Hindawi
Mobile Information Systems
Volume 2022, Article ID 9952993, 14 pages
https://doi.org/10.1155/2022/9952993

Research Article

Secured Optimized Resource Allocation in Mobile Edge Computing

Asma Bibi,[1] Muhammad Faran Majeed,[1] Sikandar Ali,[2] Irshad Ahmed Abbasi,[3] Ali Samad,[4] and Samad Baseer[5]

[1]*Department of Computer Science, Kohsar University Murree, Punjab, Pakistan*
[2]*Department of Information Technology, The University of Haripur, Haripur 22620, Khyber Pakhtunkhwa, Pakistan*
[3]*Department of Computer Science, Faculty of Science and Arts, Belqarn, Sabt Al-Alaya 61985, University of Bisha, Saudi Arabia*
[4]*Faculty of Computing, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan*
[5]*Department of Computer System Engineering, University of Engineering and Technology Peshawar, Peshawar 25000, Khyber Pakhtunkhwa, Pakistan*

Correspondence should be addressed to Sikandar Ali; sikandar@cup.edu.cn

Received 23 April 2022; Revised 22 May 2022; Accepted 25 May 2022; Published 21 August 2022

Academic Editor: Le Sun

Mobile edge computing (MEC) uses multiple mobiles to compute several complex tasks that are unable to compute on a single device. Taking advantage of all abundant mobile resources and making a mobile cloud from them will be useful. This study aims to propose and implement a novel framework to cover challenges raised by application execution on resource-constrained devices. The purpose is to overcome the waste of resources in MEC and provide time efficiency to the data packets that are sent and received between offloaded and offloading devices. The main task of MEC is to offload tasks by first selecting resources and then allocating tasks to selected resources. A multiple linear regression algorithm is used for the selection of compatible devices. Particle swarm optimization techniques are used as a benchmark technique to design an algorithm for optimized resource allocation. Multiple mobile devices acted as a major component for making edge clouds. The study finds that response time of processing tasks is reduced, ineffectual resources become beneficial, increase in demand for mobile devices, and usage of mobile resources as a replacement for mobile edge cloud servers. Abundant resource usage of two or more edge clouds, that is, interedge resource usage is the originality of this research, while others are using intraedge resources only.

1. Introduction

The recent increase in demand for mobile devices and the use of the cloud as virtual storage demanded the field to evolve, named mobile edge computing (MEC). Research in MEC is performed by load balancing and offloading. To decrease resource demand in MEC architectures, a pattern of traditional clouds is used. The majority of the MECs do not consider optimization and cost factors. MEC also provides the network to use the resources that have less memory, time, and energy consumption.

MEC provides the opportunity to reduce latency while offloading tasks in the network. MEC allows the resources to offload tasks easily and safely in the network. MEC tries to reduce the consumption of power and energy. It removes all the delays from the network. MEC designs the nodes to offload tasks in the network to remove all the delays from the network. MEC allows the nodes to get information about the other nodes that are offloading data in the network so there will be no collision. It allows nodes to use all the resources from the network. Linear programming is the most common approach to optimize an objective function, for example, to reduce resource consumption, reduce the total execution time, reduce latency, or increase the quality of experience.

One of the recent approaches to solving the offloading problem is by using deep learning [13]. Table 1 shows an overview of different surveys conducted in the field. Limitations of the surveys are also mentioned in the table.

Table 1: Survey-based analysis.

Year [paper]	Topic discussion and overview	Limitation
February 2019 [1]	Machine and deep learning techniques are mostly used in papers of this survey. Machine and deep learning are used to detect the encryption traffic in offloading tasks in the network.	There are grand challenges in edge computing security.
March 2017 [2]	Network virtualization and Lyapunov optimization-based dynamic computation offloading (LODCO) algorithm is used. Network virtualization manages the flexibility of the virtual representation provided by the MEC.	Dynamic resources are added while offloading tasks in the network.
December 2020 [3]	Markov-based, heuristic, and metaheuristic algorithms are used. Markov-based techniques reduce the time and redundancy while offloading tasks in the network.	Resources that are added to offload tasks in the network increase delay.
January 2017 [4]	Machine learning advanced communication techniques are used to offload tasks in the network.	Slow processing increases time delay.
June 2017 [5]	Advanced communication techniques are used.	Refraction and reflection increase delay.
May 2018 [6]	Migrating running service technique and compression algorithm are used. Migrating running service technique is used for migrating the task in the network.	The size of the task is not reduced, so there is a delay in the network.
2007 [7]	IDS is used to interpret the traffic while offloading tasks in the network.	The cost factor is not considered
2007 [8]	The pushback technique is used to aggregate the traffic while offloading tasks in the network.	A comparison of complexity analysis is not performed.
2014 [9]	RTT communication and task scheduling algorithms are used. RTT communication is used to handle the traffic while offloading data in the network and divide the task in the node to remove the delay.	Processing delay is not entertained.
2020 [10]	VM migration and genetic algorithm are used. VM migration is used to migrate the task in the network and makes the performance better.	There is delay in resource consumption.
2017 [11]	The virtual machine is used to migrate the task virtually in the network and make the performance better.	There is time delay.
2015 [12]	Lyapunov optimization and online control algorithm are mostly used in the research paper of this survey. Lyapunov optimization is used to run the online program to offload tasks in the network.	Maximization of resource usage is not performed.

A gap in the [14] study is that it does not consider offloading in the large-scale network, while the quality of offloading tasks in the network is not considered in [15]. Safety is not considered in [16], while task execution is difficult for the user in [17, 18]. Modern facilities to use are not well-thought-out [19]. The disadvantage in [20] is that it does not consider if any virus destroys this data, then what will be sent next. The negative of [21] is that it does not consider the multiple attacks solution. The undesirable thing about [22] is that it does not consider the solution of inputting tasks easily for the user. The ploy of [23] is that it does not consider the offloading task in the network easily and more securely. The depraved thing of [24] is that it does not consider more instructions to make the system more secure. The downside of the study is that it does not consider cryptography techniques to make the network more secure [25].

Table 2 presents the critical analysis.

The rest of the paper is arranged as follows. Section 2 contains a literature review. Section 3 states the problem statement while Section 4 elaborates on the proposed solution. Section 6 concludes the paper with a proper future direction.

2. Literature Review

The authors of [26] found that cloudlets offload the task easily by using the techniques of DOTA, CBL, and FATO. Mobile devices request for the transmission of information by making sure the message is in the network so there will be no collision.

MCC provides the facility to store large data over the network, and it also ensures the sending of large amounts of data on the network. Mobile edge computing improves the speed of the node by removing the slow node PCs from the network and adding the neighbor PCs. It is found in the paper that a single node makes the network so busy that it increases. DOTA, CBL, and FATO are used for dividing tasks in cloudlets. The limitation is that less energy consumption having less cost technique is not developed. It also provides good efficiency to the nodes in the network. So the nodes will offload tasks in the network with good quality. SDN provides the facility for MEC to divide the task into nodes. The nodes will offload tasks in the network sequentially. So there will be no loss of data in the network. MEC also provides the service of offloading tasks in the wireless network [51].

Wang et al. proposed a new architecture for computation and storage offloading based on fog computing and found that COCA offloads the task from the smartphone to the fog server [27]. The result was deduced that with the enactment of the cloud upgrade, uploading data became fast. In this research, no technique was used for uploading large amounts of data. More resources must be added due to the hinging of the network.

In [28], local computing (mobile device) combined with the computing system and found system loss function (SYLF) minimization problem. Markov

Table 2: Critical analysis.

