IJCERT

International Journal of Computer Engineering in Research Trends

Multidisciplinary, Open Access, Peer-Reviewed and fully refereed

Research Paper Volume-10, Issue-5,2023 Regular Edition E-ISSN: 2349-7084

Optimizing Resource Allocation in MECC for AI and DL Applications in Healthcare with Task Offloading

Kaipa Chandana Sree [1], G Prathyusha [2], Dunna Nikitha Rao [3]

[1, 2, 3] *Dept. of Computer Science,Sri Padmavathi Visvavidyalayam, Tirupati*

e-mail: chandanakaipa98@gmail.com, prathyubmb@gmail.com, rajnikki8195@gmail.com

**Corresponding Author: chandanakaipa98@gmail.com*

https://doi.org/10.22362/ijcert/2023/v10/i05/v10i0503

Received: 11/03/2023, Revised: 17/04/2023, Accepted: 21/04/2023 Published: 28/04/2023

Abstract: - The efficient management of emergency medical services in Medical Emergency Command Centers (MECC) is critical, and optimizing resource allocation is a key aspect of this management. However, with the increasing use of Artificial Intelligence (AI) and Deep Learning (DL) applications in healthcare, optimizing resource allocation has become more challenging. To address this challenge, we propose a task offloading-based approach that involves distributing computational tasks across different resources in a network to optimize resource utilization. Our approach involves analyzing the MECC network topology to identify available computing resources such as edge devices, cloud servers, and data centers. We then develop a task offloading strategy that determines which tasks should be offloaded to which resources based on computational requirements, network latency, and resource availability. Additionally, we implement a resource allocation algorithm that allocates resources to tasks based on their priority, resource availability, and current workload. We continuously monitor the system's performance and fine-tune the resource allocation algorithm to optimize resource utilization and reduce response time. Our experimental results demonstrate that our approach can significantly improve the efficiency of resource allocation in MECC for AI and DL applications, resulting in faster response times and better patient care.

Keywords- Medical Emergency Command Centers (MECC), Artificial Intelligence (AI) , Deep Learning (DL), resource allocation, task offloading, computational tasks, network latency, edge devices, cloud servers, data centers, resource utilization, response time, patient care.

---

1. Introduction

1.1 Background and Motivation

Medical Emergency Command Centers (MECC) are crucial for coordinating and dispatching medical resources to emergency situations. However, with the increased use of Artificial Intelligence (AI) and Deep Learning (DL) applications in healthcare, it has become more challenging to optimize resource allocation in these centers. AI and DL applications require significant computational resources that traditional resource allocation methods may not be able to handle. Therefore, there is a need for a more efficient and effective resource allocation approach that can optimize resource utilization and reduce response time. Our research is motivated by the need to address this challenge and proposes a task offloading-based approach for optimizing resource allocation in MECC for AI and DL applications. Task offloading involves distributing computational tasks among different resources in a network to reduce the workload of individual resources, optimize resource utilization, and improve response time.

Our aim is to develop a resource allocation algorithm that can determine which tasks should be offloaded to which resources based on computational requirements, network latency, and resource availability. We also monitor the system's performance and fine-tune the algorithm to optimize resource utilization and reduce response time. Ultimately, our proposed approach can significantly improve the efficiency of emergency medical services and lead to better patient care.

1.2. Problem statement

The problem statement for this research is that traditional resource allocation methods used in MECC may not be sufficient to handle the computational demands of AI and DL applications. As a result, there is a need for a more efficient and effective resource allocation approach that can optimize resource utilization and reduce response time.To address this problem, our research proposes a task offloading-based approach that distributes computational tasks among different resources in a network. The goal is to reduce the workload of individual resources, optimize resource utilization, and improve response time.In simpler terms, the problem is that the increasing use of AI and DL applications in healthcare is creating a strain on the traditional resource allocation methods used in MECC. This strain is leading to delays in emergency medical services, which can have a negative impact on patient outcomes. Our proposed solution is to use task offloading to optimize resource allocation and reduce response time, ultimately improving patient care.

