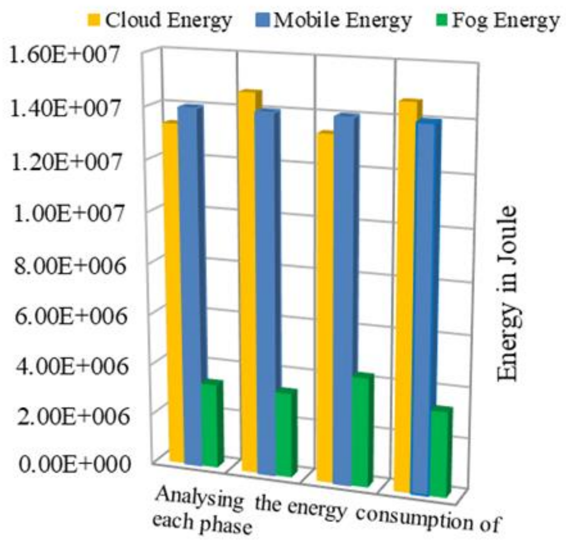

Fig 3 illustrates the consumption of the energy in all the three zones (Mobile (IoT device, Fog and cloud) because the data being received and retrieved from the IoT /Mobile device is closer to the Fog node. It distributes the load based on the load in the fog node.

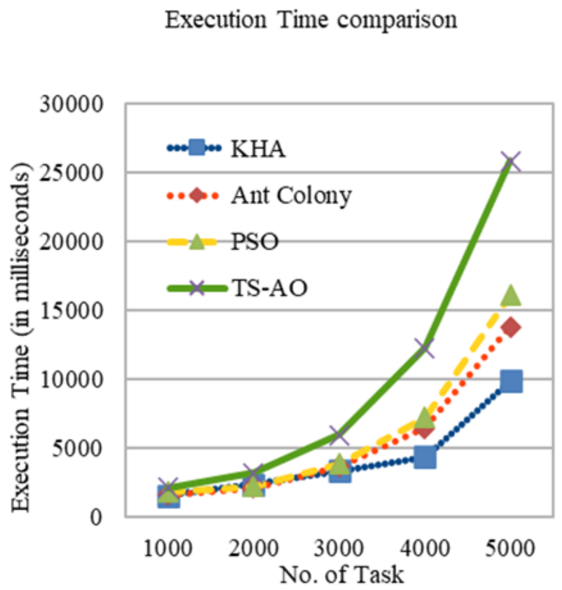

Fig 4: Execution Time comparison of KHA with traditional methods.

6. Conclusion

This paper considered the problem of executing the task in the fog computing environment. It is proposed a load balancing technique that distributes over the fog computing node to execute the given task with in the time period and also lesser energy consumption. The proposed method is improved the energy efficiency and execute the task within the time period effectively. However the proposed techniques exhibited higher efficiency while scheduling the task in order to load balancing the fog computing environment. In future, we aiming of intelligent based learning system to scheduling the task in an fog computing environment. R

Acknowledgements

This research was supported by my supervisor. We thank our colleagues from [SRM Institute of Science and Technology, KTR, Chennai] who provided insight and expertise that greatly assisted the research, although they may not agree with all of the interpretations/conclusions of this paper. I thank [Dr. K. Pradeep Mohan Kumar, Associate Professor] for assistance with [Load balancing Technology and methodology] and his comments that greatly improved the manuscript.

Author contributions

Archana R. 1: Conceptualization, Methodology, Software, Field study, Data curation, Writing-Original draft preparation, **Pradeep Mohan Kumar K.:** Software, Validation., Field study Visualization, Investigation, Writing-Reviewing and Editing.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] V. Chamola, C. -K. Tham and G. S. S. Chalapathi, "Latency aware mobile task assignment and load balancing for edge cloudlets," 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kona, HI, USA, 2017, pp. 587-592, doi: 10.1109/PERCOMW.2017.7917628.

[2] R. Archana and P. M. Kumar, "Utilization of Fog Computing in Task Scheduling and Offloading: Modern Growth and Future Challenges," 2022 International Conference on Electronic Systems and Intelligent Computing (ICESIC), Chennai, India, 2022, pp. 23-28, doi: 10.1109/ICESIC53714.2022.9783503.

[3] Talaat, F.M., Ali, H.A., Saraya, M.S. et al. Effective scheduling algorithm for load balancing in fog environment using CNN and MPSO. Knowl Inf Syst 64, 773–797 (2022). https://doi.org/10.1007/s10115-021-01649-2

[4] Atlam, H.F.; Walters, R.J.; Wills, G.B. Fog Computing and the Internet of Things: A Review. Big Data Cogn. Comput. 2018, 2, 10. https://doi.org/10.3390/bdcc2020010

[5] Tahmasebi-Pouya, N., Sarram, M.A. & Mostafavi, S. A reinforcement learning-based load balancing algorithm for fog computing. Telecommun Syst 84, 321–339 (2023). https://doi.org/10.1007/s11235-023-01049-7

[6] A. Pradhan, S. K. Bisoy, S. Kautish, M. B. Jasser and A. W. Mohamed, "Intelligent Decision-Making of Load Balancing Using Deep Reinforcement Learning and Parallel PSO in Cloud Environment," in IEEE Access, vol. 10, pp. 76939-76952, 2022, doi: 10.1109/ACCESS.2022.3192628.