Year [paper]	Technique used	Proposed technique	Results	Limitation
May 2019 [26]	Deployed cloudlets used for dividing the resources in k nodes	A single node makes the network so busy that increase. DOTA, CBL, and FATO are used for dividing tasks in cloudlets.	Cloudlets offload the task easily by using the techniques of DOTA, CBL, and FATO.	Less energy consumption having less cost technique is not developed.
October 2018 [27]	New architecture for computation and storage offloading based on fog computing	New architecture for computation and storage offloading based on fog computing	COCA offloads the task from the smartphone to the fog server.	The enactment of the cloud upgraded, uploading data become fast.
August 2020 [28]	Local computing (mobile device) that combines with the computing system	Local computing (mobile device) combines with the computing system.	System problem loss function (SYLF) minimization problem	QLCOF scheme effectively reduces the SYLF.
2017 [29]	Mixed integer programming	NP-hard problem, EcoMD	EcoMD provides an improved performance.	None
November 2020 [30]	Elliptic curve cryptosystem.	Pairing-free multiserver authentication protocol	Secure mutual authentication, anonymity, and scalability are achieved.	None
May 2020 [31]	Comparison technique (proposed MUMACO with benchmark)	Offloading of all applications is done to the cloudlets, but a fraction of cloudlets is idle.	Time consumption, energy consumption, and load balancing are optimized.	Multiobjective is performed. Optimization cannot be performed
January 2020 [32]	Offloading algorithm (hybrid intelligent optimization algorithm)	Optimization of task delay and resource consumption	The proposed algorithm effectively improves the offloading utility as compared to baseline.	Offloading in the uncertain network is not available.
March 2020 [14]	Comparison and optimizing technique (offloading)	Performance and energy of mobile device can be improved by edging.	Proposed HIQCO provides accurate results and then compared the algorithm.	Storage cannot be considered in the comparison to HIQCO and baseline algorithms.
August 2019 [33]	MCOWA technique used for uploading tasks on the network easily	By using MCOWA technique, algorithm problem is solved as it solved the complexity of the network.	Time and energy are consummated so the network becomes fast.	None
2018 [34]	Analysis technique used for the scalability and performance of an edge cloud system	Interedge is unchanged. Bandwidth should remain.	If capacity is added to the existing edge network deprived of increasing the interedge bandwidth, then it will pay for networkwide congestion.	Increasing distance and low bandwidth will increase the load.
March 2020 [35]	Cloud modeling operator introduced that deals with the execution of packets in the network	By using this strategy, the performance of computing resources improves.	Time and energy are consummated. Also, improve the utilization of computing resources and ensure the QoS, and this is critical to edge-cloud computing business models.	The problems such as management resources of MEC and the cloud are not improved and considered.
April 2020 [36]	FL technique introduced for round communication between the nodes in the network	The nodes will only send a message from one node to other when they receive a message that the network is free.	By using the FL technique, the network becomes safe from the collaboration of messages. Thus, the packets will not be lost.	Privacy is not considered and improved to make the network secure.
April 2020 [37]	GPS technique used for measuring frequency and sending the signals even from the satellite	GPS technique is used for sending the signals from satellite to the user. So the user can easily send a message from Earth to satellite, and vice versa.	The offloading task increases if the user sends a message to the satellite. There will be no interruption of other networks as the nodes only send one packet at one time.	No technique is used to make the signal powerful as if the signal is weak, the packet will be lost.
January 2021 [38]	DECCO technique used for maintaining signals from a long distance	Long distance creates the network slow. Thus, the packet delay. Energy and power consumption.	DECCO used that maintain the long distances plackets. Thus energy, time, power and quality consumption.	Many computing capabilities are not considered.

Table 2: Continued.

Year [paper]	Technique used	Proposed technique	Results	Limitation
September 2019 [39]	Mobile edge computing used	The network becomes secure and fast, and good qualities are available in the network.	The network becomes fast; delay is removed from the network.	Sending a large amount of data is not considered.
September 2019 [40]	Edge-centric IoT used that is responsible for offloading data safely in the network	Security is very important for offloading tasks in the network.	If there is no security in the network, then the delay will occur, and the data can also be hijacked.	If data is hijacked and caught by a virus, then no technical solution is considered here.
January 2019 [41]	GMaxEOQU and GMinEOIP used in the network	GMaxEOQU and GMinEOIP are used to minimize the quality errors in the network.	If there are less nodes present in the network, then there will be a delay in the network.	Offloading times by multiple nodes are not considered.
February 2020 [42]	QMPOS technique used in the network	QMPOS technique is used to derive the result in the network and evaluate the performance of VN.	By balancing the load of VN, the task will offload within time. The network becomes burdenless.	The cost of VNs is not considered.
June 2020 [43]	Fog computing used in the network	Fog computing provides management, security, and availability of resources that helps offload tasks.	Offloading tasks becomes too easy and secure. Nodes will get many resources for offloading tasks in the network.	The cost of resources is not considered.
June 2020 [44]	SMSC and RAMWS used in the web servers	SMSC works to control the requests that arrive on web servers, and RAMWS works to overcome the request time out in the web server.	The web server provides the resources to the users to use the resources and offload their tasks. It also provides the user the facility to get information from the websites.	Protection of web servers is not considered.
August 2020 [41]	Skippy technique used	Serverless is used in the network that provides all resources to offload the task in the network, and Skippy helps serverless do this.	A large amount of data is able to send in the network by using serverless.	If any unauthorized network hacks the data, then a large amount of data will have lost in the network.
March 2019 [45]	Pervasive technique used	Pervasive helps the computer provide all resources to the nodes to the nodes offload their task.	Pervasive helps the user find anything from the computer by using it. It also provides the user to interact with the computer easily.	Security is not considered.
August 2018 [46]	Routers used in the wireless network	Routers develop the communication between the two networks and make communication possible.	Wireless network also provides the facility to the nodes and make communication easy like space.	Cost is not considered.
December 2016 [23]	Load balancing (virtual machines) Apache JMeter (tool)	The majority of MCC do not consider cost factors. For multiple users, one virtual machine architecture is most suitable.	23 times task execution increases and 2/3 resource usage decreases.	Execution time is more for projected architecture.
2020 [14]	Quality of service approach	Mobile edge computing	Low energy consumption and low delay	Do not consider offloading data on large scale.
2020 [15]	EOESPA, RNOESPA	PD-NOMA-SCA, PD-NOMA	Distribute tasks in the nodes of the network.	Do not consider the quality of offloading data.
2020 [16]	Cryptography approach	RSA	Secure mobile devices by keys and passwords	Do not consider secure Internet.
2017 [17]	Reputation-based resource allocation approach	Genetic algorithm	Offload tasks safely in the network.	Do not consider the easy way of the offloading task.
2017 [19]	Virtual cryptography	Zero watermarking algorithm	Make data safe and able to update from time to time.	Do not consider modern facilities.
2019 [20]	Elliptic curve cryptography	Authentication protocol	Encrypt and decrypt data same as the user want.	Do not consider sending data next if the virus destroys it.
2021 [21]	GR cryptography	Lightweight cryptography algorithm	Make the thing download easily and safely from the Internet.	Do not consider multiple attack solutions.

Table 2: Continued.

