Hindawi
Mobile Information Systems
Volume 2022, Article ID 4840801, 12 pages
https://doi.org/10.1155/2022/4840801

Research Article

Collaborative Computation for Offloading and Caching Strategy Using Intelligent Edge Computing

Yi-fan Zheng, Ning Wei, and Yi Liu

Department of Ordnance and Control, Army Armored Force Academy, Beijing 10000, China

Correspondence should be addressed to Ning Wei; p00386@buaa.edu.cn

Received 15 April 2022; Revised 20 May 2022; Accepted 24 May 2022; Published 29 July 2022

Academic Editor: M. Praveen Kumar Reddy

Computation offloading and caching strategy is a well-established concept for allowing mobile applications that are high in resources. Furthermore, the unloaded duties can be replicated when several customers are within easy access because of the rising mobile cooperation applications. However, the problematic characteristics of offloading and caching strategy delay bandwidth transfer from mobile computing devices to cloud computing. A new technical approach to restrict the issues and unwanted functions in offloading and caching is called the intellectual power computing framework (IPCF). IPCF depends on two conventional offloading and caching strategies called systematic offloading technique and managerial migrant caching. The migration of data transfer from the destination to location basis lies in the systematic offloading technique to restrict network delays. Managerial migrant caching duplicates the data required by the mobile terminals (MTs) from the remote cloud storage to the mobile application to reduce the access time. The forbidden actions in current techniques are refused, and solutions are enhanced for better communication strategy. Thus, the simulation analysis performs better in IPCF to reach efficient outcomes for offloading and caching processes.

1. Intelligent Edge Processing Technique and Network Components

Intelligent edge processing is a concept that describes the process in which information is evaluated and collected in a location near the networking [1]. The intelligent edge process is called “intellect at the side,” which has significant consequences for telecommunications environments and the internet of things [2].

With an intelligent edge, distributed or decentralized system components can perform many types of data processing, which can be managed at a center located in a system [3]. Notably, IoT offers several notable drawbacks for a conventional paradigm for routing numerous data channels from IoT-connected equipment into a central database [4]. It is potentially wasteful and can make the system extra insecure if the information is not secured.

The edge network resources or stations can automatically handle, bundle, refine, or process authentication for transit to the database with an intelligent edge arrangement [5]. It can enhance the efficiency and security of data processing equipment. For these considerations, many cloud operators and other firms who are aware of the structure and function of IoT advocate using the intelligent edge [6].

Figure 1 shows the intelligent edge processing that contributes the sensors and other devices to the merging of the cloud system. The intelligent cloud works with Wi-Fi, modem, and LAN to merge with the monitoring system that provides contact to sensors, wind energy, supporting poles, and many transmission systems [7]. The wireless modules of the Wi-Fi modem and LAN relates with each other and communicates to detecting devices.

Storage for IoT and Big Data applications is another pricey resource. Not all raw data generated for analytical purposes are necessary; certain information can be processed and handled on the system level [8]. It enables relevant information to be created before being sent for analysis to the cloud services unit. Therefore, the cost of maintaining the database can be reduced, and the cloud services burden can be lowered. Furthermore, with edge computing, security

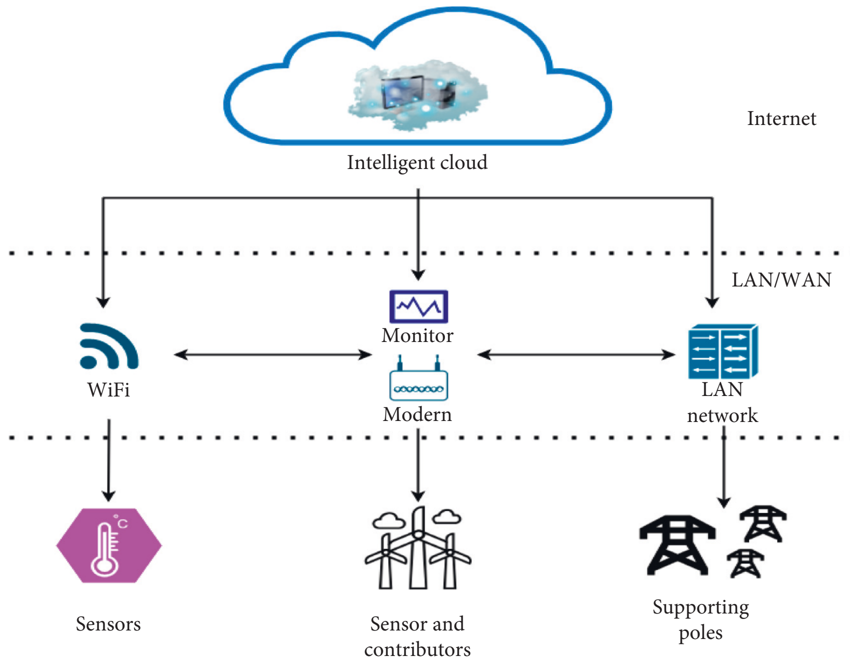

Figure 1: Intelligent edge processing.

and privacy can be enhanced by handling personal data on the device [9]. Edge computing can reduce the number of internet-enabled devices and lower the program's possible cyber-attacks. Indeed the quantity of sensitive and private information supplied across the network can be reduced by local data acquisition on the premises of the gadgets [10].

1.1. Computation of Offloading and Caching Strategy. Today's phone carriers are increasingly trustworthy for collaborating across mobile and remote cloud apps [11]. Its application and the cloud have now proven to implement cloud computing systems for a phone network, where the app acts as a compact service provider with essential capabilities, limited computing capabilities, and data backup. In contrast, the cloud provides complimentary services and has a reasonably powerful computing and storage capacity [12].

Calculation and data transforming are always closely linked in certain circumstances [13]. For example, there are two subprocesses in a total potential cloud interconnection between an app and its cloud: (1) the application accesses the appropriate cloud data, and then (2) the application runs the needed mobile device local computation. In portable contexts, the mobile station (MS), a base station (BS), and cloud-connected phone carriers are involved in this process [14]. A delay will be formed in both approaches, rendering to competencies of the connection and offloading capabilities of MS [15].

Computation downloading relocates apps from resource-restricted MS's to exterior comparative computing resources to minimize the time needed to do computational tasks [16]. Information caching doubles the process of reducing data access latency for applications from the distant cloud to a cache in the proximity of the MS [17]. It efficiently improves the ability of MT to handle a variety of mobile apps for computing and data accessibility [18].

Current work in this field focuses on intellectual edge processing (IEP) and data exchanged by extra support from the primary station to the cloud services [19]. The design of MSs' computer downloading structure on intelligent cloud computing is permitted, or cache policies with data cache capabilities are proposed [20]. Such techniques are constrained, however, to a conceptualization that separately differentiates computational offloading and data caching. When computer downloading and information cache management techniques are used together, the calculation can be downloaded to cloud edge computing and then used later [7]. Where the data are required, it can be cached at cloud storage and then accessed by BS, thereby reducing delays between the app and the cloud throughout the interconnection [21].

