SURVIVAL STUDY ON LOAD BALANCING METHODS IN EDGE COMPUTING WITH HEALTHCARE DATA

M. Madan Mohan

Research Scholar, Department of Computer Science and Engineering, Faculty of Engineering, Karpagam Academy of Higher Education,

Dr. Laxmi Raja

Assistant Professor, Department of Computer Science and Engineering, Faculty of Engineering, Karpagam Academy of Higher Education,

ABSTRACT

Edge computing was the distributed structure among enterprise utilizations by data sources such as IoT or edge servers. Edge computing is positioned among IoT devices for minimizing the latency. In order to reacts end-user necessities, IoT application selects the edge nodes. Several load balancing algorithms is developed for minimizing latency in cloud environment. Load balancing was the important feature which detects allocation as well as management strategies. Due to the diversity and heterogeneity of edge nodes, load balancing is not employed with edge computing. However, congestion was not minimized through existing load balancing methods. In order to address these issues, several load balancing methods are explained as shown below.

Keywords: Edge computing, distributed computing, load balancing, Internet of Things, diversity, heterogeneity

1. INTRODUCTION

Edge computing was the distributed structure with resources over the cloud as well as data centers. Edge computing is employed for minimizing latency. Healthcare was one of developing industries by large potential for enhancement from employment like Internet-of-Things (IoT). Edge computing was the distributed structure to move the resources over cloud. Edge computing was positioned between IoT devices and cloud to reduce latency. Load balancing was an essential feature that detects the resource allocation as well as management strategies.

2. LITERATURE REVIEW

A load balancing strategy was designed in [1] through allocation of task with help of intermediary nodes. It monitored worldwide data for achieving real-time attributes for classification. But, latency was not reduced by load balancing strategy. A mobile healthcare framework was introduced in [2] depending on edge-fog-cloud collaborative network. Edge with fog devices were used to monitor the health for analyzing the data within abnormal fitness status. But, efficiency of load balancing is not enhanced at required level by mobile healthcare framework.

An energy-aware scheduler was designed in [3] with conditional constraints for real-time streaming applications. R-CTG approach minimized latency analysis lacking to reduce efficiency of energy. But, latency was not reduced by energy-aware scheduler. M/M/c/K queuing network scheme was designed in [4] to the enhancement of IoHT. The designed model considered the medical data in edge layer to local clients through fog layer. However, the latency was not minimized by designed model.

COTBIS was developed in [5] for edge computing construction in gateway stage. IoT increased the strength as well as cleverness within video surveillance schemes. But, the latency was not reduced by COTBIS. Federated Learning (FL) was introduced in [6] for privacy-preserved collaborative model. FL has motivation for contributing (WTP) with the concealed data. However, makespan was not reduced by FL.

Edge-based hybrid network scheme was designed in [7] with hybrid routers as well as IoT gateway. It increased coverage of short-range and supported edge computing tasks. But, the load balancing efficiency was not reduced by designed architecture.

An automatic service and resource discovery mechanism was designed in [8] for efficient deployment of nano-services on IoT nodes. However, the makespan was not reduced by automatic service and resource discovery mechanism. EOESPA and RNOESPA were introduced in [9] to minimize delay. Though the delay was reduced, the load balancing was not carried out in efficient manner.

A secure framework was introduced in [10] to SDN-based edge computing. IoT were authenticated with Edge servers. It gathered the information over patients as well as transmits to Edge servers. But, the scheduling efficiency was not improved. Hybrid Priority Assigned Laxity (HPAL) was introduced in [11] to assign Virtual Machine (VM) as well as complete charge with lesser time consumption. After task allocation, load balancing was managed with minimum execution time. Though execution time was minimized, the computational cost was not reduced.

Load balancing as well as computation offloading (CO) was designed in [12] to the MEC. Security layer was employed for avoiding protection problems. Load balancing was designed to the mobile device users between the small base stations. However, computational complexity is not minimized using CO technique.

Chaotic algorithm is designed in [13] with firefly as well as optimization of load balancing plan depending on chaotic firefly to address resource scheduling issues. The designed algorithm was accelerated to avoid into the local optimal solution. Multi-agent load balancing depending on deep reinforcement learning termed DTOMALB was introduced in [14] for distributed task allocation for enhancing the user knowledge. But, resource utilization rate was not improved.