Year [paper]	Technique used	Proposed technique	Results	Limitation
2020 [47]	P2P cryptography currencies	Blockchain and task offloading techniques	Encrypt and decrypt data exactly in the network.	Do not consider secure edge computing.
2021 [22]	AES-based cryptography approach	Deep reinforcement learning based on an online algorithm	Remove delay from the thing download and access from the Internet.	Do not consider inputting task solutions.
2020 [23]	Cryptography hash	SDN	Divide task in the nodes so no collision will happen.	Do not consider a secure and easy offloading task.
2019 [24]	AES-based cryptography approach	FPGA	No collision will happen to destroy the data.	Do not consider a more secure system.
2020 [25]	AES-based cryptography approach	LLCA	Encrypt and decrypt data exactly in the network.	Do not consider more cryptographies to make the network more secure.
2020 [48]	AES-based cryptography approach	LSM	Protect the data of user from not being able to hack for the other person.	Do not consider an easy and secure offloading task network.
2020 [49]	AES-based cryptography approach	RSA	Criminal record update from time to time	Do not consider more functions to offload tasks in the network.
2021 [50]	Blockchain	ACO algorithm	Make the system more secure to offload data in the network.	Do not consider more systems to make offloading tasks in the network more secure and easy.

decision process (MDP) designed a state loss function (STLF) to measure the quality of experience. In it, multioperator multiverge cloud state was not considered. Slow nodes must remove because multiusers increase the cost. Mobility management is the reason for the disconnected link between the devices and the edge network. It manages horizontal and vertical mobility. Heterogeneity deals with the wireless network interface, for example, Wi-Fi. Low delay and high bandwidth are the main challenges. Decrease in price by adding neighbors to offload tasks early is the main challenge. MEC provides the service of offloading tasks in the network by using the Internet. MEC allows users to download anything from the Internet keeping the security in the mobile devices. MEC provides the service of using passwords on mobile devices. It provides the facility of storing data on the Internet so that when the user accesses the Internet, he will easily access the information without any delay. MEC provides biometric security so that when the user enters his data in biometrics, his data will remain safe and will not leak to anyone. MEC makes it possible that when the person will enter his fingerprint, all his data will come out. This data will not be accessible to anyone because MEC provides security. MEC provides security to the nodes when the nodes will offload data in the network. The data will be safe and will not be disclosed to anyone. MEC provides security to the nodes by using passwords and keys that will not be shared with anyone.

In [29], the authors used mixed integer programming to find NP-hard problems and EcoMD. EcoMD provides improved performance in terms of resources. But resources must be stable because increasing nodes will increase the cost. However, there are some other solutions as well that do not fit our study [52–55].

The authors of [30] used the elliptic curve cryptosystem and MSA protocol for the MCC environment to find a pairing-free multiserver authentication protocol and achieved secure mutual authentication, anonymity, and scalability, but there was no mechanism of security proposed in it.

In [31], the authors used the comparison technique (proposed MUMACO with benchmark) to find offloading of all applications to the cloudlets, but a fraction of cloudlets was idle. Time consumption, energy consumption, and load balancing were optimized but multiobjective optimization cannot be performed, and less cost and energy consumption resources must be used.

In [32], the authors proposed offloading algorithm (hybrid intelligent optimization algorithm) and found optimization of task delay and resource consumption. The proposed algorithm effectively improves the offloading utility as compared to the baseline algorithm, but offloading in an uncertain network is not available [56].

In [14], the authors introduced a cloud modeling operator that deals with the execution of packets in the network by using this strategy; the performance of computing resources improves. Also, they improve the utilization of computing resources and ensure the QoS and thus are critical to edge-cloud computing business models [57].

In [33], the FL technique was introduced for round communication between the nodes in the network. The nodes will only send messages from one node to another when they receive a message that the network is free. By

using the FL technique, the network becomes safe from the collaboration of messages. Thus, the packets will not be lost. Privacy is not considered and improved to make the network secure.

In [34], the GPS technique is used for measuring frequency and sending the signals even from the satellite. GPS technique is used for sending the signals from satellite to the user. So the user can easily send messages from Earth to satellite, and vice versa. The offloading task increases if the user sends a message to the satellite. There will be no interruption of other networks as the nodes only send one packet at one time. No technique is used to make a signal powerful as if the signal is weak, the packet will be lost. MEC does the encryption and decryption task in the network. The nodes will encrypt data in the network. MEC makes this task possible that regardless of what the user sends for encryption, the network will decrypt the same data in the network without any delay. MEC also makes the money transaction possible and safe by using the keys such as ATM keys and PINs. The PIN is only known by the user who uses the ATM. Thus, the data and the money will be safe. MEC provides security to the criminal record. If the person does any crime, then the record will be written in the file. This file will not leak to anyone and will be updated from time to time. It is possible due to MEC. MEC provides the security and privacy for storing data in the network that when the user wants to access the data, MEC makes the task present in the network. This removes the delay from the network to access the network.

In [35], the authors used the DECCO technique for maintaining signals from a long distance. Due to long distance, the network becomes slow resulting in high packet delays, energy, and power consumption. DECCO maintains the long-distance packets. Thus energy, time, power, and quality consumption are achieved. Cloud servers are far away from mobile devices that create signal issues, so the resources become weak.

The authors in [36] used edge-centric IoT that is responsible for offloading data safely in the network. Its security is very important for offloading tasks in the network. If there is no security in the network, then the delay will occur, and the data can also be hijacked. If data is hijacked and caught by a virus, then no technical solution is considered here. If there is no security and privacy in the network, then the data will create delay and be hijacked. The network must be protected by passwords, and the password must be secure. The password is not shared by anyone [58, 59].

In [37], the authors used GMaxEOQU and GMinEOIP in the network, and they are used to minimize the quality errors in the network. If there are less nodes present in the network, then there will be a delay in the network. Offloading time by multiple nodes is not considered. If there are less nodes used in the network, then the delay will occur [60, 61].

The authors in [38] used the Skippy technique. The server is less used in the network that provides all resources to offload the task in the network, and Skippy helps serverless do this. A large amount of data is able to be sent to the network by using serverless. If any unauthorized network hacks the data, then all large amounts of data will have been lost in the network. The data must be protected by using a password, and some keys so a large amount of data will be safe.

In [39], the authors used the pervasive technique. Pervasive helps the compute provide all resources to the nodes to offload their task. If any unauthorized user hacks the data, then it will give the wrong information and data to the users.

In [40], the authors used load balancing (virtual machines) Apache JMeter (Tool). The majority of MCC do not consider cost factors. For multiple users, one virtual machine architecture is most suitable. The time to execute a task increases by 23 times while the resource utilization decreases by two-third. Execution time is more for projected architecture [51].

In [41], the authors discussed that today, the Internet is too common, while using Internet security is also needed to offload tasks from the Internet, download anything from the Internet, and so on. Also, information on the Internet must be secure so that the user can access it at any time and access it without delay.

AES is a part of the block symmetric cipher. AES provides the facility for MEC to offload the message in the network by using nodes. AES also provides the facility to encrypt and decrypt the data in the network without creating any delay in the network. AES has the ability to use different keys such as 128, 192, and 256 bits. Each bit has different features [42]. AES is required in every field where security is needed. AES provides the encryption and decryption of data from one field to the other field easily and without delay in the network [62].

In paper [44], cloud computing provides the security to the user to offload tasks in the network. AES provides the facility of encryption and decryption in the cloud computing network. AES provides the facility of security to the network to exchange information without any delay [43]. As today the Internet is too common, while using the Internet, security is also needed to offload tasks from the Internet, download anything from the Internet, and so on. Also, information on the Internet must be secure so that the user can access it at any time and access it without delay [41]. AES divides the task into two portions, that is, one is the offloadable and the other is the un-offloadable program so the offloadable task can be offloaded easily in the network without any delay. AES provides some security and divides tasks into the un-offloadable task so the task can easily offload in the network without delay [45]. AES provides the security to the users to offload tasks in the network without any interference from a virus or delay in the network. AES helps the user provide antivirus to the user so the virus will not attack and destroy the user data and offload in the network without delay [46]. Edge computing became top popular as it removes the delay from the network while offloading data using the Internet. Also, it makes all Internet applications secure for offloading data. While offloading the data using the Internet, the drastic event that can happen is an attack by an attacker. Malicious attackers do the collision in the information present in the network. The collision in

the information makes the information lost; thus, the data are lost and destroyed because of malicious attacks. These attacks happen due to the recovery of the secret key of AES [63]. All the internal collisions are detected by the AES, but the linear collision is not detected by the AES. S-box gives the output that talks about the collision happening internally [23].

The issue with the proposals and techniques discussed above is that during the offloading process of the data from the Internet, there can be malicious attackers who can intervene in the communication and perform different kinds of attacks. To cope with these issues, we devise a mechanism through MEC that can help mitigate such attacks. Also, the response time, resource utilization, and fair usage of mobile devices are increased.

