HAI, T., ZHOU, J., LU, Y., JAWAWI, D., WANG, D., SELVARAJAN, S., MANOHARAN, H. and IBEKE, E. 2023. An archetypal determination of mobile cloud computing for emergency applications using decision tree algorithm. *Journal of cloud computing* [online], 12, article 73. Available from: https://doi.org/10.1186/s13677-023-00449-z

An archetypal determination of mobile cloud computing for emergency applications using decision tree algorithm.

HAI, T., ZHOU, J., LU, Y., JAWAWI, D., WANG, D., SELVARAJAN, S., MANOHARAN, H. and IBEKE, E.

2023

Hai *et al. Journal of Cloud Computing* (2023) 12:73
https://doi.org/10.1186/s13677-023-00449-z

Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH **Open Access**

An archetypal determination of mobile cloud computing for emergency applications using decision tree algorithm

Tao Hai[1,2], Jincheng Zhou[1,3*], Ye Lu[4], Dayang N. A. Jawawi[2], Dan Wang[3,5], Shitharth Selvarajan[6*], Hariprasath Manoharan[7] and Ebuka Ibeke[8]

Abstract

Numerous users are experiencing unsafe communications due to the growth of big network mediums, where no node communication is detected in emergency scenarios. Many people find it difficult to communicate in emergency situations as a result of such communications. In this paper, a mobile cloud computing procedure is implemented in the suggested technique in order to prevent such circumstances, and to make the data transmission process more effective. An analytical framework that addresses five significant minimization and maximization objective functions is used to develop the projected model. Additionally, all mobile cloud computing nodes are designed with strong security, ensuring that all the resources are allocated appropriately. In order to isolate all the active functions, the analytical framework is coupled with a machine learning method known as Decision Tree. The suggested approach benefits society because all cloud nodes can extend their assistance in times of need at an affordable operating and maintenance cost. The efficacy of the proposed approach is tested in five scenarios, and the results of each scenario show that it is significantly more effective than current case studies on an average of 86%.

Keywords Cloud computing, Mobile nodes, Emergency applications, Gain, Energy

Introduction

In this paper, we implement a mobile cloud computing procedure in the proposed technique in order to prevent unsafe or difficulty in communication as a result of the growth of big network mediums. The proposed technique uses a machine learning method known as Decision Tree optimization algorithm with a number of subset features, and indulgent network information totally removes the impure cloud computing nodes from the network.

The following are the major contributions of this paper:

- An efficient simulation model is used to demonstrate how mobile cloud computing might be used to handle various emergency situations.
- The cloud nodes in the proposed approach can extend their assistance in times of need at an affordable operating and maintenance cost.

---

*Correspondence:
Jincheng Zhou
zjc81@sgmtu.edu.cn
Shitharth Selvarajan
shitharths@kdu.edu.et
[1] School of Computer and Information, Qiannan Normal University for Nationalities, Duyun 558000, China
[2] Faculty of Computing, Universiti Teknologi Malaysia (UTM), Skudai, Johor Bahru, Johor 81310, Malaysia
[3] Key Laboratory of Complex Systems and Intelligent Optimization of Guizhou Province, Duyun 558000, China
[4] School of Computer &Communication, Lanzhou University of Technology, Lanzhou 730050, China
[5] School of Mathematics and Statistics, Qiannan Normal University for Nationalities, Duyun 558000, China
[6] Department of Computer Science, Kebri Dehar University, Kebri Dehar, Ethiopia
[7] Department of Electronics and Communication Engineering, Panimalar Engineering College, Poonamallee, 600123 Chennai, Tamil Nadu, India
[8] School of Creative and Cultural Business, Robert Gordon University, Aberdeen, UK

- The mobile cloud computing nodes are designed with strong security, ensuring that all the resources are allocated appropriately.

The remainder of this paper is organized as follows: Literature survey section provides the literature survey. In Mobile computing: system model section, the mobile computing system model is presented. Optimization algorithm section presents the optimization algorithm. In Experimental verification section, an experimental verification of the proposed models and algorithms is carried out. Finally, Conclusion section concludes the study and provides the future scope.

Literature survey

This section looks at the methods for making fundamental decisions in order to help readers comprehend the many difficulties that arise when designing mobile computing platforms. Since not all data is sent in a consistent manner, users are unable to create computing networks in a common way. Therefore, in order to create a mobile computing network, devices must be tested before installation, which leads to a number of problems that must be analyzed. The proposed method is then put into practice to address some of the most significant shortcomings in the current system. Using differential equations, the computational procedure in [1] analyses the random nodes, leading to the formation of numerous straightforward expressions. A stationary distribution of mobile nodes with significant physical space is expressed by the simple expression form. However, the model of expression is different because the intended system lacks dimensions but still produces exponential case studies. In [2], a hypothetical framework was offered for making judgments using both an ideal and an adversarial mode of operation in order to provide a multidimensional model. The entire setup is accelerated since the appropriate mode of operation is offered, and the results are significantly higher than anticipated. However, the best method of data segmentation minimizes the latency duration of mobile computing processes. The computing procedure is introduced even on other application platforms [3], where emergency medical centres are identified by mobile locations. If the computing process is only focused on mobile locations, then a certain computing channel will experience significant traffic, increasing the design's complexity.

A system architecture was designed to use sensing techniques to make mobile computing tasks less complicated in [4]. The aforementioned technique effectively uses a code to link source mobile networks with destination mobile networks. Even though persistent models are utilized to detect feedback in such situations, steady state operation demand is still in inactive mode. More energy is given [5] to mobile computer nodes in order to activate steady state networks in the system, and this case can only be applied to emergency management systems. If the mobile computer system is used in emergency situations, all system technologies will significantly advance and heterogeneous functioning will not be guaranteed. These systems are entirely avoided because if heterogeneous operations are to be formed, network connectivity must be tested in every emergency situation, which is often not practicable. Given that mobile computing nodes are placed with high handover in numerous locations, adaptive strategies are also offered for various mobile devices to govern handover across the entire operation [6]. However, if the handover is greater, a specific cell region's analytical model needs to be defined. Utilizing queuing systems, the behavior of the model is examined, and it is shown that the overall handover of the computing process is decreased. However, in a real-time process, complete handover can be minimized in specified circumstances. Initial identification methods are also made utilizing edge computing techniques, and they are used with fifth generation networks in the transportation sector [7]. Even though a new computing paradigm has been defined, most apps don't work with modernized mobile computing platforms.