Hence the emerged system of IPCF has more strategic maintainability and resource-based communication for more reliability [22]. The main contributions of the highlighted technique are As follows:

(i) Intellectual power computing framework offers on working offloading and caching sectors for communication

(ii) Systematic offloading technique ensures no delay or less traffic during network transaction

(iii) Managerial migrant caching manages the duplicate data on caching process for reducing accessing time

The description of Section 1 gives more details on intelligent cloud processing, computation offloading, and caching techniques. Section 2 provides the research survey on computational strategies on intelligent edge computing. Section 3 offers the recovering methods and problems of existing forms of offloading and caching strategies. IPCF is a managing technique whose tabulations and graphs are described in Section 4. Finally, the conclusions of the proposed IPCF method are established in Section 5 with future perspectives.

2. Survey on Offloading and Caching Strategy on Intelligent Edge Computing

This section describes several research works based on the offloading and caching strategy on intelligent edge processing. The aim is to find a way for these research themes to escape the discomfort of problematic characteristics by enhancing decision-making. Many of these terms have activated function to influence or adapt it, while others try to do a more comfortable match for influences or better management of excellence.

Tefera et al. [23] introduced adaptive resource calculation offloading and caching for edge computing networks (RCOC-ECN). Decentralized multiaccess edge computing (MEC) download and cache is an excellent way to evolve the next generation of the network. The global ecosystem connects enormous internet of everything (IoE) devices continuously. It increases traffic in the backbone network tremendously and the highly dependable low bandwidth connection for consumers. This work examined a decentralized adaptive resource-conscious telecommunication, compute, and cache architecture that may structure deep enhancement knowledge-driven dynamic network environments (DNEs). They implemented a decentralized cognitive programming algorithm using deep reinforcement learning technology to utilize IoE and intelligent electronic devices. Li et al. [24] suggested cooperative cache distribution and task scheduling applications in the edge computing environment (CT-ECE). A new paradigm, the edge computing system, could bring services closer to consumers, significantly reducing latency and improving the lives of the UE battery. The study proposed a way of cache-conscious job scheduling in cloud computing. First, the data chunk broadcast, the caching worth, and the cache management penalty were calculated from an integrated utility function. Then, for optimizing the built-in utility value, data chunks were cached on optimal edge servers. A cache-based scheduling algorithm mechanism was described after placing the caches. Moreover, experiments revealed that caching conscious task scheduler exceeds the cache hit rate, data locality, reaction time, and power costs of other existing techniques. Liu et al. [25] identified cooperative task calculation offloading in cloud-assisted edge figuring (TCO-CEF). They introduced a three-stage networking cloud-assisted computing paradigm in an IoT setting. First, a mixed-integer programming problem with the energy consumption problem was introduced, considering two limits, the task-dependence requirement and the IoT services' accomplishment term. Second, they presented a unified task-computing energy-efficient solution for the offloading IoT sensors based on a semifinite reduction and probabilistic mapper method. Wang et al. [26] presented an intelligent mobile edge computing for resource caching (IMEC-RC). This study contained cognitive agents (CAs) to assist users in caching and executing activities with MEC in preparation. CA was often used in detail to develop a custom model in conjunction with user information. The results of the trials suggested that CA could increase communications network performance, decrease network pain, and enhancing the quality of service for users.

Ren et al. [27] suggested a survey on end-edge-cloud-scored network computing paradigms (ECNCP). In the internet-of-things age, the enormous rise in the number of sensors and data roadworks imposed substantial infrastructure web and unmanageable performance delay. Adopting cloud computing becomes challenging for IoT applications' late sensitivity and situational services provided. Much novel general computing was arisen to use scattered capabilities at the edge of the network to provide timely and contextual applications, including transparent computing, edge computing computers, fog computers, and couches. They gave a thorough review from the standpoint of end-cloud management of these new computing models.

Zhao et al. [28] introduced deep learning-based mobile data offloading edge computing systems (DP-MO-ECSs). Owing to this ability to gratify a broad range of demands of the current wireless endpoints, such as data rate, reduced power, and massive data processing, Mobile edge computing (MEC) was promising. Owing to Wi-Fi AP's practical and straightforward features, various devices from tiny BSs could download the document. In addition, a multilong deep study model based on short- to medium-term storage was designed to forecast SBS's required information, which could help us complete the download process with precision. The findings shown based on a real data set demonstrate the usefulness of the forecasting and discharge method that they have put forward.

Tefera et al. [29] suggested decentralized adaptive computational offloading and caching for multiedge computing (DACO-CMEC). Innovative technologies or connected devices have been created to perform demanding tasks requiring additional computational and other central computer resources. They suggested a resource-aware decentralized multiaccess cloud technology and storage architecture to overcome the difficulty in the process [30, 31]. The downside was founded on the NP-hard concept of noncooperative gaming, and they demonstrated that the game could give a balance of optimal. They made a thorough assessment that the system delivers higher performance in terms of increased storage capacity, high user experience, and reduced energy consumption than the standard method.

Thus, the research survey on the computation of offloading and caching based on intelligent edge computing is illustrated on existing RCOC-ECN, CT-ECE, TCO-CEF, IMEC-RC, ECNCP, DP-MO-ECS, and DACO-CMEC. Furthermore, the problematic features and their related actions are analyzed well on current methods to construct a proposed plan of greater frequency.

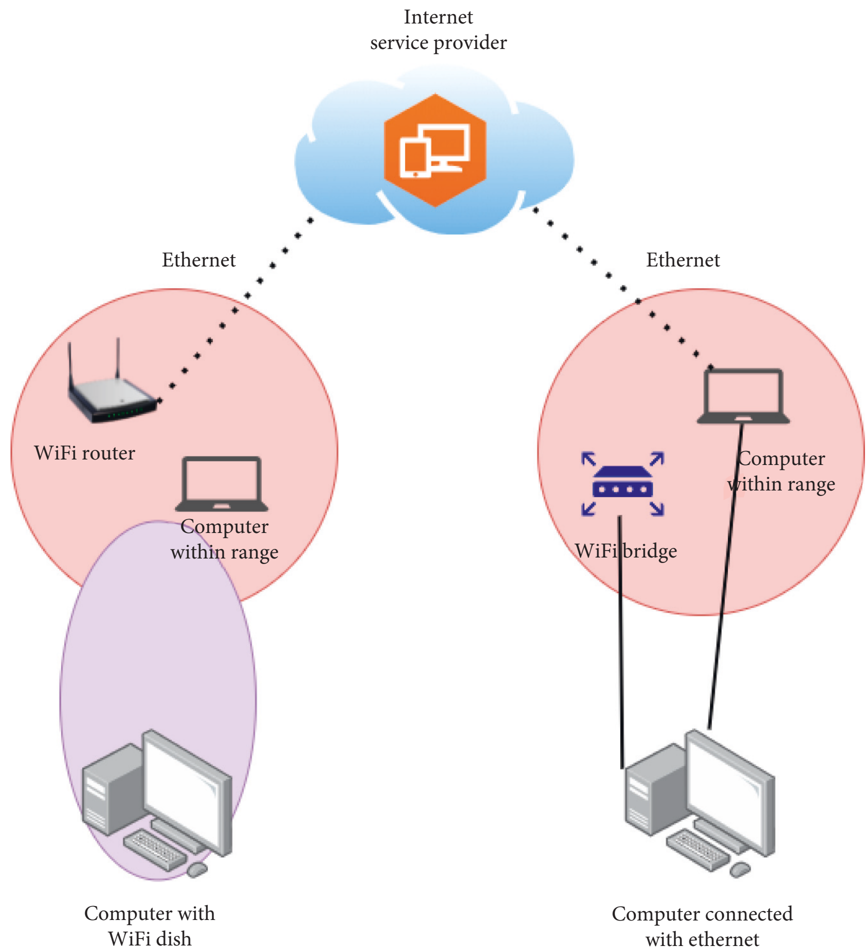

Figure 2: Data offloading technique.