3. Our Contributions to the Field

Design and implementation of a novel task placement framework that does the following.

(1) Reduces response time of processing tasks,

(2) Un-usable resources become useful,

(3) Demand of mobile device will increase, and

(4) Usage of mobile resources as a replacement of cloud servers.

4. Problem Statement

When MEC requests to buy computer resources while executing a task, it faces a delay in request and response to and from MEC, so this delay increases time. Similarly, many mobile resources were being wasted by users despite having 4 to 8 GB RAM and 128 GB plus storage.

In this section, we briefly state the three main problems to highlight the problem scenario and drawbacks that can occur due to these problems.

4.1. Problem I: Utilization of Mobile Idle Resources. MEC provides the facility of low delay rate, low cost, and high efficiency of offloading tasks in the network. So mobile resources can be used to make a local edge cloud for working in an efficient manner.

4.2. Problem II: Task Execution Delay. When MEC requests to buy computer resources while executing a task, it faces a delay in request and response to and from MEC and computer resources. This paper is basically solving a problem as per a scenario in which a user wants to process a huge amount of data at that time and users have mobile devices with either them or with their friends, so users can make the local cloud without having remote servers.

The following research questions are formed from the above problems:

(1) Question 1: How to reduce the response time of processing by making an edge instead of waiting for a single device?

(2) Question 2: How to save wastage of resources on mobile, and how will they be utilized in a timely and effective manner?

5. Proposed Solution

The particle swarm optimization technique is used and modified according to MEC requirements to gain efficiency by finding the optimal nodes, which will be used in the MEC. To find the best node, we will check its previous record of connecting time delay and its distance from its master device. We will provide a list of nodes to the swarm algorithm, and it will compare the first node with other nodes and will place the node having less connecting time and a short distance from the master device. At the first index, the comparison will continue till we sequence the list in the best node in ascending order in the FCFS list position. So we will have the best device and the best mobile edge for task execution. To overcome the problem of time delay, it is better that the MEC server should remain connected with MEC clients so that the delay of connection would not appear when a task appears as MEC is already connected so it will start executing the task without having the connection delay. To overcome the problem of resource wastage, the solution is to use mobile resources if it is available and ready to use. MEC uses multiple mobiles to compute multiple complex tasks, which are nearly impossible to compute on a single device.

This study aims to propose and implement a novel framework to cover challenges raised by application execution on resource-constrained devices. Two main tasks that the proposed solution is performing are task allocation and task execution. Breakdowns of these tasks are given below.

5.1. Task Allocation

(1) Secured resource discovery of mobile devices for connecting to edge servers

(2) Secured resource allocation algorithms used for checking device capability (whether the device is capable of executing the task); optimization methods will be used

(3) Secured resource allocation algorithms used for making an effective offloading communication that will make sure that offloading resource communication is secure

(4) Transfer of data from a mobile device to the edge nodes

5.2. Task Execution

(1) Scheduling of tasks at the edge nodes by the offloaded device

(2) Offloading of task by the offloaded device (sending task)

(3) Transfer of results back to the source mobile device

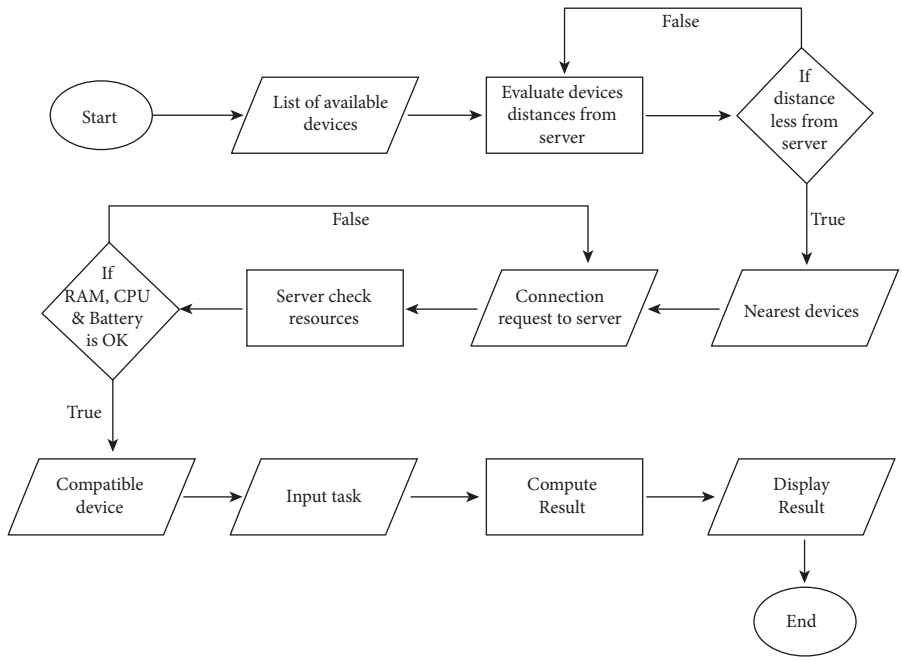

Figure 1: Flowchart of optimized resource allocation in MEC.

(4) Edge server that will gather results and perform integration of it

5.3. Flow Diagram. Figure 1 shows the flow diagram of the scheme.

5.4. Algorithms. Algorithms 1 and 2 represent problems I and II, while for optimization, we present Algorithm 3.

5.5. Implementation Steps. The following steps are performed while implementing the proposed solution:

(1) We will make a connection between offloader and offloadies devices using a nearby API and distribute the task in form of bytes. There are strategies such as P2P, P2Cluster, and others to create a connection among the devices. We will be using connection P2Star because it suits our scenario. P2Star will offload tasks in the local cloud more quickly than other connections strategies. Algorithms are used in this connection for making the handling of the offloading tasks better.

(2) After choosing a strategy, the offloader device will start discovering the offloadies, and offloadies will start advertising so that they are discoverable, and a connection can be established among the devices.

(3) Now as offloadies are discoverable, offloader will start connecting with the offloadies one by one and accept their connection of offloadies to work as a slave for the master device and to compute the task provided by the offloader devices and send back the results.

(4) After establishing the connection between the offloadies, these devices will send their specification information and available resource information so that the offloader can decide which devices are capable of serving the master device and which offloadies are not capable.

(5) After discovering the ability of devices on the basis of RAM, CPU, battery, and available RAM, the offloader will filter the devices and ignore the rest of the devices [56].

(6) Now, the offloader will split the task into a number of available devices and send the task to available devices for the sake of processing. In our case, the task is image processing; however, it also depends upon the application requirements of the user [64–66].

(7) The offloadies will process the image processing task on their end using the OpenCV library using their own power and processing power.

(8) After processing, each offloadies will send back the result to the offloader device, and the offloader will use that result for its own use.