In [8], an inter-cloud system was designed for providing transfer functions during offloading calculations to address these issues. All fundamental criteria in a multi-cloud environment will be evaluated by the transfer function that is introduced for offloading situations. In contrast, the multi-cloud environment for online systems is far more user-defined, necessitating thorough analysis. Cellular networks will offer more flexibility for end-to-end operations if device-to-device contact is developed rather than employing a transfer function. In computing networks, even device-to-device communication is permitted, and a data survey is conducted in this kind of operation [9] to provide details on discovery and route processing strategies for various cloud computing nodes. Although the method of device communication enables users to accurately access all computing nodes, it neglects to look into the standardization efforts of newly launched mobile nodes. The basic architectural paradigm of cloud computing and information systems is also used to take additional measures for managing emergency systems with mobile computing nodes [10]. When the process is examined, it becomes clear that the architecture of mobile nodes must be modified at the appropriate times since power distribution across defined networks is improper.

A performance evaluation model was developed to assess the maximum loss in the entire network with optimal resource allocation [11–13] in order to allocate the right amount of power in the network. However, because the decision-making process in cloud computing is so complex, the greatest degree of loss is seen there. So, after looking at the problems with different approaches, an analytical model is used to come up with a solution that allows all cloud computing operations to be used in emergencies in the best way possible [14–16]. In [17], the authors proposed an algorithm for the security and privacy of health information making use of the Modular Encryption Standard (MES). The authors in [18] presented an offloading technique for multi-access edge computation in a high demand Internet of Things network for energy efficiency and consumption reduction. In [19], the authors proposed a trust-based routing protocol to enhance the performance and quality of service of the Mobile Ad-hoc Network (MANET). The authors in [20] introduced an energy efficient architecture based on the cloud and the Internet of Everything (IoE) to optimize the consumption of energy and reduce data traffic. In [21], the authors designed technology for data privacy in the IoT based on the cloud technology, which hides the data transmission information between the edge servers and the cloud. The proposed algorithm can be modified to deal with real-time applications using edge computing as explained in [22].

S/N	Authors	Contribution
[1]	Garetto M, Leonardi	Using differential equations, the computational procedure proposed analyses the random nodes, leading to the formation of numerous straightforward expressions
[2]	Rathee G, Garg S, Kaddoum G, et al	A hypothetical framework was proposed for making judgments using both an ideal and an adversarial mode of operation in order to provide a multidimensional model
[3]	Hatami-Marbini A, Varzgani N, Sajadi SM, Kamali A	This research uses simulation-based and mathematical modelling optimization approaches to detect the best and closest location of medical centers in case of emergencies
[4]	Bruno R, Conti M, Gregori E	A system architecture was designed to use sensing techniques to make mobile computing tasks less complicated using a code to link source mobile networks with destination mobile networks
[5]	Ramasamy V, Gomathy B, Sarkar JL, et al	An emergency management system was proposed which uses Bluetooth technology to follow peer-to-peer communication to reduce the workload of mobile devices
[6]	Kim C, Dudin A, Dudin S, Dudina O	A multi-server queueing system model was introduced for long-term storage of new users in a communication network
[7]	Lamb ZW, Agrawal DP	An architecture is proposed which analyzes available real-time resources and allocates to the most feasible and logical resource in order to reduce networking overhead
[8]	Dou Y, Ho YH, Deng Y, Chan HCB	An inter-cloud system was designed for providing transfer functions during offloading calculations to address these issues
[9]	Gandotra P, Member S, Jha RK D2D	A data survey was conducted in computing networks, and the permission of device-to-device communication to provide details on discovery and route processing strategies for various cloud computing nodes
[10]	Mitropoulos S, Mitsis C, Valacheas P, Douligeris C	A novel medical information system for emergencies was presented which simulates the services the Greek National Instant Aid Centre provides
[11]	Krishna Keerthi Chennam, Rajanikanth Aluvalu and S.Shitharth,'	This work designed an integrated framework of an attribute-based multistage encryption standard for providing security and data confidentiality
[12]	Han S, Ma D, Kang C, et al	An offloading model that uses mobile edge computing was proposed to solve the issue of high mobile cloud computing technology
[13]	Nanda S, Panigrahi CR, Pati B	A detailed survey on mobile cloud computing and emergency medical system applications was conducted, and possible solutions to the design and development challenges were proposed
[14]	Poulymenopoulou M, Malamateniou F, Vassilacopoulos G (	An integrated EMS framework based on cloud computing was proposed which provides authorized users with access to emergency case information for exchanging data with hospitals
[15]	Rajanikanth Aluvalu, V.Uma Maheswari, Krishna Keerthi Chennam and S.Shitharth,'	A dynamic access control model is proposed for the security of cloud-stored data, and for providing users with access to the data
[16]	B. Thirumaleshwari Devi, S.Shitharth	A study was conducted on the most common security attacks and breaches, especially honey pot, in cloud computing
[17]	Shabbir, Maryam & Shabbir, Ayesha & Iwendi, Celestine & Javed, Abdul Rehman & Rizwan, Muhammad & Herencsar, Norbert & Lin, Chun-Wei	The authors proposed an algorithm for the security and privacy of health information making use of the Modular Encryption Standard (MES)

S/N	Authors	Contribution
[18]	Anajemba, Joseph & Yue, Tang & Iwendi, Celestine & Alenezi, Mamdouh & Mittal, Mohit	The authors presented an offloading technique for multi-access edge computation in a high demand Internet of Things network for energy efficiency and consumption reduction
[19]	Sirajuddin, Mohammad & Rupa, Ch & Iwendi, Celestine & Biamba, Cresantus	The authors proposed a trust-based routing protocol to enhance the performance and quality of service of the Mobile Ad-hoc Network (MANET)
[20]	Priya,, Swarna & Bhattacharya, Sweta & Reddy, Praveen & Somayaji, Siva & Lakshman, Kuruva & Kaluri, Rajesh & Hussien, Aseel & Gadekallu, Thippa	The authors introduced an energy efficient architecture based on the cloud and the Internet of Everything (IoE) to optimize the consumption of energy and reduce data traffic
[21]	Wang, Tian & Quan, Yang & Shen, Xuewei & Gadekallu, Thippa & Wang, Weizheng & Dev, Kapal	The authors designed technology for data privacy in the IoT based on the cloud technology, which hides the data transmission information between the edge servers and the cloud
[22]	T. Gadekallu, Quoc-Viet Pham, Dinh C. Nguyen, P. Maddikunta, N. Deepa, B. Prabadevi, P. Pathirana, Jun Zhao, W. Hwang	A comprehensive review of developments in and applications of blockchain, edge computing, and Internet of Things (IoT) was presented