3. Intelligent Edge Processing on Computation Offloading and Caching Strategies

Today, modern wireless communications are rapidly developing; devices such as smartphones and wearable gadgets have several varied capacities and computational wireless internet connectivity characteristics. Mobile phones will have more sophisticated in the long term, and their applications require substantial computer power and continuous access to data. However, if portable systems are added through a wireless connection to the cloud, a reasonably long latency does not fit delayed requirements.

Intelligent edge processing gives users short-term and powerful computational services using computer terminals and servers on the edge of a network that satisfies delay-sensitive task needs of users in the IPCF network. The edge cloud has two key advantages:

(i) Associated with local computing, mobile unit computing can overcome the restricted calculating control of smartphones

(ii) In contrast to the remote cloud, although the edge cloud has a little topographical space and restricted supply capacity, it can evade substantial delay caused by offloading the data content to the distant server

Therefore, intelligent edge processing over mobile units exhibits better value on delay content and traffic strategy of data transferring tasks using IPCF. Based on the current works of offloading and caching technique over delay process, the following aspects:

Data offloading: this is often referred to as data caching. The service provider can store preferred material on a cloud system to decrease user demand information delays and energy usage on IPCF.

Job offloading: the primary design problems are where, why, and how well the mobile phone may discharge client chores onto the internet edge, minimize computer delays, and save electricity.

3.1. Data Offloading

3.1. Data Offloading. Figure 2 shows the data offloading technique on the internet service provider. Systems are connected via Ethernet, one with the Wi-Fi router and another with a computer within the range. Space consists of a laptop with a Wi-Fi router and a computer with a Wi-Fi bridge on another side. One side of the system is managed to Wi-Fi's connected through a wireless section using IPCF. Another side system is merged with Wi-Fi bridge and computer with longer range with wired communication.

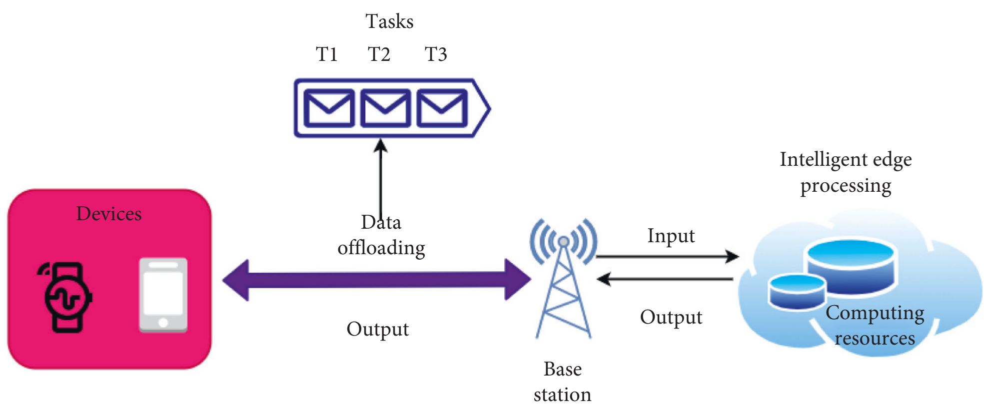

Figure 3: Computation offloading technique.

$$d_m = W \log_3\left(1 + \frac{S_m + A_m}{\varepsilon^3}\right). \quad (1)$$

Equation (1) gives the communication transfer from computer to cloud computing where d_m is the uplink information ratio, m is the number of users to communicate, and W is the channel wavelength of $\log_3$ for evaluating values. S_m is the power transmission through the mobile unit, A_m is the constant data over network, and ε^3 says the traffic over the network, which is cube rate to matter the data on IPCF.

The basic delay problem over the data offloading on edge computing is as follows:

$$E^q_{m,c} = \frac{\gamma_c}{g^q_m}. \quad (2)$$

Equation (2) gives the data offloading technique on the cloud edge state, where m is the traveling time over the section task, c is the CPU frequency and is constrained by maximum bandwidth of q, g is the energy efficiency parameter denotes the exponent parameter as g^q_m. The data offloading mechanism is written as $E^q_{m,c}$, and γ_c is the task data over traveling in IPCF.

$$E^q_{m,c} = c\left(g^q_m\right)^2 \gamma_c. \quad (3)$$

Equation (3) gives the energy corresponding data task of $c\left(g^q_m\right)^2$ to the γ_c task data over traveling. Data offloading concerns intelligent edge processing, not considering at the same moment edge cloud storage and computing capacities. However, offloading this assumption is not realistic because peripheral cloud services power is restricted, and all computational jobs cannot be supported.

3.2. Computation Offloading. Figure 3 illustrates the computation offloading technique that gives the mobile device smartwatch other devices connected to the intelligent edge sources. The data updation is based on the task which is taking place in the task sector. When a task is brought into the connection through offloading technique, the section is added to the task 1 section called T1. After getting the job into the task folder, the next working mode section is introduced to the residing task2 area over IPCF. Finally, the data move to BS and enters the computing sources of intelligent edge processing on filling the task sector.

The edge processing acts as a verifying phase to know about the query from the tasks and provides the related output to the BS. Finally, the supporting pole gives the result to the mobile and smartwatch through the networking facility on IPCF.

$$\begin{aligned} E^q_{m,c} &= E^{\text{task}}_{m,c} + E^{\text{data}}_{m,c} \\ &= \left(\frac{x_c}{d_m}\right) + \left(\frac{\gamma_c}{g^q_m}\right). \end{aligned} \quad (4)$$

Equation (4) illustrates the task and data uploading in the computation offloading technique of given programs of the IPCF network. $E^{\text{task}}_{m,c}$ is the program transmitting through the task section, and $E^{\text{data}}_{m,c}$ is the data traverse to the task sector for engaging values. x_c is the program data size of the task, and x_c/d_m is the local computation overtaking, and γ_c/g^q_m is the data from the user for communication.

$$\begin{aligned} E^q_{m,c} &= S_m E_{m,c} \\ &= S_m \frac{x_c}{d_m}. \end{aligned} \quad (5)$$

Equation (5) illustrates the task-based uploading on the data offloading capturing in edge computing x_c/d_m is the local computation overtaking, where $E_{m.c}$ is the computation delay cost of the IPCF network. For calculating a task, the individual user information and the relevant program processing are required in IPCF. The cache was recently identified as a cost-efficiency strategy for reducing computational delays, power consumption, and connectivity expenses to store the program and task data constantly in the IPCF network.