5G communication was designed in [15] depending on edge computing. The key objective was to reduce energy consumption and delay constraints. But, computational complexity level is not reduced. DIDS task scheduling technique was designed in [16] depending on Q-Learning within reinforcement learning. The designed method varied the scheduling strategies consistent with network variations in edge computing environment. However, the time for scheduling was not reduced by low load DIDS task scheduling method.

MPVEC model was designed in [17] to minimize the cost of system in delay constraint. Multiple offloading node selection was introduced for choosing PVs through MEC within computing tasks. But, the space complexity was not reduced by MPVEC model. MOACO algorithm was designed in [18] for performing resource allocation among end users for cost mapping table creation and optimal allocation in MEC. However, the computational cost was not minimized.

A workload allocation mechanism was designed in [19] to reduce the service delay. The workload optimized allocation was carried out with minimal delay among multiple edge nodes for resource optimization within single edge node. Though delay was reduced, the optimal resources were not allocated in efficient manner. Cloud edge collaborative computing was designed in [20] depending on deep reinforcement learning. But, the memory consumption was not reduced.

Game-theoretic privacy-aware task allocation (G-PATA) was introduced in [21] to adjust the task rewards for guaranteeing task allocation with end devices meet QoS needs. However, load balancing time is not minimized. Popularity based placement technique is developed in [22] to link the data items and edge servers for retrieving the data depending on their virtual coordinate in plane. The placement plan was employed to handle load balancing among edge servers. But, the computational complexity was not reduced.

3. EFFICIENT LOAD BALANCING IN EDGE COMPUTING WITH HEALTHCARE DATA

IoT linked huge volume of smart devices in many advanced application infrastructures. Edge computing was to move cloud computing utilizations, data services over the nodes towards network edge. It was situated among terminal devices as well as computing data centers to manage minimal latency as well as real-time tasks. Edge computing minimized the latency and task assignment. Edge computing load balancing was an essential research topic in the academia. Edge computing comprised two kinds of load balancing plans, namely static as well as dynamic. Static load balancing algorithm not considered earlier node state while distributing load. Static load balancing algorithm functioned as nodes contain minute difference within load. Dynamic load balancing considered preceding state of node as distributing load depending on the intermediary nodes. Different load balancing algorithms have been designed to reduce the latency in cloud environment.

3.1 A new load balancing strategy by task allocation in edge computing based on intermediary nodes

Load balancing approach was introduced through task allocation depending on intermediary nodes. It was employed for observing worldwide data with real-time attributes for classification evaluation. Light-load, normal-load, as well as heavy-load are the three edge nodes categorized by inherent attributes as well as real-time attributes. Task assignment scheme was employed to assign novel tasks towards lightest load node. The designed approach balanced the load between edge nodes as well as minimized task completion time. Load balancing approach was employed t through dynamic load balancing with allocation of task.

Edge computing was designed depending on the intermediary nodes. It added intermediary nodes within edge computing as well as cloud computing for organizing global data of edge nodes. The designed method by disturbed first state, naive Bayes was employed for categorizes node. The original data was arranged to avoid importance of larger-value indicators within complete investigation among indicators differ significantly. The mathematical construction was used for examining load balancing issues among the edge nodes. Load balancing task was attained method through task assignment depending on transmission rate among edge nodes, estimated velocity as well as present tasks estimation time.

3.2 Internet of Health Things (IoHT) for personalized health care using integrated edge-fog-cloud network

The fast development in sensor-based system and Internet technology allowed healthcare technology. IoHT is a data exchange and data processing for health status monitoring of individuals through combining the IoT devices with mobile technologies. IoHT was the challenging function to personalized fitness care leverages on fog, edge and cloud computing. The mobility data was combined and analyzed along with health data. The key aim was to combine the geo-location information. Mobility pattern as well as traffic states were examined as well as health centre was recommended depending on fitness state. In cloud based healthcare system, health data were gathered through BSN. The data was stored as well as processed within servers.

Mobile healthcare was introduced depending on edge-fog-cloud collaborative network. The designed framework employed to perform the health data investigation within abnormal health category. Location variation of users was vital problem as well as delay within health information during emergency. Mobility of users was measured as well as mobility pattern was carried out within cloud. The designed healthcare framework was developed within laboratory as well as fitness data. The volunteers were examined to forecast the health status. The designed mobility scheme attained higher precision, recall value as well as efficiency of time. IoHT framework was introduced depending on edge fog-cloud collaborative network during emergency condition. The patient mobility scheme was introduced for recommending user concerning health centre as abnormal health condition was identified.