Mobile computing: system model

Given that both the transmitter and receiver must be integrated into the closed loop system, the mobile computing process needs to be assessed using an analytical model. For the purpose of computing proper channel processing in all system formations, the surrounding environments are also examined in this factor. Without an analytical model, random transmission occurs, making it impossible for users to recognize sent packets—even in an emergency situation. Additionally, each packet in the suggested method has a security model established [23, 24] allowing for improved resource management during emergency situations. So, using Eq. (1), here is how you can figure out how much energy each packet uses:

$$Energy_i = min\sum_{i=1}^{n} P_p(i) + T_p(i) \tag{1}$$

Where,

P_p, T_p denotes power and time periods of processing units respectively.

According to Eq. (1), energy usage must be reduced by giving each packet node in the network an equal amount of electricity. The steps involved in giving a specific node packet power can be broken down into the following categories as given in Eq. (2):

$$P_p = r_p(i) + b_p(i) + w_p(i) + q_p(i) \tag{2}$$

Where,

r_p, b_p denotes rename and branch powers respectively,

w_p, q_p describes window and queue powers for different loads in the packets respectively.

Mobile devices compute the energy based on the Signal-to-Noise Ratio (SNR), which must be minimized in addition to varied energy constraints and the energy consumption during input and output operation changes.

$$SNR_i = min\sum_{i=1}^{n} \frac{\delta_i + O_1}{g_i + O_i} \tag{3}$$

Where,

δ_i denotes the threshold level of transmitting and receiving devices,

O_1, O_i represents first and last observation periods,

g_i describes gain of computing process.

The SNR, where the second objective function is framed by observing the gain of the mobile computing process, is denoted by Eq. (3). The suggested method establishes a framework to avoid SNR in the computation process by minimizing the number of loads in the network as given in Eq. (4) below:

$$L_i = min\sum_{i=1}^{n} \rho_i + \alpha\omega_i \tag{4}$$

Where,

ρ_i indicates weight of the network packets in computing process,

ω_i denotes the number of rationalized packets in the network.

Equation (5) shows that the packet nodes of all programmes used in mobile computing are based on external weighing variables. This is different from Eq. (4), where the load depends only on the internal weight of the system.

$$UL_i = min\sum_{i=1}^{n} \vartheta_l + \vartheta_{hand,out} \tag{5}$$

Where,

ϑ_l, $\vartheta_{hand,out}$ describes the number of restrictions in the server.

Implementing hand-in and out systems in server configuration will help to reduce the number of limits in mobile computing processes. Therefore, by utilizing slot configuration, which is defined by Eq. (6), the computing effort in the transmission process must be raised as follows:

$$slot_{tran} = \gamma_i + c_i \tag{6}$$

Where,

γ_i denotes attempt of transmission in computing process,

c_i indicates collision of packets during transmission.

Because nodes may become inactive during the transmission process in the event of a collision, the active period of nodes is determined using Eq. (7) as shown below:

$$active_i = max\sum_{i=1}^{n} If_{\,i} * sn_i \tag{7}$$

Where,

$If_{\,i}$ denotes operating frequency of nodes,

sn_i indicates sensing units of connected network.

Optimization algorithm

An algorithm's optimization section is utilized to establish where the appropriate models should be situated in relation to analytical models that are constructed for computing categories. The proposed method integrates a portion of an Artificial Intelligence (AI) algorithm utilizing Machine Learning (ML) techniques since mobile computing [25, 26] needs to be determined as an automatic process. The main benefit of using machine learning as an optimization technique is that it allows for accurate computation using modern mechanisms that make all historical data readily visible after undergoing any number of intricate processing steps. The learning method used in this procedure is known as supervised learning since the computing process automatically trains the specified network. Such a learning process begins with making specific decisions only on the basis of trial and error. As a result, if the computing process fails, the best knowledge about the failure instances may be obtained. In the following sequence, the computing process avoids such cases. In order to handle the challenges of computing classification and regression problems, the proposed method is also employed with a decision tree contrivance. It is indicated that only during the initial stages of the decision tree will the whole training data be taken into account, whereas following the evaluation of a specific stage, feature values are preferred, and they are ordered in a specific order. The aforementioned structures enable continuous operation of the prediction process, storing all attribute values during transmitter and receiver computation. The main benefit of decision trees is that all internal mobile nodes are ordered in accordance with a set of attribute systems, which are described in the following mathematical form:

$$E_i = \sum_{i=1}^{n} \Delta_i \log(\Delta_i) \tag{8}$$

Where,

Δ_i indicates the probability of sample computing matrix.

The process of mobile computing requires that gain must be maximized with a set of positive values. Therefore, the gain of two different subsets are formulated using Eq. (8) as follows:

$$gain_i = max\sum_{i=1}^{n} E_i - \{s_i, s\} \tag{9}$$

Where,

$\{s_i, s\}$ indicates subset of entropy functions.

The gain can also be measured using set of mobile classes with index terms and it is represented using Eq. (10) as follows:

$$index_{gain}(i) = \sum_{i=1}^{n} 1 - E_i^2 \tag{10}$$

In the proposed method, it is not possible to observe the values of mobile nodes under the same category. Thus, it is essential to provide a necessary split up model using the decision tree process. The split ratio for mobile computing can be formulated using Eq. (11) as follows:

$$R_g(i) = \sum_{i=1}^{n} \frac{gain_i + index_{gain}}{\sigma_i} \tag{11}$$

Where,σ_i describes mobile node information that is separated across different networks.