5.6. Main Function of Code in Kotlin Language

(i) fun on ConnectionResult (endpointId: String?,

(ii) result: ConnectionResolution) {

(iii) when (result.getStatus ().getStatusCode ()) { ConnectionsStatusCodes.STATUS_OK-> {

(iv) var devices:SlaveDevices = ArrayList < SlaveDevices> var nearestDevices = ArrayList < SlaveDevices>

(v) var compatibleResources = ArrayList < SlaveDevices > devices.forEach (){

```
(1)  Buy Resources (Br), MEC Task (Mt)
(2)  Input (Br)
(3)  Output: {true}
(4)  Input: {Mt}
(5)  Calculate the MEC Task using ComputeTask (Mt) Send
(6)  Result of MED
(7)  Output {Result}
```

Algorithm 1: Algorithm for problem I.

```
(1)  Device RAM Memory (Rm), Result of Offloadies (Rs),
(2)  Energy (E), FCaompatibleDevice () is a function in Android to check for RAM,
(3)  Compatible Device (Cd), CPU, and Device minimum requirements.
(4)  Input: (Cr)
(5)  for each MEC Server do
(6)     Output: {Rm, E, CPU}
(7)  end for
(7)  for each Device d do
(8)     Calculate the Compatibility check using FCompatibleDevice (), add the device to the Cd list
(9)  end for
(10) for each Cd do
(11)    Rs = Output {task}
(12) end for
```

Algorithm 2: Algorithm for problem II. Connection Request (Cr), Task (T),

```
(1)  DL = device list, D = device, Nd = nearest devices, NCR = connection request of nearest device, CD = compatible device, and
     T = task
(2)  input {DL}
(3)    for each DL do
(4)      Calculate the distance of D
(5)      if D {i}.Distance < D{I + 1}.Distance then
(6)        ND.add (D {i})
(7)      end if
(8)      if (i < DL.size() then
(9)        Move to Step 3
(10)       Output {ND}
(11)     end if
(12)     MES Input {NCR}
(13) end for
(14) for each ND do
(15)     if ND {i}.ram>3 GB && ND {i}.CPU>2.4 GHz then
(16)       CD.add (ND {i})
(17)     end if
(18)     if i< CD.size() then
(19)       Move to Step 9
(20)     end if
(21)     Output {CD}
(22)     Input {T}
(23)     Calculate Result using function Compute_Result ()
(24)     Output {Result}
(25)     end for
```

Algorithm 3: Algorithm for optimization.

Table 3: Results without optimization.

Device number	Number of images	Time (seconds)	Memory (MB)	CPU percentage (%)	Network (Kb)	Energy
1	2	4	54	7	0	Light medium
3	2	2.5	30	3.5	0	Light low
N	2	$N+1.5$				

Table 4: Results of the proposed solution (after optimization).

Device number	Number of images	Time (seconds)	Memory (MB)	CPU percentage (%)	Network (Kb)	Energy
1	2	3	32	2	0	Light low
3	2	1.5	20	2.5	0	Light low
N	2	N				

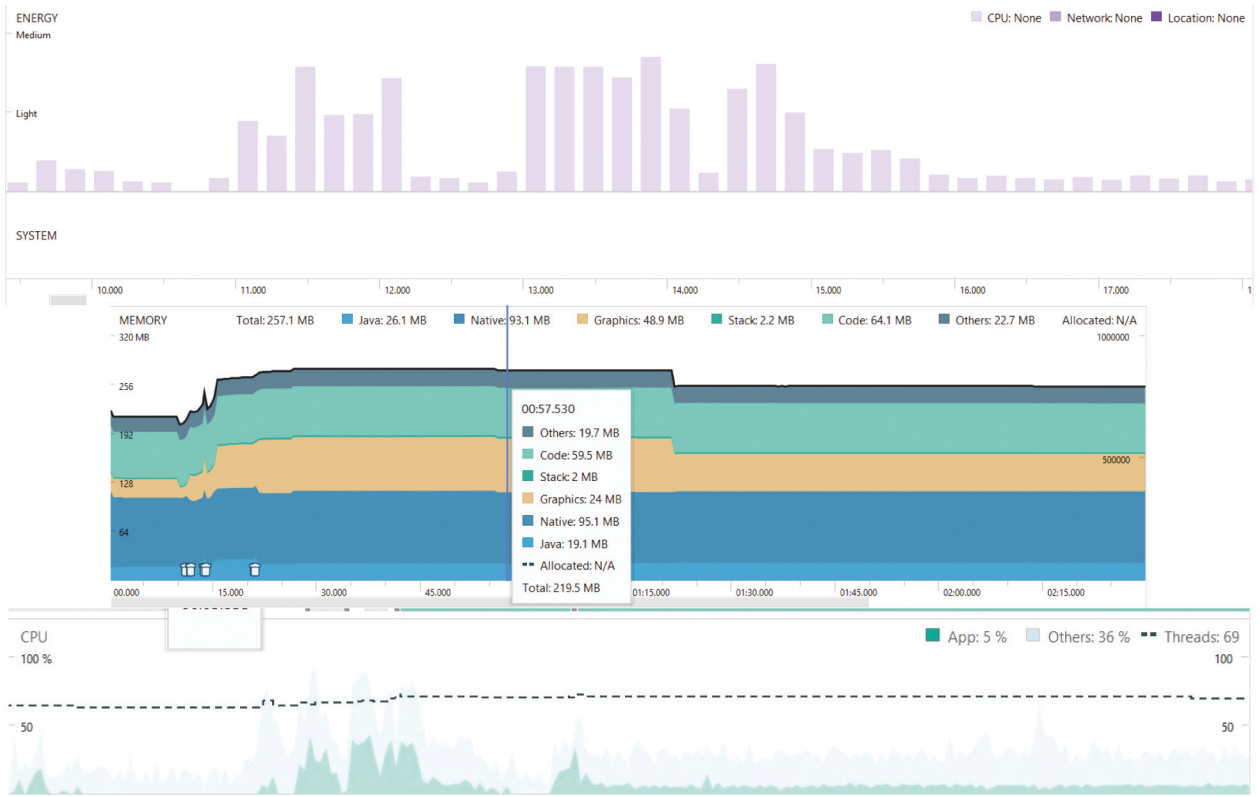

Figure 2: Results without optimization.

(vi) if (device[i].distance < device.[$i+1$].distance&&device [i].communicationdelay < de vice.[$i+1$].communicationdelay){

(vii) nearestDevices.add (device)}} nearestDevices.forEach {

(viii) if (nearestDevices [i].ram > 3 && nearestDevices [i].cpu > 24) { compatibleResources.add (nearestDevices)}}

(ix) var noofimages = 0; noofimages = imageList.size/compatibleResources.size compatibleResources.forEach {

(x) var list = imageList.subList (noofimages) imageList.removeAll (list) SendTask (list)}}

(xi) ConnectionsStatusCodes.STATUS_CONNECTION_REJECTED -> {} ConnectionsStatusCodes.STATUS_ERROR -> {}

(xii) else -> {}}}.

6. Simulation and Results

This section thoroughly describes the equations, simulation, and results of the study.