3.3. Intellectual Power Computing Framework over the Computational State of Offloading and Caching. Figure 4 intellectual power computing framework based on the IoT devices, which consists of laptops, smartwatches, and smartphones. Intelligent cloud processing is the cloud service technique to store and retrieve values to mobile devices.

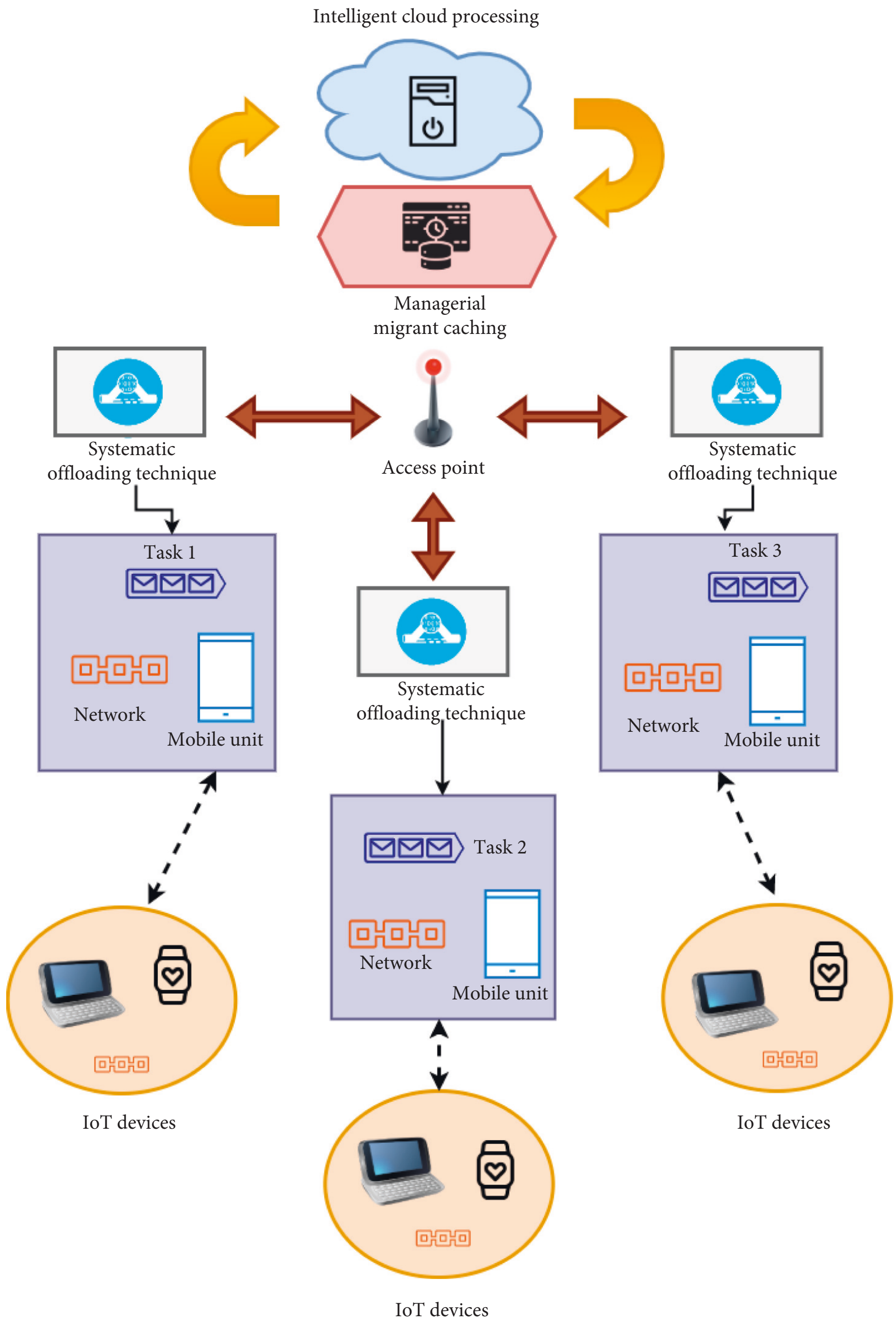

Figure 4: Intellectual power computing framework.

3.3.1. Access Point. Access Point is a communication device that operates on the IPCF network. An Access point for networking, data packets between end systems is received and transmitted. To verify the IP and MAC addresses of the target, review the header part of the received packet over IPCF. Computers can be accessed by Wi-Fi, Fast Ethernet, and Optical Fibre connection via the access point. The message includes its operational systems to track the address of the devices connected to the different access point connections.

3.3.2. Systematic Offloading Technique. The offloading of tasks is the primary form of IPCF intertask, and it can be used among activities, disruptions, duties to transmit messages. The thread-secure FIFO buffers are mainly employed, with new data being sent back to the loading mechanism, even though information can be forwarded from the front.

Figure 5 gives the task loading technique to the task queue of 1 and 2 sections. The task 1 batch consists of data to be uploaded into cloud computing for processing. The values are entered into the task queue are loading into it, and the

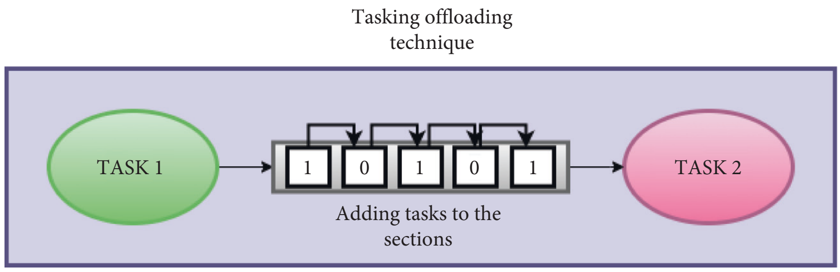

Figure 5: Task offloading technique.

values are managed as digital variables. These digital values are getting into the modified selection of data using the FIFO technique. After uploading data into the task queue, these values are taken to task 2 manager for other performances to cloud computing.

$$E^{q}_{m,c} = x_c \frac{\gamma_c}{g^{q}_{m}} + (1 - x_c)\left[\beta_m E^{q}_{m,c} + (1 - \beta_m) E_{m.c}\right]. \quad (6)$$

Equation (6) shows the energy consumption of the mobile device, and β_m is resource allocation over the offloading technique. $(1 - \beta_m)$ is the minimum resource allocation for task queue segment, $(1 - x_c)$ is the binding area, and $E_{m.c}$ is the computation delay cost. $\beta_m E^{q}_{m,c}$ is the transmit power over the transmission lines for cloud processing.

$$G_{m.c} = (1 - x_c)\left[\beta_m G^{q}_{m,c} + (1 - \beta_m) G_{m.c}\right]. \quad (7)$$

Equation (7) gives the minimization of power price called $G_{m.c}$, and $G^{q}_{m,c}$ is the caching placement decision for energy offloading. Thus, the threshold state through the strategic sector of loading data with less power consumption is given.