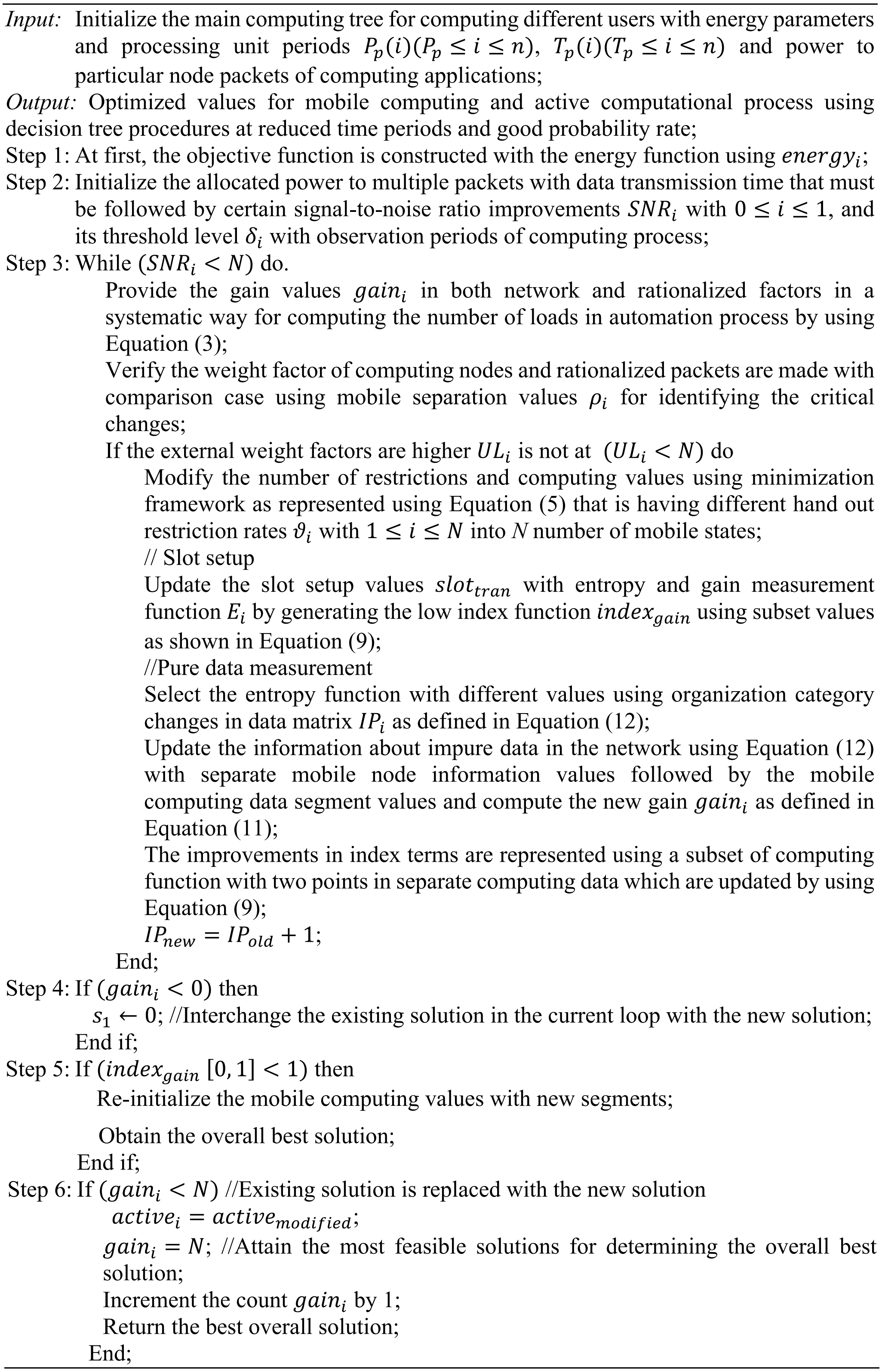
The mobile nodes will have impure data in the system, thus, it needs to be removed (Fig. 1). The analytical model for identifying the impure data is formulated using Eq. (12):

$$IP_i = min\sum_{i=1}^{n} \varphi_i * pure_i \tag{12}$$

Where,

φ_i indicates organization category of computing nodes,

$pure_i$ describes available data in pure arrangements.



```
Input: Initialize the main computing tree for computing different users with energy parameters
       and processing unit periods $P_p(i)(P_p \leq i \leq n)$, $T_p(i)(T_p \leq i \leq n)$ and power to
       particular node packets of computing applications;
Output: Optimized values for mobile computing and active computational process using
decision tree procedures at reduced time periods and good probability rate;
Step 1: At first, the objective function is constructed with the energy function using $energy_i$;
Step 2: Initialize the allocated power to multiple packets with data transmission time that must
        be followed by certain signal-to-noise ratio improvements $SNR_i$ with $0 \leq i \leq 1$, and
        its threshold level $\delta_i$ with observation periods of computing process;
Step 3: While ($SNR_i < N$) do.
            Provide the gain values $gain_i$ in both network and rationalized factors in a
            systematic way for computing the number of loads in automation process by using
            Equation (3);
            Verify the weight factor of computing nodes and rationalized packets are made with
            comparison case using mobile separation values $\rho_i$ for identifying the critical
            changes;
            If the external weight factors are higher $UL_i$ is not at ($UL_i < N$) do
                Modify the number of restrictions and computing values using minimization
                framework as represented using Equation (5) that is having different hand out
                restriction rates $\vartheta_i$ with $1 \leq i \leq N$ into N number of mobile states;
                // Slot setup
                Update the slot setup values $slot_{tran}$ with entropy and gain measurement
                function $E_i$ by generating the low index function $index_{gain}$ using subset values
                as shown in Equation (9);
                //Pure data measurement
                Select the entropy function with different values using organization category
                changes in data matrix $IP_i$ as defined in Equation (12);
                Update the information about impure data in the network using Equation (12)
                with separate mobile node information values followed by the mobile
                computing data segment values and compute the new gain $gain_i$ as defined in
                Equation (11);
                The improvements in index terms are represented using a subset of computing
                function with two points in separate computing data which are updated by using
                Equation (9);
                $IP_{new} = IP_{old} + 1$;
             End;
Step 4: If ($gain_i < 0$) then
            $s_1 \leftarrow 0$; //Interchange the existing solution in the current loop with the new solution;
        End if;
Step 5: If ($index_{gain}$ [0, 1] < 1) then
            Re-initialize the mobile computing values with new segments;
            Obtain the overall best solution;
        End if;
Step 6: If ($gain_i < N$) //Existing solution is replaced with the new solution
            $active_i = active_{modified}$;
            $gain_i = N$; //Attain the most feasible solutions for determining the overall best
            solution;
            Increment the count $gain_i$ by 1;
            Return the best overall solution;
        End;
```