[7] Onyema, E.M., Gude, V., Bhatt, A. et al. Smart Job Scheduling Model for Cloud Computing Network Application. SN COMPUT. SCI. 5, 39 (2024). https://doi.org/10.1007/s42979-023-02441-5

[8] S. Lipsa, R. K. Dash, N. Ivković and K. Cengiz, "Task Scheduling in Cloud Computing: A Priority-Based Heuristic Approach," in IEEE Access, vol. 11, pp. 27111-27126, 2023, doi: 10.1109/ACCESS.2023.3255781.

[9] Kaur, N., Kumar, A. & Kumar, R. TRAP: task-resource adaptive pairing for efficient scheduling in fog computing. Cluster Comput 25, 4257–4273 (2022). https://doi.org/10.1007/s10586-022-03641-z

[10] Fahimullah, M., Ahvar, S., Agarwal, M. et al. Machine learning-based solutions for resource management in fog computing. Multimed Tools Appl (2023). https://doi.org/10.1007/s11042-023-16399-2

[11] M. K. Hussein and M. H. Mousa, "Efficient Task Offloading for IoT-Based Applications in Fog Computing Using Ant Colony Optimization," in IEEE Access, vol. 8, pp. 37191-37201, 2020, doi: 10.1109/ACCESS.2020.2975741.

[12] Kishor, A., Chakarbarty, C. Task Offloading in Fog Computing for Using Smart Ant Colony Optimization. Wireless Pers Commun 127, 1683–1704 (2022). https://doi.org/10.1007/s11277-021-08714-7

[13] M. Abdel-Basset, N. Moustafa, R. Mohamed, O. M. Elkomy and M. Abouhawwash, "Multi-Objective Task Scheduling Approach for Fog Computing," in IEEE Access, vol. 9, pp. 126988-127009, 2021, doi: 10.1109/ACCESS.2021.3111130.

[14] Vishalika, Deepti Malhotra, “LD_ASG: Load Balancing Algorithm in Cloud Computing”, 2018, Fifth International Conference on Parallel, Distributed and Grid Computing (PDGC)

[15] 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.

[16] Talaat, F.M., Saraya, M.S., Saleh, A.I. et al. A load balancing and optimization strategy (LBOS) using reinforcement learning in fog computing environment. J Ambient Intell Human Comput 11, 4951–4966 (2020). https://doi.org/10.1007/s12652-020-01768-8

[17] H. Tran-Dang, S. Bhardwaj, T. Rahim, A. Musaddiq and D. -S. Kim, "Reinforcement learning based resource management for fog computing environment: Literature review, challenges, and open issues," in Journal of Communications and Networks, vol. 24, no. 1, pp. 83-98, Feb. 2022, doi: 10.23919/JCN.2021.000041.

[18] Archana, R., Pradeep Mohan, K.K., 2023. Fog Offloading and Scheduling in Traffic Monitoring System by Deep Reinforcement. IoT, Cloud and Data Science. https://doi.org/10.4028/p-c63rtd

[19] Singh, J., Singh, P., Amhoud, E. M., & Hedabou, M. (2022). Energy-efficient and secure load balancing technique for SDN-enabled fog computing. Sustainability, 14(19), 12951.

[20] Phan, L. A., Nguyen, D. T., Lee, M., Park, D. H., & Kim, T. (2021). Dynamic fog-to-fog offloading in SDN-based fog computing systems. Future Generation Computer Systems, 117, 486–497.

[21] Singh, P., Kaur, R., Rashid, J., Juneja, S., Dhiman, G., Kim, J., & Ouaissa, M. (2022). A fog-cluster based load-balancing technique. Sustainability, 14(13), 7961.

[22] Sharma, S., & Saini, H. (2019). A novel four-tier architecture for delay aware scheduling and load balancing in fog environment. Sustainable Computing: Informatics and Systems, 24, 100355.

[23] Xu, X., Fu, S., Cai, Q., Tian, W., Liu, W., Dou, W., Sun, X., & Liu, A. X. (2018). Dynamic resource allocation for load balancing in fog environment. Wireless Communications and Mobile Computing, 2018, 6421607.

[24] Pradhan, A., Bisoy, S. K., Kautish, S., Jasser, M. B., & Mohamed, A. W. (2022). Intelligent decision-making of load balancing using deep reinforcement learning and parallel PSO in cloud environment. IEEE Access, 10, 76939–76952.

[25] Zhu, X. k., Zhang, Q., Liu, L., Cheng, T., Yao, S., Zhou, W., & He, J. (2019). DLB: deep learning based load balancing. arXiv:1910.08494.

[26] Alvi, A. N., Javed, M. A., Hasanat, M. H. A., Khan, M. B., Saudagar, A. K. J., Alkhathami, M., & Farooq, U. (2022). Intelligent task offloading in fog computing based vehicular networks. Applied Sciences, 12(9), 4521.

[27] R. Archana, Pradeep Mohan Kumar K.,Multilevel scheduling mechanism for a stochastic fog computing environment using the HIRO model and RNN,Sustainable Computing: Informatics and

Systems,Volume 39,2023,100887,ISSN 2210-5379,https://doi.org/10.1016/j.suscom.2023.100887.

[28] A. Kadiyala and A. Kumar, "Applications ofpython to evaluate environmental data scienceproblems," Environmental Progress & Sustain-able Energy, vol. 36, no. 6, pp. 1580–1586,2017. (PDF) Load-balancing of computing resources in vehicular fog computing. Available from: https://www.researchgate.net/publication/348330314.