3.3.3. Threshold State. It is the minimum value to be outsourced by an implementation on IPCF. In processing time, energy usage, and memory utilization, the minimum value can be evaluated. When a specific application needs some time to be discharged to calculate on a mobile phone, it generates a high degree of customer electricity and battery capacity over IPCF. Demands on the large quantities of storage batteries and exceeds the lowest possible threshold value are efficient at discharging the implementation into the cloud.

3.3.4. Storage of an Application. The option to download depends on the magnitude of the request to be discharged over the IPCF network. If the size of the code is large, it saves energy to compile the code. Unless the compression ratio is modest, the download uses more energy and batteries than global applications on the IPCF network. The download costs for tiny matrices are higher regarding energy effectiveness, whereas costs of matrix multiplication might be up to 37%.

3.3.5. Significant Section. A significant constraint is the portion of the code that requires sophisticated calculations and extra time to perform over IPCF. The software component will be downloaded if it falls within the critical area. As a result, comprehensive application processing is discharged into the internet, saving many information systems power, memory, and CPU use.

$$g(\beta) = \sum_{a=1}^{a} \left(1 - x^{1}_{c}\right)\left[\beta_m G^{q}_{m,c} + (1 - \beta_m) G_{m.c}\right]. \quad (8)$$

Equation (8) gives the task caching over the offloading factor $g(\beta)$ of the IPCF network, which works best for the data allocation on the systematic offloading technique. $(1 - x^{1}_{c})$ is the total battery capacity where a is the transmission rate over the queue capacity starting from 1, with the summation values.

$$g(w) = \sum_{a=1}^{a} \left(1 - x^{1}_{c}\right)\left[\beta^{*}_{m} G^{q}_{m,c} + (1 - \beta^{*}_{m}) G_{m.c}\right]. \quad (9)$$

Equation (9) gives the power efficiency of w, where $g(w)$ is the maximum power efficiency over the IPCF network. β^{*}_{m} is the total power to the intelligent edge processing system for further energy transmission with a minor delay in energy efficiency. $(1 - \beta^{*}_{m})$ is the complete processing over the network with a traffic-less area.

3.3.6. Managerial Migrant Caching. Figure 6 shows the migrant caching formation of data adding to the buffer area. Cache displacement has a memory unit for entering the data into the cache system to the intelligent edge processing. A1, A2, and A3 are the three different memory areas of the caching section to add the information from the mobile device to the storage area with less delay and less traffic intervention. The $l1$, $l2$, to l_{n+1} are the wireless communication stages where the interaction of devices is started. Caching process of the first two areas is the offloading section to insert data from the particular application to the caching team. It happens from and to the cloud section with some delayed communication to the cache buffer. The wireless communication has an interface unit of loading data from the MT to the intelligent edge processing. The MT has several mobile users from 1 to n to communicate with the application and cloud processing.

A novel theory of managerial migrant caching has been presented by considering better feasible and energy-efficient intelligent edge computing for the mobile unit. Migrant caching management refers to the processing task program

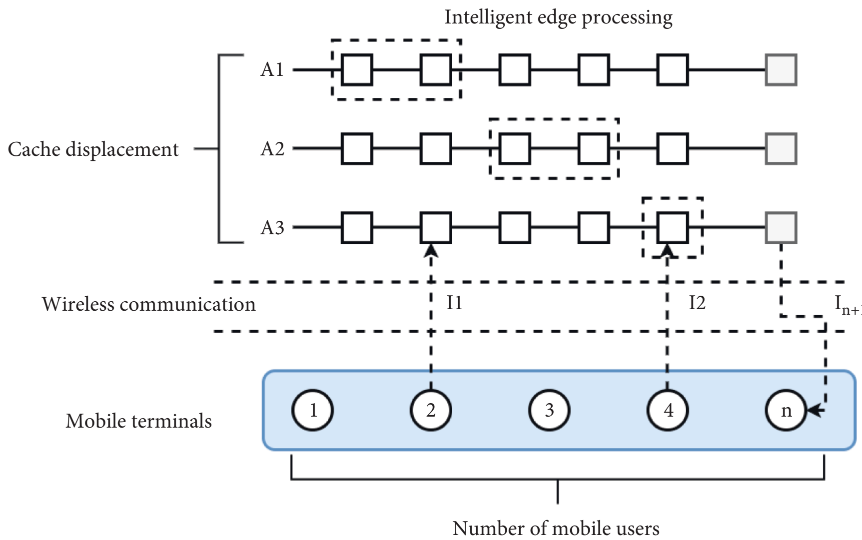

Figure 6: Managerial migrant caching.

and its associated data. Thus, caching considers both calculation and intelligent edge processing storage restriction since it needs memory and computation.

$$E^{\text{data}} = \sum_{a=1}^{n+1} [(1-\beta_m)\rho_m + \beta_m \rho_m^c]. \tag{10}$$

Equation (10) gives the processing unit of managerial migrant caching technique E^{data}, where the summation value of $a = 1$ gives the value from a single mobile unit to the number of mobile users called $n+1$. ρ_m^c is the wireless communication interface for data communication and ρ_m is the data transfer from the cloud to the mobile unit using the caching capacity.

$$E^{\text{task}} = \sum_{a=1}^{n+1} \left[(1-\beta_{m-1})\beta_m \rho_m + (1-\beta_m)\rho_m^1 + \beta_m \rho_m^1\right]. \tag{11}$$

Equation (11) gives the E^{task} input data size over the task of wireless media for information transfer. β_{m-1} is the required cycles for computational task based on the CPU capacity on the IPCF technique, and ρ_m^1 is the offloading decision vector on the network. $(1-\beta_m)\rho_m^1$ is the caching decision profile to take part in resource allocation and $\beta_m \rho_m^1$ is the computing delay factor.

$$\begin{aligned} E &= E^{\text{data}} + E^{\text{task}} \\ &= \sum_{a=1}^{n+1} \left[(1-\beta_{m-1})\beta_m \rho_m + (1-\beta_m)\rho_m^1 + \beta_m \rho_m^1 \right. \\ &\quad \left. + (1-\beta_m)\rho_m + \beta_m \rho_m^c\right]. \end{aligned} \tag{12}$$

Equation (12) gives the integration of systematic offloading technique and managerial migrant caching called E. Here, E^{data} is the offloading technique to the E^{task} caching section of migrant data. Thus, the adding of the equations (10) and (11) gives the found methods of the IPCF mechanism.

Figure 7 gives the parameter relationship between equations (10) and (11), which is integrated into equation (12). The added solution of IPCF's networks has two systematic offloading techniques and managerial migrant caching to form a solution. Thus, those data are included to have a complete equation of the IPCF method in solution (12) for more performance.

$$E = \sum_{a=1}^{n+1} [(1-\beta_m) r_s + (1-\beta_{m-1})\beta_m r_s + \beta_m r_s^*]. \tag{13}$$

Equation (13) gives the energy source data on the IPCF mechanism to provide less delay with less energy production. r_s is the total computation delay function and r_s^* is the error rate analysis on an overall network of IPCF.

$$E_{s,a} = \rho \frac{Y_a}{D_i} * J. \tag{14}$$

Equation (14) illustrates the complete unit of cloud server computing as $E_{s,a}$, ρ is the field transfer, and Y_a is the comparison ratio over the existing methods to the proposed network of the IPCF mechanism. D_i is the data loading and computation analysis to multiply total entity in J as offloading rate ratio.

$$E = \frac{Y_a}{D_i} + \frac{Y_1}{r} + \frac{\gamma_c}{g_m^q}. \tag{15}$$

Equation (15) shows the energetic expression of Y_a/D_i to monotonically increasing function of Y_1/r where r is the rate. To increase the quality of data transfer over the wireless terminal and the analysis of the IPCF network.