Algorithm 1. Decision Tree Optimization (DCO)

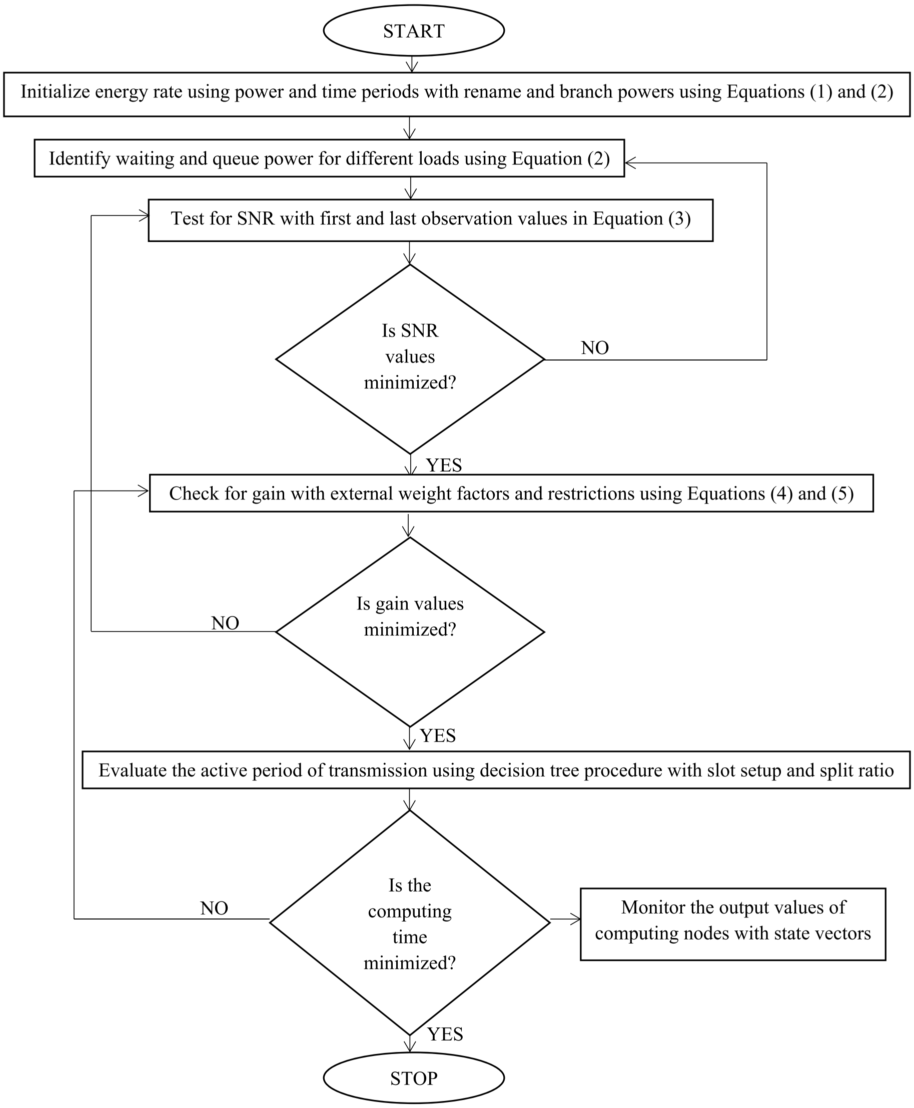


Fig. 1 Flowchart of mobile computing using decision tree

Experimental verification

The analytical model, which is referred to as a functional aspect of mobile computing, is examined and validated in order to demonstrate its efficacy in actual implementation scenarios. A loop-based system is used to create all analytical equations, and a separate hardware setup device is set up. The initial stage is completed with the presumption that all of the solutions are time-invariant. In order to prevent frequent path breaks in computing systems, additional time durations are taken into consideration once some transmission stages are finished. With more than 670 mobile cloud computing devices taking into account various routes, the suggested technique is examined clearly in both transmission and reception scenarios. Each mobile cloud computing device is built with the ability to send data to a destination and store that data in the cloud with encrypted codes. A unique cloud platform with network connectivity is established for each mobile computing setup using the node installation technique. Therefore, once the data transmission is finished, the nodes that are deployed in the cloud can transition to unload. When using the suggested method for mobile computing cases to look at every possible way to send data, the following things are taken into account:

- Scenario 1: Energy of computing nodes
- Scenario 2: Signal to Noise Ratio
- Scenario 3: External weighting factors
- Scenario 4: Activation periods of mobile nodes
- Scenario 5: Gain of decision tree