3.3 Energy-Aware Scheduling of Streaming Applications on Edge-Devices in IoT-Based Healthcare

A new energy-aware scheduler was introduced through task consideration. R-CTG was introduced by non-linear programming-based scheduling as well as voltage scaling method. The R-CTG approach reduced latency through re-timing lacking to minimize efficiency of energy. R-CTG minimized re-timing latency as re-times tasks reduced wasted slacks. The computational complexity of real-time utilizations was reduced through proliferation and VFI-based MPSoC architecture was employed for effective energy management. The complex scheduling issue was addressed for priority with VFI-NoC-MPSoC.

R-CTG approach employed priority model with successor-tree-consistent deadline as it allowed DVFS for using slack as well as minimize the energy. Successor-tree consistent deadline was described by upper bound on terminate time of nodes within CTG. The designed approach addressed the resource constraints of MPSoC with lesser time. NLP-based offline method scheduled tasks as well as nodes within successor-tree-reliable deadline way. The nodes by successor-tree consistent deadline were planned using successor-tree-consistent deadline. CA-TMES-Search approximated initial time to every task as considering conflict. CA-TMES-Quick mapped tasks as well as determined routes to communications. CA-TMES-Search saved better energy because of task mapping coordination and reduced the make span considerably.

4. PERFORMANCE ANALYSIS OF LOAD BALANCING METHODS IN EDGE COMPUTING WITH HEALTHCARE DATA

Experimental evaluation of existing load balancing techniques is developed by Java language. Simulation of existing load balancing methods is conducted using Cardiovascular Disease dataset consumed over Kaggle. Dataset is taken by https://www.kaggle.com/sulianova/cardiovascular-disease-dataset.The dataset comprises 13 attributes, namely id, age, gender, height, weight, smoke, etc. The dataset includes the 70000 records of patient data. Result analysis are carried out with existing methods with parameters are,

- Load balancing efficiency
- Load balancing time and
- Memory consumption

4.1 Impact on Load Balancing Efficiency

It was referred by proportion of number of user requested tasks were correctly balanced load to entire number of user requested tasks. It is formulated as,

$$LBE = \frac{Number\ of\ user\ requested\ correctly\ balanced\ the\ load}{Total\ number\ of\ user\ requested\ tasks} * 100 (1)$$

From (1), LBE was determined. It is calculated by percentage (%). When the load balancing efficiency was higher, the technique is very effective.

Table 1 Tabulation of Load Balancing Efficiency

Number of user requested tasks	Load Balancing Efficiency (%)		
	Load Balancing Strategy	Mobile Healthcare Framework	Energy-Aware Scheduler
10	85	78	70
20	88	80	72
30	90	82	75
40	92	84	78
50	89	81	76
60	87	79	73
70	85	77	71
80	83	74	69
90	86	76	72
100	89	79	74

Table 1 explains the load balancing efficiency by number of user requested tasks which varies of 10 to 100. Load balancing efficiency has load balancing strategy, mobile healthcare framework and energy-aware scheduler. Load balancing strategy has intermediary nodes via task allocation in edge computing. New re-timing technique termed R-CTG integrated using non-linear programming-based scheduling and voltage scaling approach called as ALI-EBAD. Mobile healthcare framework was depending on edge-fog-cloud collaborative network. Consider the number of user requested tasks was 40, load balancing efficiency is 92%. The load balancing efficiency of mobile healthcare framework and energy-aware scheduler are 84% and 78%. The Visual illustration representation of load balancing efficiency is explained from figure 1.