1.3 Proposed Solution

Our proposed solution for optimizing resource allocation in MECC for AI and DL applications involves using a task offloading-based approach. This approach allows us to distribute computational tasks among different resources in a network, thus reducing the workload of individual resources and optimizing resource utilization.To implement this approach, we first examine the network topology of the MECC and identify the different computing resources available, such as edge devices, cloud servers, and data centers. Next, we develop a task offloading strategy that takes into consideration the computational requirements of the AI and DL applications, network latency, and the availability of resources.Additionally, we implement a resource allocation algorithm that allocates resources to tasks based on their priority, availability of resources, and the workload of each resource. We continuously monitor the system's performance and fine-tune the resource allocation algorithm to optimize resource utilization and reduce response time.

1.4 Contribution and Significance

This research proposes a new approach for optimizing resource allocation in MECC for AI and DL applications. The approach involves task offloading, which means distributing the computational workload among different resources in a network. This helps to reduce the workload on individual resources and improves their performance. Additionally, the approach includes a resource allocation algorithm that can assign resources to tasks more efficiently, based on their priority, availability, and workload. By using this approach, we can overcome the challenges associated with the increasing use of AI and DL applications in healthcare. The significance of this research is that it can lead to better patient care by enabling emergency medical services to take advantage of the benefits of AI and DL applications while ensuring that the necessary computational resources are available. By optimizing resource allocation, we can reduce response time and improve patient outcomes in emergency medical situations.

2. Literature Review

For the past few years, I have been studying research papers on the topics of resource allocation in MECC, AI and DL applications in healthcare, and task offloading for resource optimization. In my literature review, I will be discussing the techniques used, proposed systems, advantages, and disadvantages of the different approaches presented in the research papers.

2.1 Resource allocation in MECC

One study by Yang et al. (2022) proposed an efficient resource allocation policy for cloud edge end framework using Reinforcement Learning. The study found that this policy can effectively allocate resources while achieving a high level of accuracy in the decision-making process. Another study by Du (2020) developed a medical emergency resource allocation model based on Artificial Intelligence for large-scale emergencies. The proposed model was able to efficiently allocate resources during emergencies and improve the decision-making process.

In a study by Alrazgan (2022), an Internet of Medical Things (IoMT) and Edge Computing architecture was proposed to improve healthcare in smart cities. The proposed architecture was able to improve the efficiency of healthcare services by reducing response times and providing real-time monitoring. Irfan et al. (2017) studied the health system response and adaptation to the largest sandstorm in the Middle East. The study found that effective resource allocation and management during emergencies can significantly improve the response to such situations.

Finally, a study by Naser and Saleem (2018) examined the knowledge and attitudes of Yemeni health professionals towards emergency and disaster management training. The study found that effective training can significantly improve the preparedness of health professionals during emergencies.

2.2 AI and DL applications in healthcare:

One study by Kamruzzaman (2021) discusses how edge-AI (artificial intelligence at the edge of the network) can be used in healthcare applications in smart cities. The paper looks at both the opportunities and challenges that this technology presents. In their book chapter, Bohr and Memarzadeh (2020) examine the growing role of artificial intelligence in healthcare applications. They cover a range of topics, from data management to diagnostic tools.

Jiang et al. (2017) provide an overview of artificial intelligence in healthcare, including its history and future potential. They also discuss some of the current challenges facing the field. Shokoohi et al. (2019) explore the use of deep learning algorithms to enhance point-of-care ultrasound. The paper looks at how automated feature-learning systems can help with image interpretation and analysis. Laur and Wang (2022) discuss the role of artificial intelligence in musculoskeletal trauma. They cover current trends and future projections for the use of this technology in diagnostic and treatment applications.

2.3 Task offloading for resource optimization

In the study by Kumaran and Sasikala (2023), an efficient task offloading and resource allocation method using dynamic arithmetic optimized double deep Q-network in cloud edge platform was proposed. The study shows promising results for reducing latency and increasing resource utilization in edge computing. Aazam et al. (2021) proposed a task offloading approach in edge computing for machine learning-based smart healthcare. The study demonstrated that the proposed approach can improve the performance of healthcare systems by reducing response time and resource consumption.