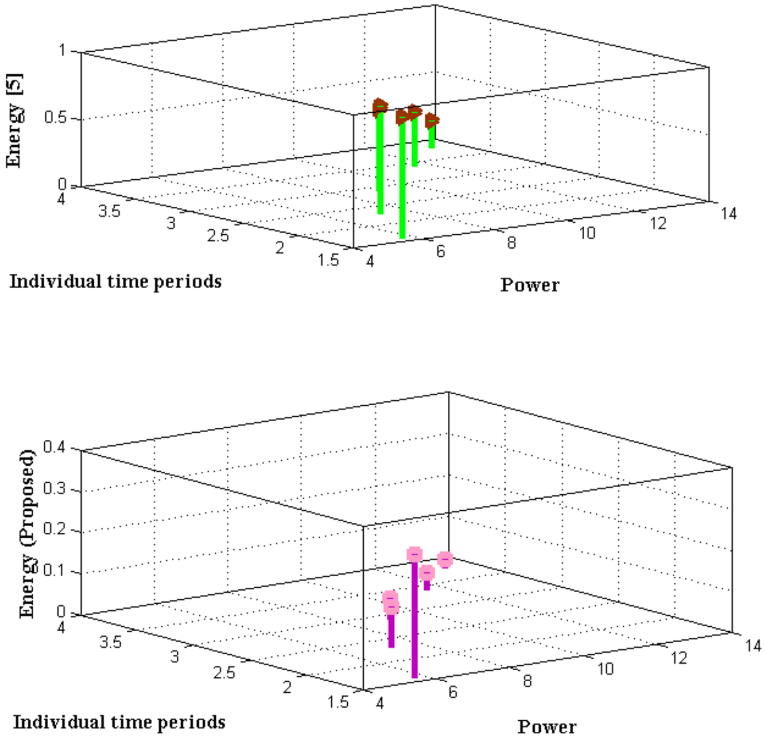


Fig. 2 Energy measurements of computing nodes

With the help of the MATLAB toolbox, all the aforementioned scenarios are directly simulated, and the programmability of mobile nodes is also examined. In order to demonstrate the effectiveness of the design, it is compared to the existing methods that are applied to the identical scenarios. The integrated hardware arrangement includes a wireless module with extra battery life that is not required throughout the entire network. Thus, in a network where only stable functioning is guaranteed, network complexity is fully decreased. Some adjustments are made manually to ensure stable functioning since resources must be correctly distributed throughout the system. The scenarios are described below.

Scenario 1: energy of computing nodes

Different types of mobile computing systems are used, and each form requires the integration of a certain number of mobile nodes in order to provide the receiver with the proper data transfer. In order to define the processing units, the amount of power supplied to mobile nodes must be adjusted appropriately. Additionally, the amount of time needed to process each unit must be reduced while using a finite amount of power. If both are increased, each node will be represented by a different energy. The aforementioned energy representation example necessitates careful testing because in an emergency, mobile nodes must be energy source error-free. As a result, the suggested solution uses the rename, branch, window, and queue powers in turn to equally distribute the energy allocated to mobile nodes. A significant power supply unit is set aside in addition to all the other power sources as a secondary backup during transmission times. The amount of energy delivered to mobile nodes is seen in Fig. 2.

According to Fig. 2 and Table 1, mobile nodes receive 5.69, 7.23, 9.02, 11.56, and 13.28 watts of power apiece, with corresponding time periods of 1.6, 2.3, 2.9, 3.4, and 3.8 for each amount of power delivered. The energy usage is monitored and compared with existing models for these supply power and time periods [6]. It is obvious that the suggested approach uses less energy to operate, and even when using minimal energy, the data is successfully delivered to the target location. However, the energy

Table 1 Energy representation

Power	Time period	Energy [5]	Energy (Proposed)
5.69	1.6	0.9	0.3
7.23	2.3	0.8	0.1
9.02	2.9	0.6	0.05
11.56	3.4	0.4	0.04
13.28	3.8	0.2	0.02

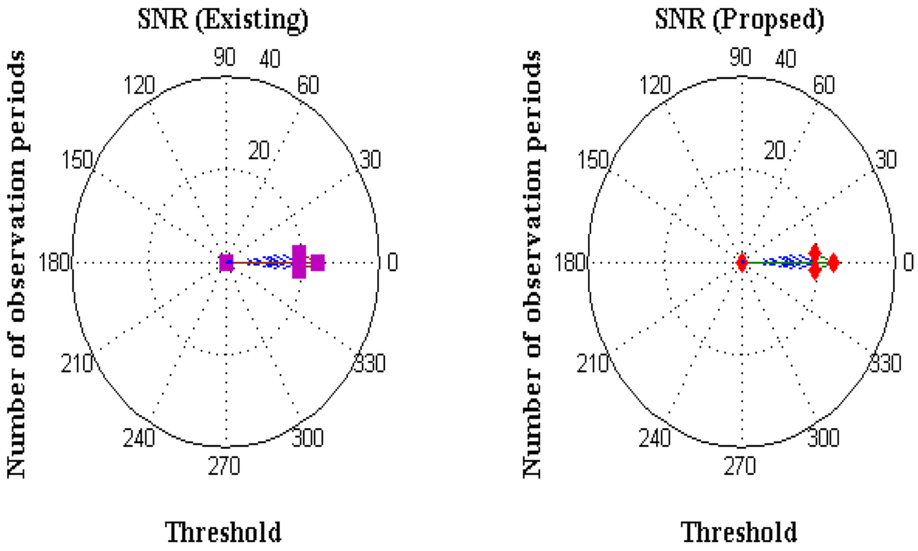


Fig. 3 Signal to noise ratio with threshold factor

Table 2 Signal to noise ratio with observation periods

Threshold levels	Number of observation periods	SNR [5]	SNR (Proposed)
7.24	10	2	0.8
9.23	14	1.8	0.6
12.45	16	1.4	0.3
13.92	19	1.3	0.1
14.36	24	1	0

of mobile computer nodes is only partially optimized by the methods now in use, leaving more energy untapped. This can be demonstrated by the 9.02 mill watts of power that is delivered to each CPU node every 2.9 s. This period sees the suggested method's energy consumption drop to 0.05 from the old method's 0.6. Hence, it is important to avoid this type of energy increase because it will seriously harm all of the network's computing nodes.

Scenario 2: Signal to Noise Ratio

The Signal-to-Noise Ratio (SNR) is derived using threshold levels and observation times to improve quality in mobile computing nodes. The device operates at a low noise factor thanks to the analytical model that is created for SNR in the suggested technique. In comparison to systems with high noise factors, the complete system will deliver higher computing quality due to its low noise factor. Threshold levels are measured in this scenario for both transmitting and receiving devices, and they must be within 0 decibel points. When a device has a high noise factor, the SNR for recognizing the device grows exponentially. The observation periods are measured in order to determine whether the noise level has increased. Any one observation period is then used for delivery cases. When better network services are offered at a high percentage of times, the chosen observation periods are then separated from the gain values. The comparison values of SNR for the computation process are shown in Fig. 3.