The IoT devices consist of smartphones, laptops, and other smartwatches for more interaction communication of data to the next offloading and caching sectors. The request from these devices moves to parts of created network, where the tasks are allocated for the queue sectionof the IPCF network. According to the areas in the task queue, the working of particular jobs will take place. After knowing the priority of the data, the required pieces of information are sent to the access point. An access point of the IPCF network

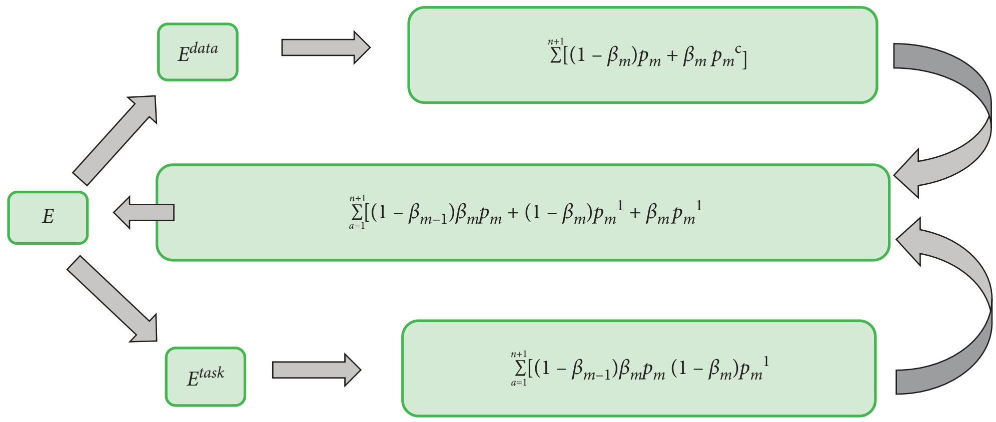

Figure 7: Systematic offloading and managerial migrant caching.

is a unit that allows data to the cloud for giving the respective response to the device. The particular response is moved to the task manager for sending it to the mobile device connected to it. These are done using the network officials, which are called interfaces, to reduce the delay time. After completing these tasks, the response is moved to the respective devices of IoT or mobile devices.

The monitoring of the intelligent edge processing of the IPCF network is done with respective equations and its solution. The references based on existing systems like RCOC-ECN, CT-ECE, TCO-CEF, IMEC-RC, ECNCP, DP-MO-ECS, and DACO-CMEC have been studied thoroughly. The implementation based on the IPCF model on intelligent edge processing with two methods is enclosed with its related equations and tabulations for future reference.

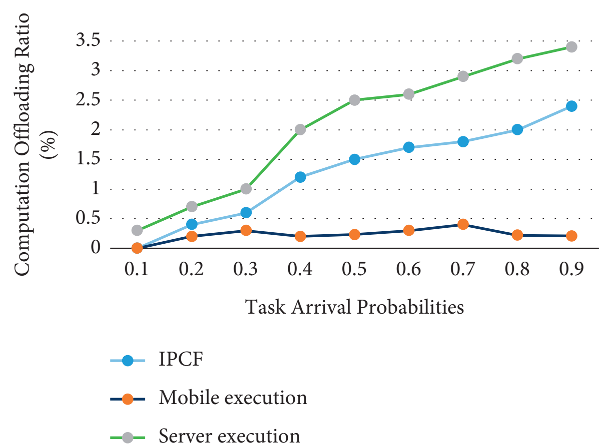

Figure 8: Computation offloading and scheduling.

4. Results and Discussion

As the computation offloading and caching is a little complicated because of delay measuring problems and traffic techniques. The emerged framework of IPCF has two standard methods of ratio for rendering it to all possible monitoring outcomes with more significant results. Many different categories are launched to restrict data traversing from source location to destination for less delay communication. Yet, the crafted system of IPCF needs systematic offloading techniques and managerial migrant caching to ensure the data transaction of mobile devices. There are extensive assessment techniques to ensure the directory terms, particularly challenging facts of the proposed system, have been fortunately determined.

4.1. Stochastic Computation Offloading and Scheduling. Figure 8 shows the graphical representation of computation offloading and scheduling technique of data communication given in equation (4).To monitor the system efficiency, the networking operator must adopt an optimum approach for downloading and transmitting planning throughout the infrastructure while ensuring that all cellular participants are allowed upkeep. The proposed system of IPCF provides the primary identity in offloading and caching section to data interaction. The mobile execution always depends on the mobile device that offers to form a source place for communication. The mobile device is primary in offloading and computational capabilities of statements. As a source file, the mobile device transfers data to edge computing to catch the information in caching segment. The server execution is the primary sector to execute the leading standard of files from the mobile device for storing in memory. Thus, the section of the proposed data system is satisfied with mobile execution and server execution.

4.2. Average Marginal Delay Sensitivity. Figure 9 provides the average marginal delay sensitivity in equation (13) of IPCF on some related data sets like 20, 50, and 80. Marginal delay is caused by an increase in delay time due to the previous growth in approaching volumes and its original opportunity cost. Owing to the state of the situation delays formulas, a marginal delayed means the measurement of the greatest latency for a certain traffic cycle.The conquering sections of the first 20 data sets result in more delay rate to transfer information for the cache memory. The next 50 sets give in moderate delay and traffic rate of communication of data for cloud computing. Marginal delay rate modifies from time to time according to the data usage and its storing capacity. A marginal latency is a crucial variable in the

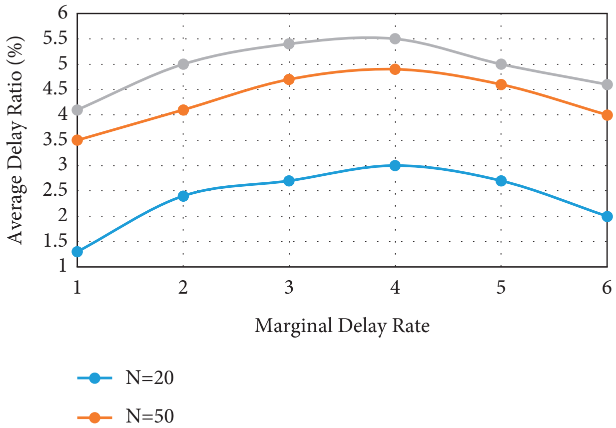

Figure 9: Average marginal delay sensitivity.