6.1. Equations. The following generalized equations are formed:

$$R_{s,t} = R_{c,t} + U_{s,t}, \quad (1)$$

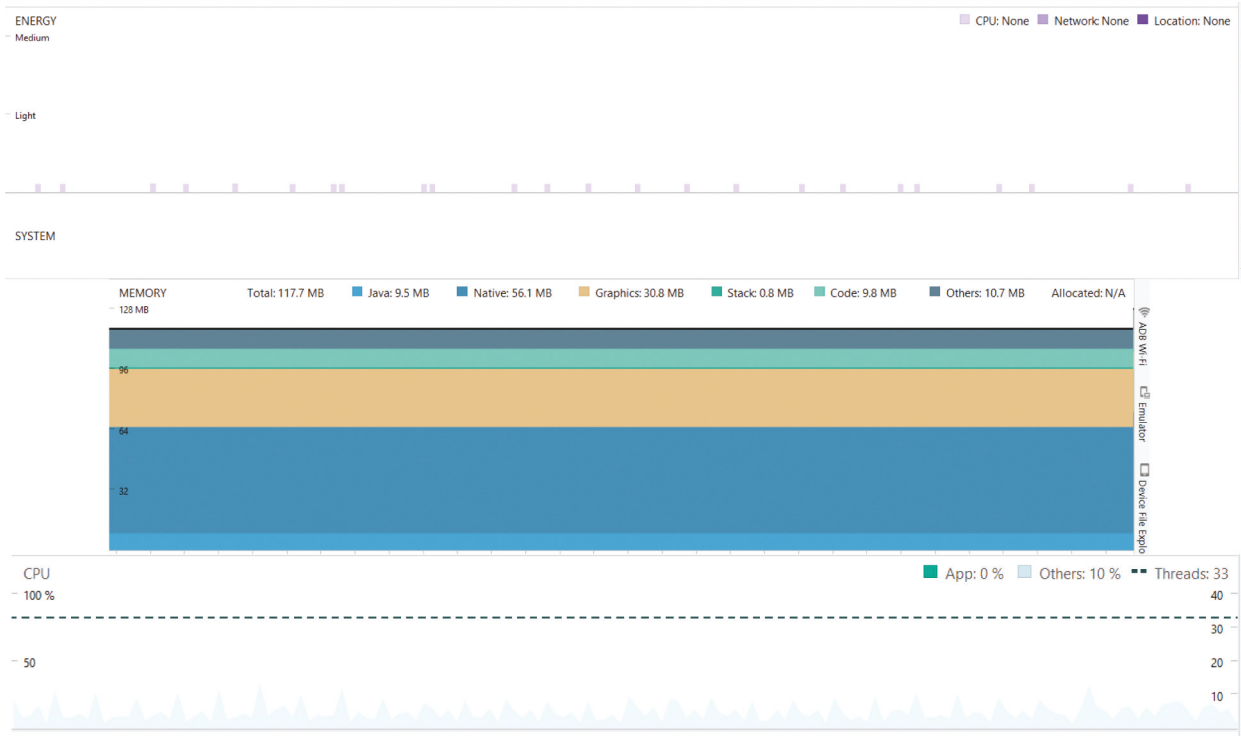

Figure 3: Results after applying optimization with three devices.

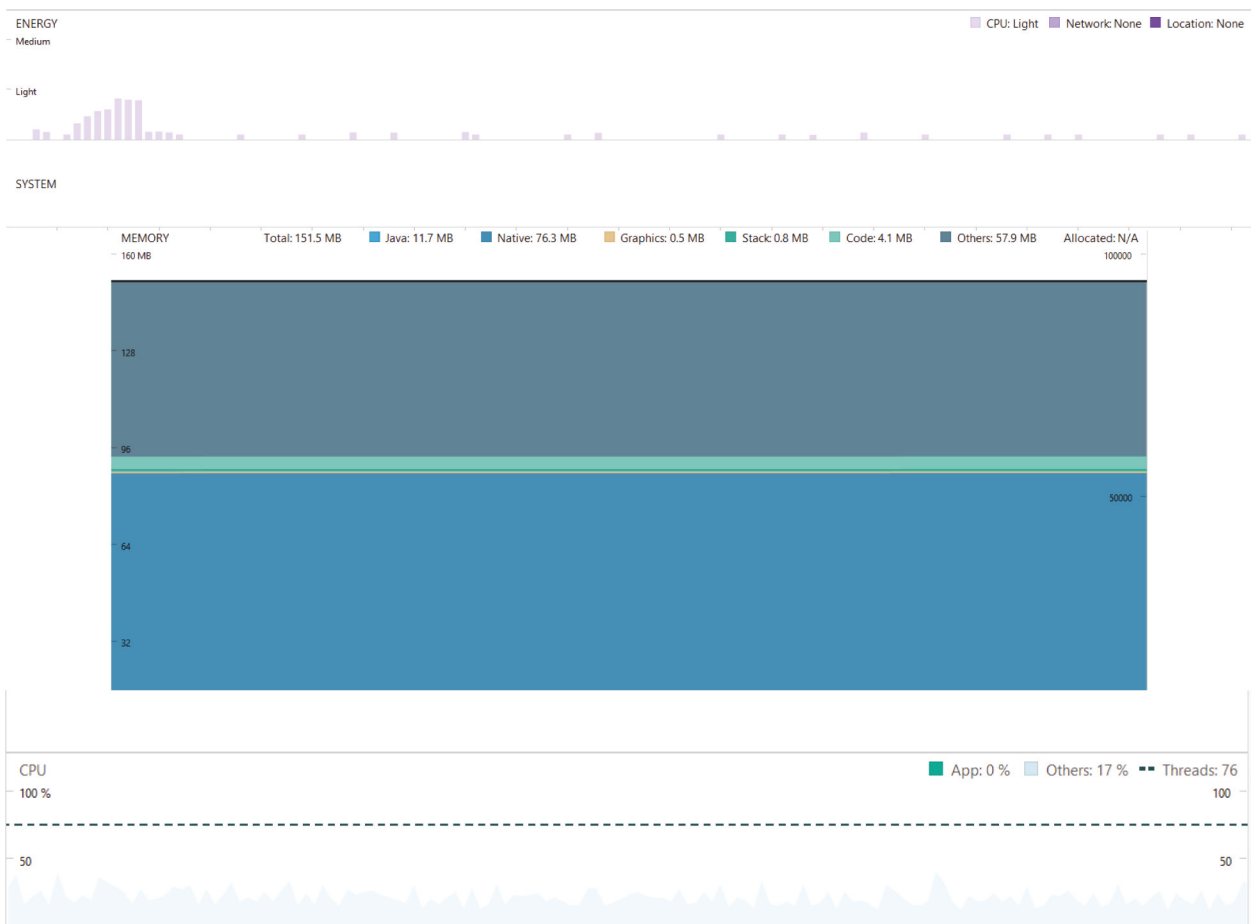

Figure 4: Results after applying optimization with one device.

where Rs,t is reserved resources of server at t-th time location, Rc,t is reserved resources of the client at t-th time location, and Uc,t is unused resources of the clients at t-th time location.

$$R_{s,t} = R_{c,t} + \mathrm{UW}_{c,t}, \tag{2}$$

where $R_{s,t}$ is reserved resources of server at t-th time location, Rc,t is reserved resources of the client at t-th time location, and $UWc,t.$ is unused-wholesaled resources of clients at t-th time location.

$$R'_{s,t} = R'_{c,t} + \mathrm{UB}'_{c,t}, \tag{3}$$

where $R_{s,t}$ is reserved resources of server at k-th time location, $R_{c,t}$ is reserved resources of the client at k-th time location, and $U\,B$ is unused-buyback resources of clients at the k time location

Equations (1)–(3) show the generalized working of the proposed solution by seeing results that total resource utilization occurred in this pattern, while the following equation is showing the total time that is consumed in executing a task:

$$\mathrm{Time} = \frac{\mathrm{CD} + Ts}{\mathrm{DL}_{(\mathrm{RAM\,CPU})}}, \tag{4}$$

where total consumption time for all tasks = $T\,T$, communication delay = CD, tasks list = Ts, and number of devices = DL.

6.2. Results without Optimization. It can be seen in Table 3 that without optimization the time consumption is 4 seconds for a task. It is due to the lack of a particle swarm algorithm. Also, the CPU percentage is high with relatively high memory in use. Table 3 represents the results without optimization.

6.3. Results of Proposed Solution (after Optimization). Previously, time consumption was 4 seconds for a task, and now after optimization, 1 second decreases because we have chosen the device, which is nearest by applying the particle swarm algorithm. Similarly, for three devices, the time was 2.5 previously, and now, it is 1.5 seconds. The memory usage also decreased. While CPU consumption previously in 1 device was 7%, after optimization, it is 2%, and for 3 devices, it is 3.5%, and after optimization, it is 2.5%. Table 4 presents the results of the proposed solution after optimization.

6.4. Simulation Results. In Figures 2–4, simulation results are clearly showing that the results without optimization are improved by applying optimization techniques of particle swarm. Moreover, adding a greater number of devices improved the results significantly.

7. Conclusion and Future Work

Hence, it is concluded that in this paper, selection of optimized resources and then their allocation in mobile edge computing decreased time, energy, and memory while executing tasks. These tasks if executed on a single device can increase these resources in a linear order. Complex and tedious tasks can easily be executed by making a mobile edge, and resources can be utilized in a better way. Edge can reduce consumption delay (CD) by adding N number of devices, which will improve the utilization of resources and will ensure the quality of service. Mobile edge shares resources to other edges by wholesale (sending resources) and buyback (receiving resources) scheme. In the future, wholesale and buyback resources from edge-to-edge servers will be used for profit maximization. Experimentation research techniques will be used to optimize resource allocation between two MECs, and algorithms will be designed for optimal memory, CPU, time, and power consumption between two MECs. Besides this, mobile edge computing has still faced a lot of challenges, and these are mobility management, heterogeneity, price, scalability, and security. We will also work on these mentioned sides in the future.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no potential conflicts of interest.