Figure 3 and Table 2 shows that the respective threshold values for mobile computing nodes are 7.24, 9.23, 12.45, 13.92, and 14.36. The first and last observation periods, which vary at different step sizes, are discovered to be 10 and 24 using the aforementioned threshold values. The number of observation periods can be thought of in a random manner, where benefit is maximized as a result of such changes. Additionally, the suggested method isolates the prescribed values when the gain is maximized, resulting in the achievement of acceptable SNR values. It is also practical that the suggested method provides low noise levels for the whole observation period, even if it exceeds the required intervals. With threshold values of 12.45 and observation periods of 16, it is evident from the comparison scenario that the proposed technique's SNR is significantly lower than the existing method's. In this situation, the SNR is 1.4 decibels for the existing method and 0.3 decibels for the projected approach.

Scenario 3: external weighting factors

Since the SNR for each connected mobile node can be directly reduced, this scenario offers experimental results

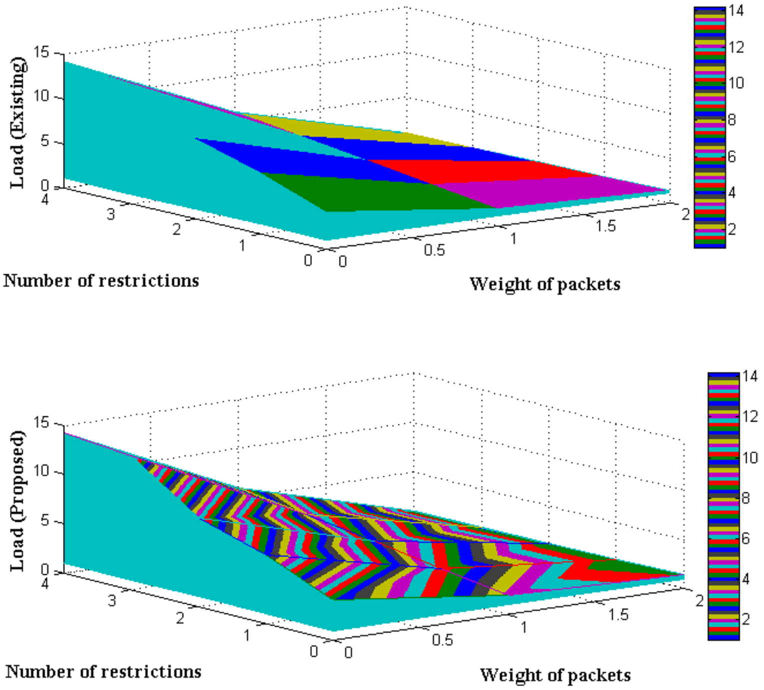


Fig. 4 Total loads with restrictions

based on load allocation because the SNR is reduced but cannot be entirely avoided in the network system. For the experimental scenario, the system rationalises the packets using both internal and external weighting factors. By avoiding the SNR at each network, the weight of specific networks is measured, and because of this, resources are not distributed equally in this situation. Although a separate protocol is not created in the proposed method, some server constraints are also reduced to some extent. The mobile computing nodes are defined in each slot because there is no set of regulations, and the total weight of the slots is taken into account. The system's total number of nodes is looked at and compared. The results are shown in Fig. 4 as hand and out weights, which are said to be the main cause of an increase in external weights.

According to Fig. 4 and Table 3, it is reasonable to assume that packet weights are distributed at factors of 4.16, 6.94, 9.01, 13.57, and 14.21, respectively, and that the number of limitations varies for each weight factor in steps of 1, as 2, 3, 4, and 5. The limitations are applied continuously because at certain stages of cloud computing, a higher restriction would force network packets to maintain a steady state. Even further restrictions would cause the entire network to degrade from a point when load is not required. There is a slight increase in weights determined relative to initial determinations in this comparison case where individual weight factors of packets are shown to be in addition to total weight. The various limits on the data transit paths are evident from the 14.21 g of weight per packet. In the case of the projected method using the appropriate decision tree, these constraints result in a weight reduction of the packet of 0.71 g. However, the weight can only be reduced by 1.03 g with the current system, so decisions are not being made properly.

Table 3 Individual load with restrictions

Weight of packets	Number of restrictions	Load [5]	Load (Proposed)
4.16	2	1.41	0.9
6.94	3	1.22	0.86
9.01	4	1.19	0.82
13.57	5	1.05	0.74
14.21	6	1.03	0.71

Scenario 4: activation periods of mobile nodes

The proposed approach is constructed in such a way that all nodes cannot be manipulated simultaneously. When connecting distinct cloud computing nodes utilizing onboard processes, it is required to activate each node using separate frequencies in order to increase its performance. When only one cloud computing node is present, figuring out the nodes' activation times is significantly easier. Additionally, the network operation

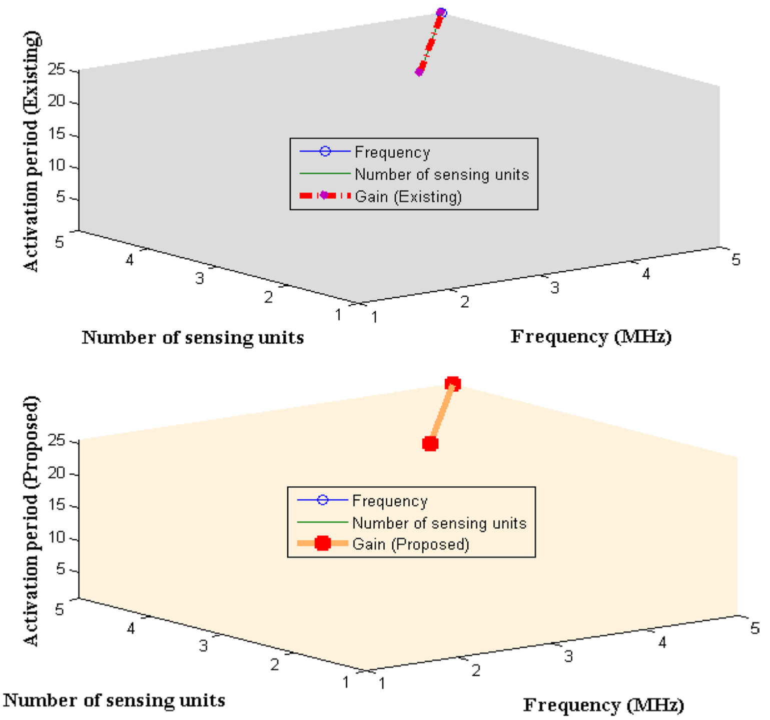


Fig. 5 Activation period of computing nodes

requires that the cloud sensing units always be connected in a fashion that permits sharing of the available frequencies. However, the projected technique makes it much more difficult to share the frequency of operating numerous nodes at once since the introduction of the decision tree procedure. The remaining frequencies are thus fully occupied by following cloud computing nodes in flexible circumstances. Because of what was said above, only the strategy shown in Fig. 5 makes the most of the active periods.