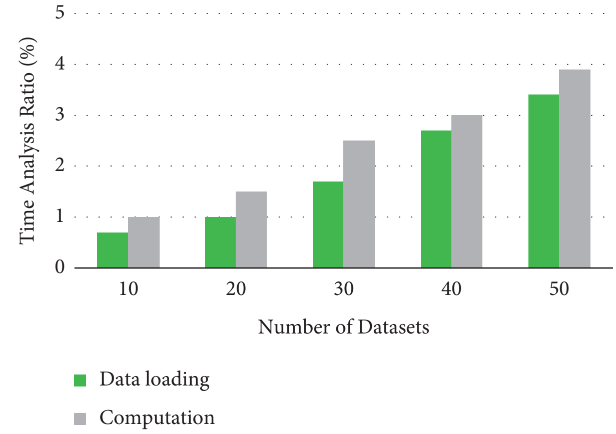

Figure 10: Data loading and computation time.

signals parameters computation, particularly if the quantities of several techniques are very varied. The last data set is computed to give less latency for data communication on rising and falling distribution time. Thus the computation sequence time is 4.6 in delay rate, as it is less when compared to other current systems.

4.3. Data Loading and Computation Time Analysis. Figure 10 illustrates the data loading and computation timing segments of data transfer from the mobile unit to cloud edge computing mentioned in equation (14). This section of the graph contains 10 to 50 data sets based on data loading and computation of offloading. Data loading means transferring and reloading information or groups to the cloud or the like from a data source, a directory, or a program. For example, it is typical to duplicate digitized sensor data and then paste or insert it into the data collection or processing application. In database recovery and reloading strategies, information loading is utilized. In general, this file is recorded differently from the source location in the programs that seek. Computation means all mathematical and nonarithmetic, following a well-defined paradigm—mechanical or electronic equipment for cloud computers and data transmission. The data loading works less due to delay sequence in transferring communication, and computation makes more for higher interaction. Thus, the 50 data sets emerge to provide less data loading and enhanced calculation as it is the main formulation of communication.

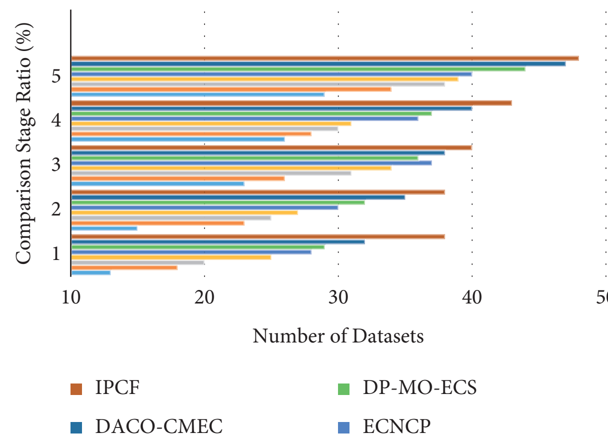

Figure 11: Comparison stage of methods.

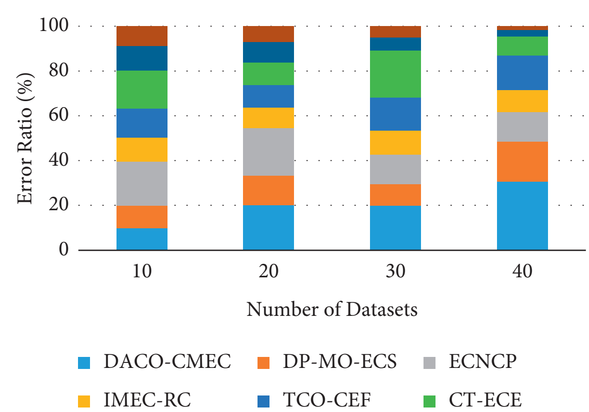

Figure 12: Error rate determination.

4.4. Comparison Stage of Methods. Figure 11 displays the comparative status of the methods currently used in equation (14) and starts an IPCF process that delivers a more excellent efficiency ratio. The data sets between 10 and 50 are compared to each other when the section values are committed on each stage. In the launched system, the downloading and caching of data from source to destination must be higher. In addition, the employment of intelligence should be necessary as technology is a trend now. The first 10 data sets are compared in sequence to increase the caching and offloading capacity. Thus, the first launch is less while comparing, and the second setting is implemented. When comparing the second set to the previous group shows the changes of increasing capacity. Meanwhile, other sets of 30, 40, and 50 are compiled for comparison. Thus, the last unit of derived values provides good efficiency.

4.5. Error Rate Analysis. Figure 12 shows the error rate determination of existing RCOC-ECN, CT-ECE, TCO-CEF, IMEC-RC, ECNCP, DP-MO-ECS, and DACO-CMEC proposed system of IPCF. The error control function and the error rate in the previously mentioned equation (13) are found with many sequential solutions and verified. The error rate is happened in many states of execution to know the specific error condition. As the existing and proposed systems are mingled to simplify, the percentage values for showing the error rate of the launched method are less compared to other systems.

As the discussion is based on many literature surveys, the importance and evaluation of computation offloading and caching are accomplished. The need for data transfer from one mobile network system to another end of computing is encountered using many equations. The created graphs and data are entirely enhanced based on comparing to existing methods of records. Thus, the level of determining the data sets used in IPCF, comparison stage of processes, data loading and computation time, average marginal delay sensitivity, computation offloading and scheduling, and error control rate is successfully implemented in IPCF for better growth.

5. Conclusion and Discussion

This research analyzed the user-assisted IPCF system for caching where the servers can cache the programs previously written for storage purposes selectively. The investigation of systemic download techniques and management migrant caching of an improved service is presented to minimize the delay in the calculation and mobile network energy consumption. The complex transformed problem becomes a pure problem by selecting the simple analytical expressions of the optimal resource allocation. As the proposed IPCF approach for network reconfiguration of caching is achieved by determining a judgment on offloading, and conversely. Comprehensive simulation results show that cache improvement results in significant mobile phone resource reductions compared to the other relevant benchmarking approaches. In addition, a proper fit between system speed and computer architecture is achieved with the optimal alternative caching strategy. IPCF helped conclude the investigation with specific exciting development guidelines for caching services. First, it is interesting to consider the adaptive allocation of resources in a typical multi-users IPCF network to increase usage efficiency. In specifically, cached mobile network programs might be distributed to lower the cost of program upload. Moreover, it poses several latest innovative concerns, including interloading data and privacy difficulties in exchanging resources. Secondly, the one-server configuration to a multi-server arrangement is equally interesting. Finally, the edge network can acquire publicly available programs from the network infrastructure, such that service data can be cached in advance to reduce calculation time further.

Data Availability

Data sharing is not applicable to this article as no data sets were generated or analyzed during the current study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] J. Kondragunta, R. Seidel, and G. Hirtz, "Machine learning based identification of elderly persons with cognitive impairment using dynamic time warping," *Current Directions in Biomedical Engineering*, vol. 6, no. 3, pp. 360–363, 2020.

[2] S. Jegadeesan, M. Azees, P. M. Kumar et al., "An efficient anonymous mutual authentication technique for providing secure communication in mobile cloud computing for smart city applications," *Sustainable Cities and Society*, vol. 49, Article ID 101522, 2019.