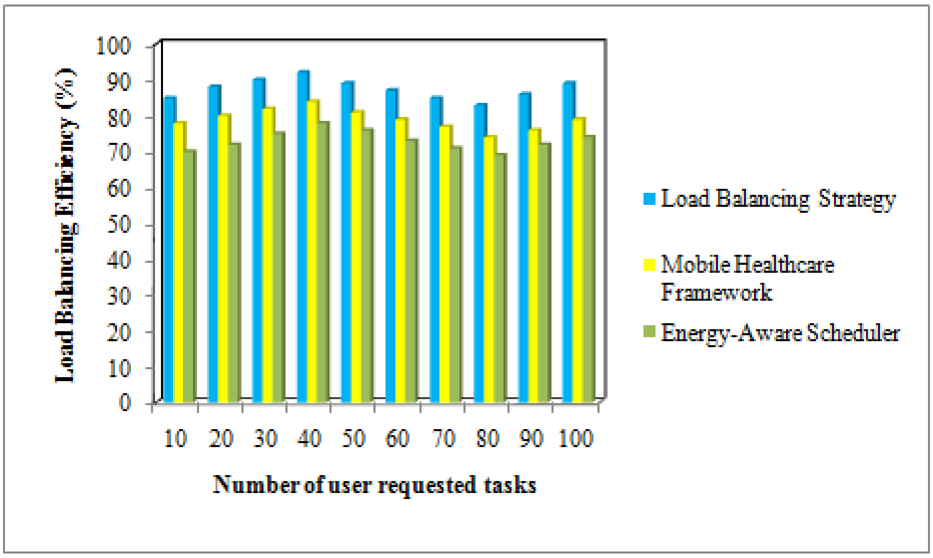

Figure 1 Measurement of Load Balancing Efficiency

From figure 1, the load balancing efficiency for different number of user requested task is explained. The blue color bar indicates load balancing efficiency of load balancing strategy. The red color bar and green color bar indicates load balancing efficiency of mobile healthcare framework and energy-aware scheduler correspondingly. It is observed that the load balancing efficiency using load balancing strategy is higher when compared to the mobile healthcare framework and energy-aware scheduler. This is due to the application of task assignment model to assign the novel tasks towards lightest load node. The designed approach balanced load between edge nodes as well as reduced task completion time. Thus, load balancing efficiency was enhanced using 11% when compared with mobile healthcare framework and 20% compared with energy-aware scheduler.

4.2 Impact on Load Balancing Time

Load balancing time (LBT) was described by product of number of user requested tasks and time consumed to perform load balancing of one task. It is formulated as,

$$LBT = N * Time\ consumed\ to\ balance\ one\ user\ requested\ task \quad (2)$$

From (2), the load balancing time is calculated. It was calculated by milliseconds (ms). When LBT was lesser, the technique is very effective.

Table 2 Tabulation of Load Balancing Time

Number of user requested tasks	Load Balancing Time (ms)		
	Load Balancing Strategy	Mobile Healthcare Framework	Energy-Aware Scheduler
10	31	25	42
20	33	27	45
30	36	29	48
40	38	32	50
50	40	34	52
60	43	37	55
70	45	39	57
80	47	42	60
90	49	44	62
100	51	47	64

Table 2 describes the load balancing time by number of user requested tasks ranging from 10 to 100. Load balancing time comparison takes place on the existing load balancing strategy, mobile healthcare framework and energy-aware scheduler. Consider the user requested tasks is 90, the load balancing time of load balancing strategy is 49ms. The load balancing time of mobile healthcare framework and energy-aware scheduler are 44ms and 64ms. The graphical representation of load balancing time is illustrated in figure 2.