References

[1] Y. Xiao, Y. Jia, C. Liu, X. Cheng, J. Yu, and W. Lv, "Edge computing security: state of the art and challenges," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1608–1631, 2019.

[2] P. Mach and Z. Becvar, "Mobile edge computing: a survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials,* vol. 19, no. 3, pp. 1628–1656, 2017.

[3] A. Shakarami, M. Ghobaei-Arani, M. Masdari, and M. Hosseinzadeh, "A survey on the computation offload- ing approaches in mobile edge/cloud computing environment: a stochastic-based perspective," *Journal of Grid Computing*, vol. 18, no. 4, pp. 639–671, 2020.

[4] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: the communication perspective," *IEEE communications surveys & tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[5] S. Ali, M. Adeel, S. Johar, M. Zeeshan, S. Baseer, and A. Irshad, "Classification and prediction of software incidents using machine learning techniques," *Security and Communication Networks*, vol. 2021, Article ID 9609823, 16 pages, 2021.

[6] S. Wang, J. Xu, N. Zhang, and Y. Liu, "A survey on service migration in mobile edge computing," *IEEE Access*, vol. 6, pp. 23511–23528, 2018.

[7] B. Bill, F. Flavien, M. Richard, and J. Graves, "A methodology to evaluate rate-based intrusion prevention system against distributed denial-of-service (DDoS)," in *Proceedings of the Cyber Security, Cyber Crime and Cyber Forensics 2011*, UK, March 2011.

[8] K. Kumar, R. C. Joshi, and K. Singh, "A distributed approach using entropy to detect DDoS attacks in ISP domain," in *Proceedings of the 2007 International Conference on Signal*

Processing, Communications and Networking, pp. 331–337, IEEE, Chennai, India, February 2007.

[9] M. Abderrahmen, A. Harras Khaled, and F. Afnan, "Towards computational offloading in mobile device clouds," in *Proceedings of the 2013 IEEE 5th International Conference on Cloud Computing Technology and Science*, pp. 331–338, IEEE, Bristol, UK, December 2013.

[10] M. Masdari and M. Zangakani, "Green cloud computing using proactive virtual machine placement: challenges and issues," *Journal of Grid Computing*, vol. 18, no. 4, pp. 727–759, 2020.

[11] Y. Li, W. Li, and C. Jiang, "A survey of virtual machine system: current technology and future trends," in *Proceedings of the 2010 Third International Symposium on Electronic Commerce and Security*, pp. 332–336, IEEE, Nanchang, China, July 2010.

[12] R. Urgaonkar, S. Wang, T. He, M. Zafer, K. Chan, and K. K. Leung, "Dynamic service migration and workload scheduling in edge-clouds," *Performance Evaluation*, vol. 91, pp. 205–228, 2015.

[13] U. Ihtisham, R. Basit, S. Ali, A. Ahmed, B. Samad, and I. Azeem, "Software defined net- work enabled fog-to-things hybrid deep learning driven cyber threat detection system," *Security and Communication Networks*, vol. 2021, Article ID 6136670, 15 pages, 2021.

[14] Y. Zhang, X. Lan, J. Ren, and L. Cai, "Efficient computing resource sharing for mobile edge-cloud computing networks," *IEEE/ACM Transactions on Networking*, vol. 28, no. 3, pp. 1227–1240, 2020.

[15] Z. Lv and L. Qiao, "Optimization of collaborative resource allocation for mobile edge computing," *Computer Communications,* vol. 161, pp. 19–27, 2020.

[16] A. Rahman, G. Wu, and M. Liton Ali, "Mobile edge computing for internet of things (IOT): security and privacy issues," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 18, no. 3, pp. 1486–1493, 2020.

[17] X. Huang, R. Yu, J. Kang, and Y. Zhang, "Distributed reputation management for secure and efficient vehicular edge computing and networks," *IEEE Access*, vol. 5, pp. 25408–25420, 2017.

[18] R. Abubakar, A. Aldegheishem, M. Faran Majeed et al., "An effective mechanism to mitigate real-time DDoS attack," *IEEE Access*, vol. 8, pp. 126215–126227, 2020.

[19] W. Abdul, Z. Ali, S. Ghouzali, B. Alfawaz, G. Muhammad, and M. S. Hossain, "Biometric security through visual encryption for fog edge computing," *IEEE Access*, vol. 5, pp. 5531–5538, 2017.

[20] K. Kuljeet, S. Garg, K. Georges, G. Mohsen, and D. N. K. Jayakody, "A lightweight and privacy- preserving authentication protocol for mobile edge computing," in *Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, Waikoloa, HI, USA, December 2019.

[21] R. Gao, S. Li, Y. Gao, and R. Guo, "A lightweight cryptographic algorithm for the transmission of images from road environments in self-driving," *Cybersecurity*, vol. 4, no. 1, pp. 3–11, 2021.

[22] E. Ibrahim, M. Ammar, H. Mohammad, S. H. Ahmed, U. Devrim, and K. Mashael, "Security-aware data offloading and resource allocation for mec systems: a deep reinforcement learning," *TechRxiv*, 2021.

[23] Y. Niu, J. Zhang, A. Wang, and C. Chen, "An efficient collision power attack on AES encryption in edge computing," *IEEE Access*, vol. 7, pp. 18734–18748, 2019.

[24] Y. Ding, Y. Shi, A. Wang, X. Zheng, Z. Wang, and G. Zhang, "Adaptive chosen-plaintext collision attack on masked AES in edge computing," *IEEE Access*, vol. 7, pp. 63217–63229, 2019.

[25] P. Akomolafe Oladeji and O. Abodunrin Matthew, "A hybrid cryptographic model for data storage in mobile cloud computing," *International Journal of Computer Network and Information Security*, vol. 9, no. 6, p. 53, 2017.

[26] M. Houssemeddine, A. Nadjib, and B. Khaled, "Maximizing mobiles energy saving through tasks optimal offloading placement in two-tier cloud," in *Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 137–145, Montreal QC Canada, November 2018.

[27] T. Wang, J. Zhou, A. Liu, M. Z. A. Bhuiyan, G. Wang, and W. Jia, "Fog-based computing and storage offloading for data synchronization in IoT," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4272–4282, 2019.

[28] Z. Gao, W. Hao, Z. Han, and S. Yang, "Q-learning-based task offloading and resources optimization for a collaborative computing system," *IEEE Access*, vol. 8, pp. 149011–149024, 2020.

[29] C. Xu, W. Quan, A. V. Vasilakos, H. Zhang, and G. M. Muntean, "Information-centric cost- efficient optimization for multimedia content delivery in mobile vehicular networks," *Computer Communications*, vol. 99, pp. 93–106, 2017.

[30] A. Irshad, S. A. Chaudhry, O. A. Alomari, K. Yahya, and N. Kumar, "A novel pairing-free lightweight authentication protocol for mobile cloud computing framework," *IEEE Systems Journal*, vol. 15, no. 3, pp. 3664–3672, 2021.

[31] K. Peng, H. Huang, W. Pan, and J. Wang, "Joint optimisation for time consumption and energy consumption of multi-application and load balancing of cloudlets in mobile edge computing," *IET Cyber-Physical Systems: Theory & Applications*, vol. 5, no. 2, pp. 196–206, 2020.

[32] J. Sun, Q. Gu, T. Zheng, P. Dong, A. Valera, and Y. Qin, "Joint optimization of computation offloading and task scheduling in vehicular edge computing networks," *IEEE Access*, vol. 8, pp. 10466–10477, 2020.