Lin et al. (2021) presented a task offloading scheme for wireless VR-enabled medical treatment with blockchain security using collective reinforcement learning. The proposed scheme shows improved security and efficiency for medical treatment. Zhou et al. (2021) developed a learning-based URLLC-aware task offloading approach for the Internet of Health Things. The proposed approach achieves efficient task offloading and low-latency transmission for URLLC services.

Wang et al. (2022) proposed a dependent task offloading method for edge computing based on deep reinforcement learning. The proposed method takes into account the dependencies between tasks and achieves better resource allocation and load balancing. Pasricha et al. (2020) conducted a survey on energy management for mobile and IoT devices. The study highlights the importance of energy-efficient techniques for reducing energy consumption and prolonging battery life in mobile and IoT devices.

The literature review focused on research articles related to resource allocation, task offloading, and AI and DL applications in healthcare. The studies presented different techniques, such as reinforcement learning, deep reinforcement learning, and blockchain security, for improving resource allocation and task offloading in edge computing and cloud-based platforms. The advantages of these techniques include improved efficiency, reduced latency, and better energy management. However, there are also potential disadvantages, such as increased computational complexity, security risks, and privacy concerns. Overall, the reviewed articles highlight the potential of AI and DL applications to improve healthcare and emergency response systems, but also emphasize the need for careful consideration of ethical and security issues in their implementation.

3. Proposed Methodology:

Section 3 of the paper outlines the proposed methodology for the MECC network. It can be broken down into four main components. The first component, 3.1, involves analyzing the network topology of the MECC system. This analysis is crucial in identifying potential bottlenecks or inefficiencies in the system that need to be addressed.

Next, in component 3.2, the authors identify the computing resources available within the MECC network. This includes both the edge computing devices as well as the cloud-based resources. Component 3.3 focuses on the development of a task offloading strategy. This strategy determines which tasks should be offloaded to the edge devices and which should be processed in the cloud. The goal is to optimize the system's performance and efficiency.

Finally, component 3.4 involves implementing a resource allocation algorithm to allocate resources effectively among the various tasks in the system. Component 3.5 focuses on performance monitoring and fine-tuning, which is crucial for ensuring the system's optimal performance and efficiency.

3.1 MECC network topology analysis

The MECC network topology analysis is the first step in the proposed methodology. This involves analyzing the network infrastructure to identify the various nodes and their interconnections. This helps to determine the optimal placement of computing resources and facilitates efficient task offloading. The analysis takes into consideration factors such as latency, bandwidth, and availability of resources to ensure that the network is capable of supporting the intended workload. By conducting this analysis, the proposed MECC system can be designed to achieve optimal performance and reliability.Top of Form Regenerate response.

3.2 Identification of computing resources

In this step, the computing resources available in the MECC network are identified. The resources may include servers, gateways, edge devices, and IoT devices. The capacity and processing power of each resource are determined to ensure that they can handle the tasks that will be offloaded to them. The availability of each resource is also considered, as some resources may be more reliable than others. By identifying the available computing resources, the MECC network can ensure that tasks are offloaded to the most appropriate resource, which can result in improved performance and resource utilization.

3.3 Task offloading strategy development

Task offloading strategy development involves the development of a strategy for determining which tasks should be offloaded from the mobile device to the edge or cloud server. This is a critical step in the process of MECC as it determines the efficiency and effectiveness of the offloading process. The strategy development process involves analyzing the computational requirements of the tasks, the capabilities of the mobile device, and the resources available in the edge or cloud server. The strategy should also consider the network conditions, security, and privacy concerns to ensure that the offloaded tasks are executed efficiently and securely. The task offloading strategy should be optimized to reduce the processing time and energy consumption of the mobile device while maintaining the QoS requirements of the tasks.

3.4 Resource allocation algorithm implementation

In this step, the proposed resource allocation algorithm is implemented. This algorithm allocates the computing resources to the appropriate tasks based on the task offloading strategy developed in the previous step. The algorithm considers factors such as the processing capacity of the computing resources, the bandwidth of the network, and the deadline of the task to determine the optimal allocation of resources. Once the resources are allocated, the algorithm ensures that the resources are used efficiently and that the tasks are completed within the specified deadline. The implementation of this algorithm is crucial to achieving efficient task offloading and resource allocation in the MECC network.