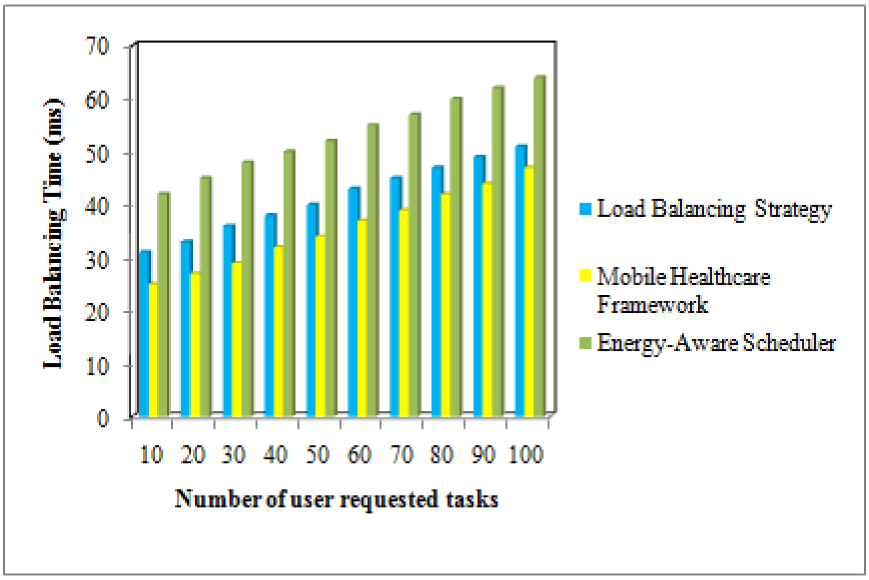

Figure 2 Measurement of Load Balancing Time

From the figure 2, the load balancing time for different number of user requested task is described. Blue color bar indicates load balancing time of load balancing strategy. Red color bar as well as green color bar represents load balancing time of mobile healthcare framework as well as energy-aware scheduler correspondingly. Load balancing time with mobile healthcare was minimal compared with load balancing strategy and energy-aware scheduler. This is due to the application of mobility prediction model and IoHT framework for improving the time-efficiency. IoHT fraework depends on edge fog-cloud collaborative network during emergency condition. This in turn helps to reduce the load balancing time. As a result, load balancing time of mobile healthcare was minimized using 14% compared with load balancing strategy and 34% when compared to the energy-aware scheduler.

4.3 Impact on Memory Consumption

It is referred by product of number of user requested tasks and memory consumed to balance one user requested task. It was determined by MegaBytes (MB). When the memory consumption is lesser, the technique is said to be more efficient.

Table 3 Tabulation of Memory Consumption

Number of user requested tasks	Memory Consumption (MB)		
	Load Balancing Strategy	Mobile Healthcare Framework	Energy-Aware Scheduler
10	29	24	18
20	31	26	21
30	33	28	22
40	35	31	25
50	38	33	27
60	40	35	29
70	42	37	31
80	44	41	33
90	46	43	35
100	48	44	38

Table 3 describes the tabulation of memory consumption by number of user requested tasks ranging from 100 to 1000. Memory consumption comparison takes place on the existing load balancing strategy, mobile healthcare framework and energy-aware scheduler. Consider the user requested tasks was 60, the memory consumption of load balancing strategy is 40MB. The memory consumption of mobile healthcare framework and energy-aware scheduler are 35MB and 29MB. The graphical representation of memory consumption is described in figure 3.

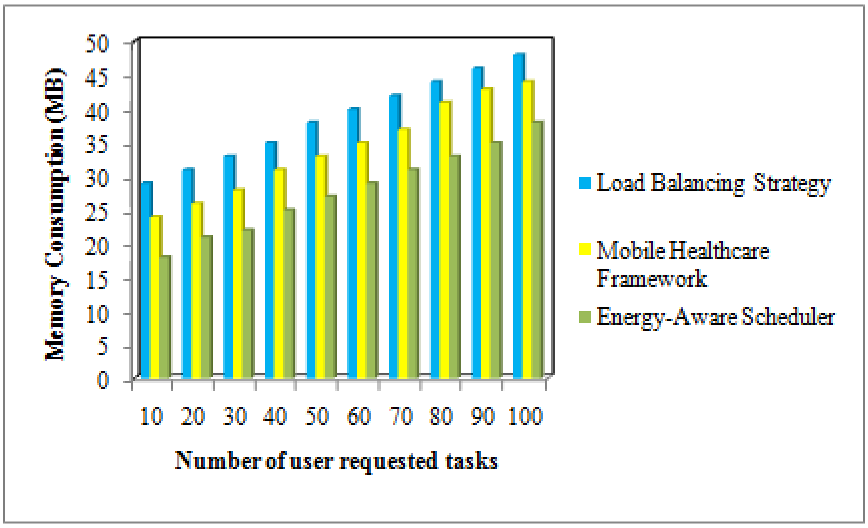

Figure 3 Measurement of Memory Consumption

From the figure 3, the memory consumption for different number of user requested task was explained. Blue color bar denotes memory consumption of load balancing strategy. Red color bar as well as green color bar represents memory consumption of mobile healthcare framework and energy-aware scheduler correspondingly. It is seen that the memory consumption using energy-aware scheduler is lesser when compared to the load balancing strategy and mobile healthcare framework. This is due to the application of NLP-based offline method to schedule tasks as well as communication within successor-tree-consistent deadline way. It aids to minimize memory consumption. As a result, memory consumption of energy-aware scheduler is reduced by 28% when compared to the load balancing strategy and 19% when compared to the mobile healthcare framework.