[33] W. Y. B. Lim, N. C. Luong, D. T. Hoang et al., "Federated learning in mobile edge networks: a comprehensive survey," *IEEE Communications Surveys & Tutorials.*vol. 22, no. 3, pp. 2031–2063, 2020.

[34] S. Bi, L. Huang, and Y.-J. A. Zhang, "Joint optimization of service caching placement and computation offloading in mobile edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 19, no. 7, pp. 4947–4963, 2020.

[35] L. Chen, J. Wu, and J. Zhang, "Long-term optimization for MEC-enabled HetNets with device–edge–cloud collaboration," *Computer Communications*, vol. 166, pp. 66–80, 2021.

[36] K. Sha, T. A. Yang, W. Wei, and S. Davari, "A survey of edge computing-based designs for IoT security," *Digital Communications and Networks*, vol. 6, no. 2, pp. 195–202, 2020.

[37] M. Li, Y. Gao, M. Wang, C. Guo, and X. Tan, "Multi-objective optimization for multi-task allocation in mobile crowd sensing," *Procedia Computer Science*, vol. 155, pp. 360–368, 2019.

[38] T. Rausch, A. Rashed, and S. Dustdar, "Optimized container scheduling for data-intensive serverless edge computing," *Future Generation Computer Systems*, vol. 114, pp. 259–271, 2021.

[39] M. L. Montoya Freire, D. Potts, N. R. Dayama, A. Oulasvirta, and M. Di Francesco, "Foraging-based optimization of

pervasive displays," *Pervasive and Mobile Computing*, vol. 55, pp. 45–58, 2019.

[40] P. Nawrocki and W. Reszelewski, "Resource usage optimization in mobile cloud computing," *Computer Communications*, vol. 99, pp. 1–12, 2017.

[41] I. A. Elgendy, W. Zhang, Y.-C. Tian, and K. Li, "Resource allocation and computation offloading with data security for mobile edge computing," *Future Generation Computer Systems*, vol. 100, pp. 531–541, 2019.

[42] A. A. Muhamad, "Advanced encryption standard (AES) algorithm to encrypt and decrypt data," *Cryptogra- phy and Network Security*, vol. 16, pp. 1–11, 2017.

[43] K. Taha Zahraa, "Text encryption using modified AES-2 keys," *International Journal of Computer Application*, vol. 149, no. 4, pp. 27–31, 2016.

[44] V. Akhilesh, P. Ramya, and J. (Selena) He, *Security in Fog Computing through Encryption*, DigitalCommons@ Kennesaw State University Kennesaw, Georgia, USA, 2016.

[45] B. Karthikeyan, T. Sasikala, and P. S. Baghavathi, "Key exchange techniques based on secured energy efficiency in mobile cloud computing," *Applied Mathematics & Information Sciences*, vol. 13, no. 6, pp. 1039–1045, 2019.

[46] I. A. Elgendy, A. Muthanna, M. Hammoudeh, H. Shaiba, D. Unal, and M. Khayyat, "Advanced deep learning for resource allocation and security aware data offloading in industrial mobile edge computing," *Big Data*, vol. 9, no. 4, pp. 265–278, 2021.

[47] S. Liu, X. Yu, S. Ji, and Y. Peng, "A fast hybrid data encryption for FPGA based edge computing," in *Proceedings of the 2019 14th IEEE International Conference on Electronic Measurement & Instruments (ICEMI)*, pp. 1820–1827, IEEE, Changsha, China, November 2019.

[48] C.-H. Chu, "Task offloading based on deep learning for blockchain in mobile edge computing," *Wireless Networks*, vol. 27, no. 1, pp. 117–127, 2021.

[49] P. Velmurugadass, S. Dhanasekaran, S. Shasi Anand, and V. Vasudevan, "Enhancing Blockchain security in cloud computing with IoT environment using ECIES and cryptography hash algorithm," *Materials Today Proceedings*, vol. 37, pp. 2653–2659, 2021.

[50] B. Hamza, A. Ahmad, and K. Attila, "P. F.-BTS: A privacy-aware fog-enhanced blockchain-assisted task scheduling," *Information Processing & Management*, vol. 58, no. 1, Article ID 102393, 2021.

[51] S. Ali, S. Baseer, I. A. Abbasi, B. Alouffi, W. Alosaimi, and J. Huang, "Analyzing the interactions among factors affecting cloud adoption for software testing: a two-stage ISM-ANN approach," *Soft Computing*, vol. 26, no. 16, pp. 8047–8075, 2022.

[52] M. F. Majeed, M. N. Dailey, R. Khan, and A. Tunpan, "Precaching: a proactive scheme for caching video traffic in named data mesh networks," *Journal of Network and Computer Applications*, vol. 87, pp. 116–130, 2017.

[53] M. F. Majeed, S. H. Ahmed, and M. N. Dailey, "Enabling push-based critical data forwarding in vehicular named data networks," *IEEE Communications Letters*, vol. 21, no. 4, pp. 873–876, 2017.

[54] M. Faran Majeed, S. Hassan Ahmed, S. Muhammad, H. Song, and D. B. Rawat, "Multimedia streaming in information-centric networking: a survey and future perspectives," *Computer Networks*, vol. 125, pp. 103–121, 2017.

[55] M. Muhammad Faran, A. S. Hassan, M. Siraj, and N. Dailey Matthew, "Poster PDF: push-based data forwarding in vehicular NDN," in *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services Companion*, p. 54, Singapore Singapore, June 2016.

[56] M. Xiao, W. Lin, Y. Dai, and Y. Zeng, "A fast algorithm to compute maximum k-plexes in social network analysis," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, CA USA, February 2017.

[57] Z. Qu, S. Chen, and X. Wang, "A secure controlled quantum image steganography algorithm," *Quantum Information Processing*, vol. 19, no. 10, pp. 1–25, 2020.

[58] M. Wu, L. Tan, and N. Xiong, "A structure fidelity approach for big data collection in wireless sensor networks," *Sensors*, vol. 15, no. 1, pp. 248–273, 2014.

[59] S. Huang, A. Liu, S. Zhang, T. Wang, and N. N. Xiong, "BD-VTE: a novel baseline data based verifiable trust evaluation scheme for smart network systems," *IEEE transactions on network science and engineering*, vol. 8, no. 3, pp. 2087–2105, 2021.

[60] H. Li, J. Liu, K. Wu, Z. Yang, R. W. Liu, and N. Xiong, "Spatio-temporal vessel trajectory clustering based on data mapping and density," *IEEE Access*, vol. 6, pp. 58939–58954, 2018.

[61] K. Gao, F. Han, P. Dong, N. Xiong, and R. Du, "Connected vehicle as a mobile sensor for real time queue length at signalized intersections," *Sensors*, vol. 19, no. 9, p. 2059, 2019.

[62] Y. Pan, N. Xiong, and J. Ren, "Data security and privacy protection for cloud storage: a survey," *IEEE Access*, vol. 8, pp. 131723–131740, 2020.

[63] Z. Yang, A. I. Ahmed, A. Fahad, S. Ali, and M. Zhang, "An IoT time series data security model for adversarial attack based on thermometer encoding," *Security and Communication Networks*, vol. 2021, Article ID 5537041, 11 pages, 2021.

[64] Z. Qu, H. Sun, and M. Zheng, "An efficient quantum image steganography protocol based on improved EMD algorithm," *Quantum Information Processing*, vol. 20, no. 2, pp. 53–29, 2021.

[65] W. Tan, P. Huang, X. Li, G. Ren, Y. Chen, and J. Yang, "A review on segmentation of lung parenchyma based on deep learning methods," *Journal of X-Ray Science and Technology*, vol. 29, no. Preprint, pp. 1–15, 2021.

[66] W. Tan, L. Zhou, X. Li, X. Yang, Y. Chen, and J. Yang, "Automated vessel segmentation in lung CT and CTA images via deep neural networks," *Journal of X-Ray Science and Technology*, vol. 29, no. 6, pp. 1123–1137, 2021.