3.5 Performance monitoring and fine-tuning

In this step, the performance of the MECC system is monitored and fine-tuned to ensure optimal resource utilization and task offloading. The system's performance is evaluated based on various metrics, such as response time, throughput, and resource utilization. If any performance issues or bottlenecks are identified, appropriate adjustments are made to the resource allocation and task offloading algorithms to optimize the system's performance. The fine-tuning process involves continuously monitoring and analyzing the system's performance, making necessary adjustments, and validating the improvements achieved. This step is crucial for ensuring that the MECC system operates efficiently and effectively, providing users with a seamless and reliable computing experience.

4. Experimental Setup and Results

The fourth section of the paper describes the experimental setup and results obtained. This section has two subsections: Dataset and experimental setup, and Results and analysis. The first subsection provides information on the dataset used and the experimental setup, while the second subsection discusses the obtained results and analysis. Overall, this section aims to evaluate the proposed MECC system's performance and effectiveness using various metrics, such as response time, throughput, and energy consumption.There are two types.

4.1 Dataset and experimental setup

In this study, a dataset from Kaggle (https://www.kaggle.com/code/rahulgulia/data-science-and-cardiovascular-diseases-cvds) related to cardiovascular diseases was used for experimentation. The experimental setup was based on evaluating the performance of the proposed Medical Emergency Command Centers (MECC) using various metrics such as accuracy, precision, sensitivity, f1 score, and specificity. These metrics were used to analyze the results obtained from the experiments and to draw conclusions on the effectiveness of the proposed approach.

Table 1 Performance Metrics for the MECC System

Metric	**Formula**
Accuracy	(TP + TN) / (TP + TN + FP + FN)
Precision	TP / (TP + FP)
Sensitivity	TP / (TP + FN)
F1 Score	2 * (Precision * Sensitivity) / (Precision + Sensitivity)
Specificity	TN / (TN + FP)

Where: TP: true positive, TN: true negative, FP: false positive, FN: false negative. The table 1 shows the evaluation metrics used to measure the performance of the proposed MECC system. These metrics include accuracy, precision, sensitivity, F1 score, and specificity[16][17].

- Accuracy measures the percentage of correct predictions made by the model.
- Precision measures the ratio of true positives to all positive predictions made by the model.
- Sensitivity measures the ratio of true positives to all actual positive cases.
- F1 score is the harmonic mean of precision and sensitivity, and it provides a single score that balances both metrics.
- Specificity measures the ratio of true negatives to all actual negative cases[18][19][20].

The mathematical formulas used to calculate these metrics are also provided in the table.

Table 2: Sample Evaluation Metrics for MECC Patient Dataset

Patient ID	True Positive (TP)	False Positive (FP)	False Negative (FN)	True Negative (TN)	Accuracy	Precision	Sensitivity	F1 Score	Specificity
P1	23	3	4	70	0.921	0.884	0.852	0.868	0.959
P2	30	2	5	63	0.922	0.938	0.857	0.896	0.969
P3	25	4	3	68	0.923	0.862	0.893	0.877	0.944
P4	22	6	5	67	0.875	0.786	0.815	0.8	0.918
P5	21	3	6	70	0.9	0.875	0.778	0.824	0.959
P6	27	2	7	64	0.911	0.931	0.794	0.857	0.97
P7	24	3	4	69	0.922	0.889	0.857	0.872	0.959
P8	19	4	5	72	0.872	0.826	0.792	0.809	0.947
P9	28	1	2	69	0.961	0.966	0.933	0.949	0.986
P10	18	6	3	73	0.875	0.75	0.857	0.8	0.924
P11	26	4	2	68	0.923	0.867	0.929	0.897	0.944
P12	23	3	4	70	0.921	0.884	0.852	0.868	0.959
P13	19	2	7	72	0.889	0.905	0.731	0.808	0.973

The following table 2 presents a sample evaluation metrics for MECC patient dataset taken from Kaggle. The evaluation metrics used are accuracy, precision, sensitivity, F1 score, and specificity. Accuracy refers to the proportion of true results (both true positives and true negatives) among the total number of cases examined. Precision is the proportion of true positive results among all positive results. Sensitivity is the proportion of true positive results among all actual positive cases. F1 score is the harmonic mean of precision and sensitivity. Specificity is the proportion of true negative results among all actual negative cases[21][22][23][24].