COMPARISON TABULATION

METHODS	Load Balancing Efficiency	Load Balancing Time	Memory Consumption
Load Balancing Strategy	87.3 %	41.3 ms	38.6 MB
Mobile Healthcare Framework	79 %	35.6 ms	34.2 MB
Energy-Aware Scheduler	73%	53.5 ms	27.9 MB

5. DISCUSSION AND LIMITATION ON EXISTING LOAD BALANCING METHODS IN EDGE COMPUTING

Load balancing strategy was introduced with intermediary nodes through allocation of tasks within edge computing. The intermediary node monitored global data for achieving edge nodes with better performance. The designed strategy balanced load between edge nodes as well as minimized task completion time. But, latency was not reduced by load balancing strategy.

The designed framework employed edge as well as fog strategy to the fitness observation in abnormal health status. The designed framework minimized delay as well as consumption of energy. But, load balancing efficiency was not increased at required level by mobile healthcare framework. The optimum resource allocation was not carried out for edge-fog-cloud based collaborative network.

A new energy-aware scheduler considered tasks by conditional limitations on VFI-based heterogeneous NoC-MPSoCs for real-time applications. R-CTG approach reduced latency foundation with re-timing energy-efficiency. QoE not considered the motivating metrics over user perspectives. But, latency was not reduced by energy-aware scheduler.

5.1 Future Direction

The future direction of employment is performed by machine learning methods to enhancing load balancing through improved efficiency as well as minimal time consumption.

6. CONCLUSION

A comparison of different load balancing methods is explained. From study, it is examined that the latency is not minimized using energy-aware scheduler. It explains the optimum resource allocation was not carried out for edge-fog-cloud based collaborative network. In addition, load balancing efficiency was not increased at required level by mobile healthcare framework. Simulation of several load balancing techniques achieves the performance and advantages. Lastly, research work is carried out by machine learning methods for increasing load balancing performance.

REFERENCES

[1] Guangshun Li, Yonghui Yao, Junhua Wu, Xiaoxiao Liu, Xiaofei Sheng andQingyan Lin, "A new load balancing strategy by task allocation in edge computing based on intermediary nodes", EURASIP Journal on Wireless Communications and Networking, Springer, Volume 2020, Issue 3, 2020, Pages 1-15

[2] Anwesha Mukherjee, Shreya Ghosh, AabhasBehere and Soumya K. Ghosh, RajkumarBuyya, "Internet of Health Things (IoHT) for personalized health care using integrated edge-fog-cloud network", Journal of Ambient Intelligence and Humanized Computing, Springer, Volume 12, 2021, Pages 943-959

[3]Umair Ullah Tariq, Haider Ali, Lu Liu, James Hardy, Muhammad Kazim and Waqar Ahmed, "Energy-Aware Scheduling of Streaming Applications on Edge-Devices in IoT-Based Healthcare", IEEE Transactions on Green Communications and Networking, Volume 5, Issue 2, June 2021, Pages 803-815

[4] Francisco Airton Silva, Tuan Anh Nguyen, Iure Fe, Carlos Brito, Dugki Min and Jae-Woo Lee, "Performance Evaluation of an Internet of Healthcare Things for Medical Monitoring Using M/M/c/K Queuing Models", IEEE Access, Volume 9, April 2021, Pages 55271 - 55283

[5]RajkumarRajavel, Sathish Kumar Ravichandran, Karthikeyan Harimoorthy, Partheeban Nagappan and KanagachidambaresanRamasubramanian, "IoT-based smart healthcare video surveillance system using edge computing", Journal of Ambient Intelligence and Humanized Computing, Springer, 2021, Pages 1-15

[6] Wei Yang Bryan Lim, Sahil Garg, ZehuiXiong, Dusit Niyato, Cyril Leung, Chunyan Miao and Mohsen Guizani, "Dynamic Contract Design for Federated Learning in Smart Healthcare Applications", IEEE Internet of Things Journal, Volume 8, Issue 23, December 2021, Pages 16853 – 16862