[3] T. G. Nguyen, T. V. Phan, D. T. Hoang, T. N. Nguyen, and C. So-In, "Efficient SDN-based traffic monitoring in IoT networks with double deep Q-network," in *Proceedings of the International Conference on Computational Data And Social Networks*, pp. 26–38, Springer, Montreal, Quebec, December 2020.

[4] R. K. Dash, T. N. Nguyen, K. Cengiz, and A. Sharma, "Fine-tuned support vector regression model for stock predictions," *Neural Computing & Applications*, pp. 1–15, 2021.

[5] S. Garg, A. Singh, K. Kaur et al., "Edge computing-based security framework for big data analytics in VANETs," *IEEE Network*, vol. 33, no. 2, pp. 72–81, 2019.

[6] G. Manogaran, B. S. Rawal, and M. Alazab, "A boosted tree classifier algorithm based collaborative computing framework for smart system," *IEEE Transactions on Network Science and Engineering*, vol. 9, 2020.

[7] A. Jolfaei, P. Ostovari, M. Alazab, I. Gondal, and K. Kant, "Guest editorial special issue on privacy and security in distributed edge computing and evolving IoT," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2496–2500, 2020.

[8] U. Verma and D. Bhardwaj, "Design of lightweight Authentication protocol for fog enabled internet of things- A centralized authentication framework," *International Journal of Communication Networks and Information Security*, vol. 12, no. 2, pp. 162–167, 2022.

[9] G. Manogaran, P. Shakeel, H. Fouad et al., "Wearable IoT smart-log patch: an edge computing-based Bayesian deep learning network system for multi access physical monitoring system," *Sensors*, vol. 19, no. 13, p. 3030, 2019.

[10] M. Sheibani and G. Ou, "The development of Gaussian process regression for effective regional post-earthquake building damage inference," *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, no. 3, pp. 264–288, 2021.

[11] S. Namasudra, D. Devi, S. Kadry, R. Sundarasekar, and A. Shanthini, "Towards DNA based data security in the cloud computing environment," *Computer Communications*, vol. 151, pp. 539–547, 2020.

[12] S. Vimal, Y. H. Robinson, S. Kadry, H. V. Long, and Y. Nam, "IoT based smart health monitoring with CNN using edge computing," *Journal of Internet Technology*, vol. 22, no. 1, pp. 173–185, 2021.

[13] B. Muthu, C. B. Sivaparthipan, G. Manogaran et al., "IOT based wearable sensor for diseases prediction and symptom analysis in healthcare sector," *Peer-to-peer networking and applications*, vol. 13, no. 6, pp. 2123–2134, 2020.

[14] S. Wang, Y. Zhao, J. Xu, J. Yuan, and C.-H. Hsu, "Edge server placement in mobile edge computing," *Journal of Parallel and Distributed Computing*, vol. 127, pp. 160–168, 2019.

[15] S. Wang, Y. Zhao, L. Huang, J. Xu, and C.-H. Hsu, "QoS prediction for service recommendations in mobile edge computing," *Journal of Parallel and Distributed Computing*, vol. 127, pp. 134–144, 2019.
[16] Y. Guo, S. Wang, A. Zhou, J. Xu, J. Yuan, and C. H. Hsu, "User allocation-aware edge cloud placement in mobile edge computing," *Software: Practice and Experience*, vol. 50, no. 5, pp. 489–502, 2020.
[17] P. P. Ray, D. Dash, and D. De, "Edge computing for Internet of Things: a survey, e-healthcare case study and future direction," *Journal of Network and Computer Applications*, vol. 140, pp. 1–22, 2019.
[18] G. Carvalho, B. Cabral, V. Pereira, and J. Bernardino, "Edge computing: current trends, research challenges and future directions," *Computing*, vol. 103, no. 5, pp. 993–1023, 2021.
[19] H. Kurdi, A. Alfaries, A. Al-Anazi et al., "A lightweight trust management algorithm based on subjective logic for interconnected cloud computing environments," *The Journal of Supercomputing*, vol. 75, no. 7, pp. 3534–3554, 2019.
[20] K. Senthilkumar, M. Palanivelan, V. Asokan, J. Karthi, and M. Lakshmanan, "Energy efficient hamming coded cooperative communication," *Telecommunications and Radio Engineering*, vol. 79, no. 17, pp. 1521–1528, 2020.
[21] M. Kaur, G. Kaur, P. K. Sharma, A. Jolfaei, and D. Singh, "Binary cuckoo search metaheuristic-based supercomputing framework for human behavior analysis in smart home," *The Journal of Supercomputing*, vol. 76, no. 4, pp. 2479–2502, 2020.
[22] Y. Miah, C. N. E. Prima, S. J. Seema, M. Mahmud, and M. Shamim Kaiser, "Performance comparison of machine learning techniques in identifying dementia from open access clinical datasets," *Advances on Smart and Soft Computing*, Springer, Singapore, pp. 79–89, 2021.
[23] G. Tefera, K. She, M. Shelke, and A. Ahmed, "Decentralized adaptive resource-aware computation offloading & caching for multi-access edge computing networks," *Sustainable Computing: Informatics and Systems*, vol. 30, Article ID 100555, 2021.
[24] C. Li, J. Tang, H. Tang, and Y. Luo, "Collaborative cache allocation and task scheduling for data-intensive applications in edge computing environment," *Future Generation Computer Systems*, vol. 95, pp. 249–264, 2019.
[25] F. Liu, Z. Huang, and L. Wang, "Energy-efficient collaborative task computation offloading in cloud-assisted edge computing for IoT sensors," *Sensors*, vol. 19, no. 5, p. 1105, 2019.
[26] R. Wang, M. Li, L. Peng, Y. Hu, M. M. Hassan, and A. Alelaiwi, "Cognitive multi-agent empowering mobile edge computing for resource caching and collaboration," *Future Generation Computer Systems*, vol. 102, pp. 66–74, 2020.
[27] J. Ren, D. Zhang, S. He, Y. Zhang, and T. Li, "A survey on end-edge-cloud orchestrated network computing paradigms," *ACM Computing Surveys*, vol. 52, no. 6, pp. 1–36, 2020.
[28] X. Zhao, K. Yang, Q. Chen et al., "Deep learning based mobile data offloading in mobile edge computing systems," *Future Generation Computer Systems*, vol. 99, pp. 346–355, 2019.
[29] G. Tefera, K. She, F. Deeba, and A. Ahmed, "Resource-aware decentralized adaptive computational offloading & task-caching for multi-access edge computing," in *Proceedings of the 2019 3rd High Performance Computing And Cluster Technologies Conference*, pp. 39–43, China, Guangzhou Shi, June 2019.
[30] H. Tang, "Analysis of the driving factors of building energy consumption growth in smart cities based on the STIRPAT model," *Journal of Interconnection Networks*, Article ID 2146008, 2021.
[31] G. Wang, Q. Zhai, and H. Liu, "Cross self-attention network for 3D point cloud," *Knowledge-Based Systems*, vol. 247, no. 247, Article ID 108769, 2022.