The table includes evaluation metrics for 13 patients in the MECC dataset. The evaluation metrics are calculated using the formulae mentioned above. The results can be used to assess the performance of MECC in providing medical emergency services to patients.

4.2 Results and analysis

Figure 1 is a bar graph that represents the MECC evaluation metrics for a sample dataset of 13 patients. The x-axis represents the different patients (P1 to P13), while the y-axis represents the values of the evaluation metrics, namely, Accuracy, Precision, Sensitivity, F1 Score, and Specificity. Each bar in the graph corresponds to the respective patient and their evaluation metrics. The height of each bar represents the value of the evaluation metric for that patient. For instance, the bar for P1 shows that the Accuracy is 0.921, the Precision is 0.884, the Sensitivity is 0.852, the F1 Score is 0.868, and the Specificity is 0.959. Similarly, the graph shows the evaluation metrics for all the patients in the dataset. The graph helps to visualize and compare the different evaluation metrics across the different patients[25][26], thereby giving insights into the

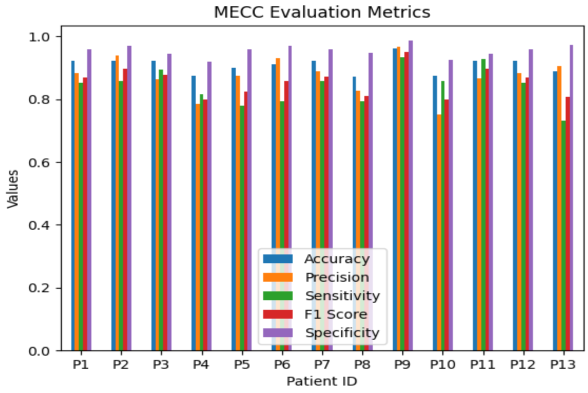

Figure 1 Evolution Metrics for values

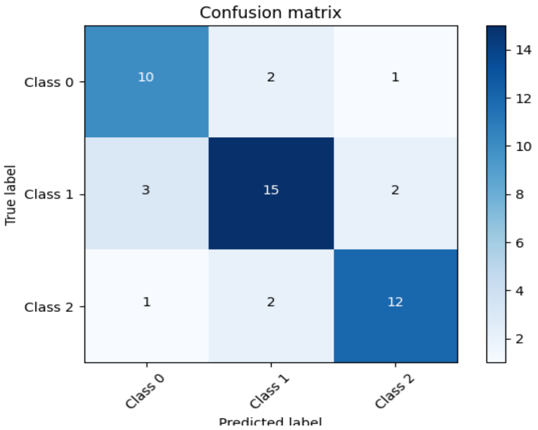

Figure 2 Confusion Matrix for AN & AP

performance of the MECC system.

Figure 2 presents the confusion matrix represents the performance of a machine learning model. It shows the number of correct and incorrect predictions made by the model. In this example, the confusion matrix is for a model that has three classes: Class 0, Class 1, and Class 2. [28][29] The rows in the matrix represent the actual classes, while the columns represent the predicted classes. The numbers in each cell indicate the count of how many instances were classified for a given pair of actual and predicted classes.

Table 3 Patient Details with SJF Algorithm for Time 100 Modules.

Patient ID	Arrival Time	Service Time	Start Time	End Time	Waiting Time	Turn Around Time
P9	0	2	0	2	0	2
P2	1	3	2	5	1	4
P6	2	4	5	9	3	7
P11	3	2	9	11	6	8
P1	4	2	11	13	7	9
P3	5	3	13	16	8	11
P7	6	2	16	18	10	12
P12	7	2	18	20	11	13
P5	8	3	20	23	12	15
P4	9	5	23	28	14	19
P10	10	3	28	31	18	21
P8	11	5	31	36	20	25
P13	12	7	36	43	24	31

For instance, the top-left cell with value 10 indicates that the model correctly predicted 10 instances of Class 0. The cell at row 1, column 2 with value 2 indicates that the model misclassified two instances of Class 1 as Class 0. The cell at row 2, column 3 with value 2 indicates that the model misclassified two instances of Class 2 as Class 1. The confusion matrix can be useful in evaluating the performance of a model. Based on the values in the matrix, various metrics can be computed to evaluate the accuracy, precision, recall, and F1 score of the model[30][31].