[7]Fan Wu,ChunkaiQiu, Taiyang Wu and Mehmet RasitYuce,"Edge-Based Hybrid System Implementation forLong-Range Safety and Healthcare IoT Applications", IEEE Internet of Things Journal, Volume 8, Issue 12, June 2021, Pages 9970-9980

[8] Johirul Islam, Tanesh Kumar, Ivana Kovacevic and ErkkiHarjula,"Resource-Aware Dynamic Service Deployment forLocal IoT Edge Computing: Healthcare Use Case", IEEE Access, Volume 9, August 2021, Pages 115868 - 115884

[9] ZhihanLv and LiangQiao,"Optimization of collaborative resource allocation for mobile edge computing", Computer Communications, Elsevier, Volume 161, 1 September 2020, Pages 19-27

[10] Junxia Li, Jinjin Cai, Fazlullah Khan, Ateeq Ur Rehman, VenkiBalasubramaniam, Jiangfeng Sun, and P. Venu, "A Secured Framework for SDN-Based Edge Computing in IoT-Enabled Healthcare System", IEEE Access, Volume 8, July 2020, Pages 135479 – 135490

[11] Simar Preet Singh, Rajesh Kumar, Anju Sharma, Jemal H. AbawajyandRavneetKaur,"Energy efficient load balancing hybrid priority assigned laxity algorithm in fog computing",Cluster Computing, Volume 2022, March 2022, Pages 1-15

[12] Wei-Zhe Zhang, Ibrahim A. Elgendy, Mohamed Hammad, Abdullah M. Iliyasu,Xiaojiang Du, Mohsen Guizani and Ahmed A. Abd El-Latif,"Secure and Optimized Load Balancing for Multitier IoT and Edge-Cloud Computing Systems",IEEE Internet of Things Journal, Volume 8, Issue 10, May 2021, Pages 8119 – 8132

[13] Huafu Yong, "Load balancing strategy for medical big data based on low delay cloud network", The Journal of Engineering, IET Jounals, Volume2020, Issue9, September 2020, Pages 799-804

[14] Zhenjiang Zhang, Chen Li, ShengLung Peng and Xintong Pei, "A new task offloading algorithm in edge computing", Eurasip Journal on Wireless Communications and Networking, Volume 2017, Issue 17, 2017, Pages 1-21

[15] Lin Cao., “5G communication resource allocation strategy based on edge computing”, The journal of Engineering, Volume 2022, Issue 3, March 2022, Pages 311-319

[16] XuZhao, Guangqiu Huang, Ling Gao, MaozhenLi and QuanliGao, “Low load DIDS task scheduling based on Q-learning in edge computing environment”, Journal of Network and Computer Applications, Elsevier, Volume 188, August 2021, Pages 1-15

[17]Xinyue Hu,Xiaoke Tang, Yantao Yu,SihaiQiu and ShiyongChen,“Joint Load Balancing and Offloading Optimization in Multiple Parked Vehicle-Assisted Edge Computing”, Wireless Communications and Mobile Computing, Hindawi Publishing Corporation, Volume 2021, 2021, Pages 1-16

[18] S.Vimal, ManjuKhari, NilanjanDey, Ruben GonzalezCrespo and Y.HaroldRobinson,“Enhanced resource allocation in mobile edge computing using reinforcement learning based MOACO algorithm for IIOT”,Computer Communications, Elsevier, Volume 151, February 2020, Pages 355-364

[19] XudongNiu,Sujie Shao, Chen Xin, Jun Zhou,Shaoyong Guo, Xingyu Chen and Feng Qi, “Workload Allocation Mechanism for Minimum Service Delay in Edge Computing-Based Power Internet of Things”, IEEE Access, Volume 7, May 2019, Pages 83771 – 83784

[20] ZengrongZhang,“A computing allocation strategy for Internet of things’ resources based on edge computing”, International Journal of DistributedSensor Networks, Volume 17, Issue 12, 2021, Pages 1-11

[21] Daniel Zhang, Yue Ma, Xiaobo Sharon Hu and Dong Wang “Toward Privacy-Aware Task Allocation in Social Sensing-Based Edge Computing Systems”, IEEE Internet of Things Journal, Volume 7, Issue 12, December 2020, Pages 11384 – 11400

[22] Xinliang Wei and Yu Wang, “Popularity-based Data Placement with Load Balancing in Edge Computing”,IEEE Transactions on Cloud Computing, July 2021, Pages 1 – 12