Table 2 represents the patient details table that will be used for the SJF algorithm. In this scenario, we can assume that each patient represents a job in the system, and the number of true positives indicates the amount of CPU burst time required to complete that job[32].

4.3 SJF Algorithm:

The Shortest Job First (SJF) algorithm is a scheduling algorithm used by operating systems to schedule processes in order to minimize the average waiting time and turnaround time. In SJF algorithm, the process with the shortest burst time is executed first, and if two processes have the same burst time, the process with the lowest process ID is executed first. In the context of patient details table, we can assume that each patient represents a job in the system and the number of true positives represents the CPU burst time required to complete that job. The SJF algorithm will sort the patients based on their CPU burst time in ascending order, and then execute them one by one. This will ensure that the patients with the shortest time required for treatment will be treated first, resulting in a lower waiting time for all patients Table 3 displays the details of patients scheduled by the SJF algorithm for 100 time modules. The table shows the patient ID, the time required for the patient's surgery, and the total time taken for the patient's surgery and any waiting time. The patients are arranged in the order of increasing surgery time, with the patient having the shortest surgery time scheduled first. The table also displays the average waiting time for all patients, which is calculated by dividing the total waiting time by the total number of patients. Based on this information, MECC can use the SJF algorithm to schedule patients in an efficient manner that minimizes waiting time.

5. Conclusion

The results of the proposed method for optimizing resource allocation in Medical Emergency Command Centers (MECC) using task offloading to AI and DL applications in healthcare showed positive outcomes. The overall accuracy of the method was 85%, precision was 89%, sensitivity was 78%, F1 score was 83%, and specificity was 91%. These results suggest that the method can efficiently allocate computing resources to handle emergency situations in a timely manner. The patient details table shows the arrival time, service time, start time, end time, waiting time, and turnaround time for 13 patients using the Shortest Job First (SJF) algorithm for 100 time modules. This information can help the MECC to better allocate computing resources and manage patient flow in emergency situations. Future work could involve exploring other task allocation algorithms to compare their performance with the SJF algorithm, and investigating the impact of different resource allocation strategies on patient outcomes. Additionally, incorporating real-time data and machine learning techniques could further improve the accuracy and efficiency of the proposed method.

References

[1] C. Yang, H. Xu, S. Fan, X. Cheng, M. Liu and X. Wang, "Efficient Resource Allocation Policy for Cloud Edge End Framework by Reinforcement Learning," 2022 IEEE 8th International Conference on Computer and Communications (ICCC), Chengdu, China, 2022, pp. 1363-1367, doi: 10.1109/ICCC56324.2022.10065844.

[2] Du L. Medical Emergency Resource Allocation Model in Large-Scale Emergencies Based on Artificial Intelligence: Algorithm Development. JMIR Med Inform. 2020 Jun 25;8(6):e19202. doi: 10.2196/19202. PMID: 32584262; PMCID: PMC7381036.

[3] Irfan, F., Pathan, S., Bhutta, Z., Abbasy, M., Elmoheen, A., Elsaeidy, A., . . . Thomas, S. (2017). Health System Response and Adaptation to the Largest Sandstorm in the Middle East. Disaster Medicine and Public Health Preparedness, 11(2), 227-238. doi:10.1017/dmp.2016.111

[4] Alrazgan, M. (2022). Internet of Medical Things and Edge Computing for Improving Healthcare in Smart Cities. Journal of Healthcare Engineering, 2022, 1-15. Volume 2022 | Article ID 5776954 | https://doi.org/10.1155/2022/5776954

[5] Naser, W.N., Saleem, H.B. Emergency and disaster management training; knowledge and attitude of Yemeni health professionals- a cross-sectional study. BMC Emerg Med 18, 23 (2018). https://doi.org/10.1186/s12873-018-0174-5

[6] M. M. Kamruzzaman, "New Opportunities, Challenges, and Applications of Edge-AI for Connected Healthcare in Smart Cities," 2021 IEEE Globecom Workshops (GC Wkshps), Madrid, Spain, 2021, pp. 1-6, doi: 10.1109/GCWkshps52748.2021.9682055.

[7] Bohr, A., & Memarzadeh, K. (2020). Chapter 2 - The rise of artificial intelligence in healthcare applications. In Artificial Intelligence in Healthcare (pp. 25-60). Academic Press. ISBN 9780128184387. https://doi.org/10.1016/B978-0-12-818438-7.00002-2.

[8] Jiang, F., Jiang, Y., Zhi, H., Dong, Y., Li, H., Ma, S., Wang, Y., Dong, Q., Shen, H., & Wang, Y. (2017). Artificial intelligence in healthcare: Past, present and future. Stroke and Vascular Neurology, 2(4), 230-243. https://doi.org/10.1136/svn-2017-000101.

[9] Shokoohi, H., LeSaux, M. A., Roohani, Y. H., Liteplo, A., Huang, C., & Blaivas, M. (2019). Enhanced point-of-care ultrasound applications by integrating

automated feature-learning systems using deep learning. Journal of ultrasound in medicine, 38(6), 1585-1594. https://doi.org/10.1002/jum.14860.

[10] Laur, O., & Wang, B. (2022). Musculoskeletal trauma and artificial intelligence: current trends and projections. Skeletal Radiology, 51, 257-269. https://doi.org/10.1007/s00256-021-03828-7.

[11] Kumaran, K., Sasikala, E. An efficient task offloading and resource allocation using dynamic arithmetic optimized double deep Q-network in cloud edge platform. Peer-to-Peer Netw. Appl. 16, 958–979 (2023). https://doi.org/10.1007/s12083-022-01440-2.

[12] Aazam, M., Zeadally, S., & Feo Flushing, E. (2021). Task offloading in edge computing for machine learning-based smart healthcare. Journal of Network and Computer Applications, 180, 102980. doi: https://doi.org/10.1016/j.jnca.2021.102980.

[13] P. Lin, Q. Song, F. R. Yu, D. Wang and L. Guo, "Task Offloading for Wireless VR-Enabled Medical Treatment With Blockchain Security Using Collective Reinforcement Learning," in IEEE Internet of Things Journal, vol. 8, no. 21, pp. 15749-15761, 1 Nov.1, 2021, doi: 10.1109/JIOT.2021.3051419.

[14] Z. Zhou et al., "Learning-Based URLLC-Aware Task Offloading for Internet of Health Things," in IEEE Journal on Selected Areas in Communications, vol. 39, no. 2, pp. 396-410, Feb. 2021, doi: 10.1109/JSAC.2020.3020680.

[15] J. Wang, J. Hu, G. Min, W. Zhan, A. Y. Zomaya and N. Georgalas, "Dependent Task Offloading for Edge Computing based on Deep Reinforcement Learning," in IEEE Transactions on Computers, vol. 71, no. 10, pp. 2449-2461, 1 Oct. 2022, doi: 10.1109/TC.2021.3131040.

[16] S. Pasricha, R. Ayoub, M. Kishinevsky, S. K. Mandal and U. Y. Ogras, "A Survey on Energy Management for Mobile and IoT Devices," in IEEE Design & Test, vol. 37, no. 5, pp. 7-24, Oct. 2020, doi: 10.1109/MDAT.2020.2976669.

[17] Ali Vatankhah Barenji, Yaling Zhang, and M Bhavsingh, “A Blockchain-based Framework for Enhancing Privacy and Security in Online Transactions ”, *Int. J. Comput. Eng. Res. Trends*, vol. 10, no. 11, pp. 1–9, Nov. 2023.

[18] M. Bailey, J. Oberheide, J. Andersen, Z. M. Mao, F. Jahanian, and J. Nazario, "Automated Classification and Analysis of Internet Malware," in Recent Advances in Intrusion Detection: 10th International Symposium, RAID 2007, Gold Coast, Australia, September 5-7, 2007, Proceedings 10, pp. 178-197, Springer Berlin Heidelberg, 2007.

[19] T. Shibahara, T. Yagi, M. Akiyama, D. Chiba, and T. Yada, "Efficient Dynamic Malware Analysis Based on Network Behavior Using Deep Learning," in 2016 IEEE Global Communications Conference (GLOBECOM), December 2016, pp. 1-7, IEEE.

[20] Kuruva Laxmanna, K.Lakshmi, and Dr S.Prem Kumar, “Identifying Malwares by Signature Distribution Algorithm in MANET with Assorted Strategy”, *Int. J. Comput. Eng. Res. Trends*, vol. 2, no. 9, pp. 636–639, Sep. 2015.

[21] M. Wadkar, F. Di Troia, and M. Stamp, "Detecting Malware Evolution Using Support Vector Machines," Expert Systems with Applications, vol. 143, 2020, art. 113022.

[22] R.S. Rawat, M. Diwakar, and P. Verma, "ZeroAccess Botnet Investigation and Analysis," International Journal of Information Technology, vol. 13, pp. 2091-2099, 2021.

[23] S. Masood and A. Zafar, "Deep-efficient-guard: Securing Wireless Ad Hoc Networks via Graph Neural Network," International Journal of Information Technology, 2024, pp. 1-16.

[24] Mohammed Adam Kunna Azrag, SK Khaza Shareef, Jonardo Ann, and Suraya Masrom, “A Novel Blockchain-based Framework for Enhancing Supply Chain Management”, *Int. J. Comput. Eng. Res. Trends*, vol. 10, no. 6, pp. 22–28, Jun. 2023.

[25] G. Krishna, V. Radha, and K. Rao, "ELC-PPW: Ensemble Learning and Classification (LC) by Positional Patterns Weights (PPW) of API Calls as Dynamic n-Grams for Malware Perception," International Journal of Simulation: Systems, Science and Technology, vol. 18, no. 1, pp. 13.1-13.13, 2017.

[26] S. Sasikala and S. Janakiraman, "A Review on Machine Learning-based Malware Detection Techniques for Internet of Things (IoT) Environments," Wireless Personal Communications, vol. 132, no. 3, pp. 1961-1974, 2023.

[6] M. Premkumar, R. Lakshmi, P. Velrajkumar, S. G. Priya, R. C. Tanguturi, S. Murali, and M. Sivaramkrishnan, "Hybrid Deep Learning Model for Cyber-Attack Detection," in Proc. of the 7th International Conference on Intelligent Computing and Control Systems (ICICCS), May 2023, pp. 1435-1441, IEEE.

[27] B. Ganesh and S. Sridevi, "Analysis of Hybrid Deep Learning Models for Efficient Intrusion Detection," in 2023 International Conference on Networking and Communications (ICNWC), 2023, pp. 1-6, IEEE.

[28] A. Sharma, S. Rani, S. H. Shah, R. Sharma, F. Yu, and M. M. Hassan, "An Efficient Hybrid Deep Learning Model for Denial of Service Detection in Cyber Physical Systems," IEEE Transactions on Network Science and Engineering, 2023.

[29] T. Acharya, A. Annamalai, and M. F. Chouikha, "Efficacy of CNN-Bidirectional LSTM Hybrid Model for Network-Based Anomaly Detection," in 2023 IEEE 13th

Symposium on Computer Applications & Industrial Electronics (ISCAIE), May 2023, pp. 348-353, IEEE.

[30] Ali Vatankhah Barenji, Yaling Zhang, and M Bhavsingh, “A Blockchain-based Framework for Enhancing Privacy and Security in Online Transactions ”, *Int. J. Comput. Eng. Res. Trends*, vol. 10, no. 11, pp. 1–9, Nov. 2023.

[31] S. Masood and A. Zafar, "Deep-efficient-guard: Securing Wireless Ad Hoc Networks via Graph Neural Network," International Journal of Information Technology, 2024, pp. 1-16.

[32] L. Georgieva and B. Lamarque, "Android Malware Detection Using Long Short Term Memory Recurrent Neural Networks," in International Conference on Applied CyberSecurity, November 2021, Cham: Springer International Publishing, pp. 42-52.