Figure 5 and Table 4 show that the activation duration of cloud computing nodes is maximized for the suggested strategy due to a number of factors. For the purposes of the verification scenario, frequency changes of 8.12, 12.25, 16.79, 21.34, and 25.5 MHz are taken into consideration with sensing units of 4, 8, 12, 16, and 20 accordingly. In each instance, the appropriate activation periods are chosen by duplicating each frequency term with the appropriate number of cloud sensing units. When 16.79 MHz is taken into account, 12 sensing units are found, and all activation periods must be larger than 1. The fact that the projected technique reproduces the activation period starting at 1 s shows that all relevant parameters have been examined and the frequency is being delivered in the right manner. However, the activation period is decreased by the current method [5] even though relevant components have been tested due to poor frequency allocation and sharing. This may be demonstrated using the aforementioned frequency and sensing units, where computing nodes activate every 2.3 s and existing cloud node segments activate at most once every 0.9 s.

Table 4 Activation period of cloud computing nodes

Frequency	Number of sensing units	Activation period (Existing)	Activation period (Proposed)
8.12	4	0.2	1
12.25	8	0.3	1.4
16.79	12	0.6	1.9
21.34	16	0.9	2.3
25.5	20	1	2.6

Scenario 5: gain of decision tree

The proposed method is employed to express a large number of external parameters and computing node attributes as a subset in order to calculate the gain of the integrated optimization process. In mobile cloud computing, the entropy values are tested using expectation probability values, which are explicitly determined using logarithmic values, to ascertain the decision tree's gain. The decision tree's entropy function is offered as a subset, allowing index gain to be

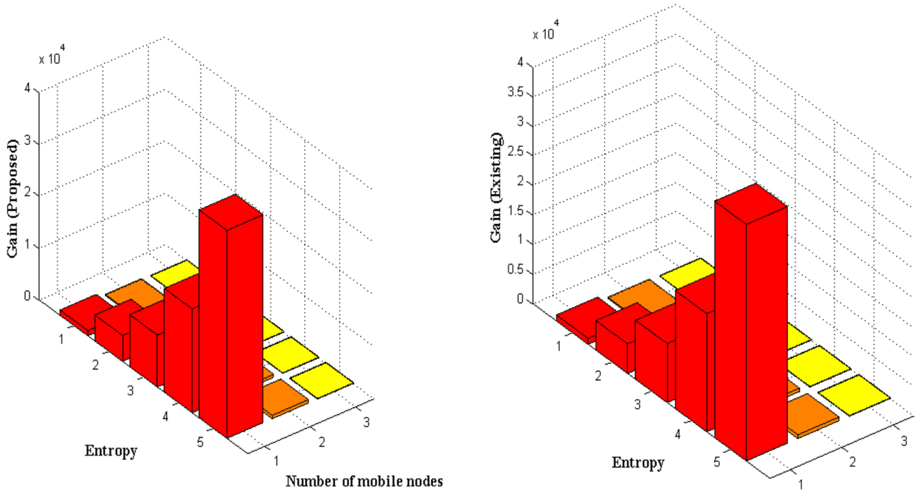


Fig. 6 Comparison of gain values

Table 5 Comparison of gain (Proposed Vs Existing)

Number of mobile nodes	Entropy	Gain [5]	Gain (Proposed)
1000	230	72	83
5000	375	76	86
10000	425	81	93
20000	565	84	95
40000	635	87	98

calculated for various cases. The amount of information present in each computer node must be verified during the index gain process, and in this case, the total amount of information spread over the entire network is seen. All impurities are eliminated from the system once the gain has been maximised, and only pure data is sent to cloud computing systems. The entire degree of computing nodes is discernible during this removal process, which is simulated and depicted in Fig. 6.

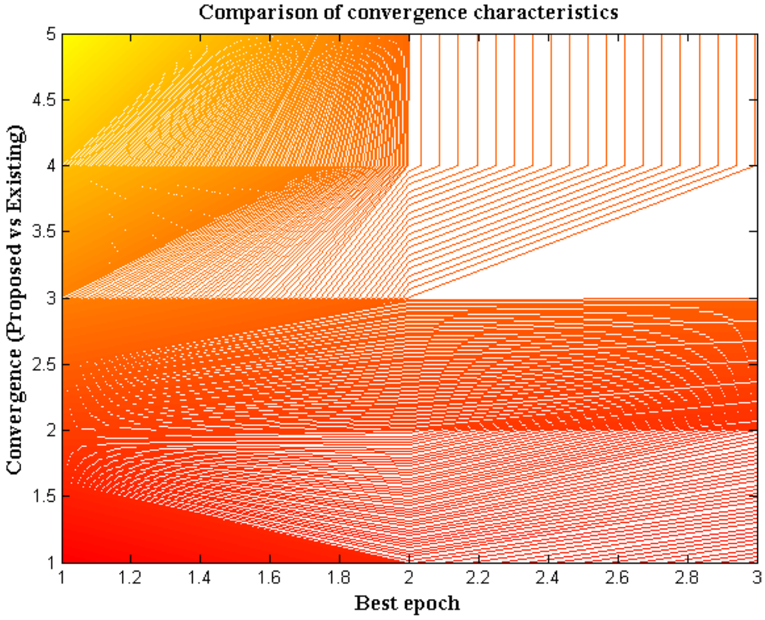


Fig. 7 Early convergence representation

Table 6 Convergence characteristics with best iteration periods

Best epoch	Convergence [5]	Convergence (Proposed)
20	32	30
40	39	35
60	50	50
80	53	50
100	53	50

Table 7 Comparison of robustness

Best epoch	Robustness [5]	Robustness (Proposed)
20	75	97
40	72	92
60	77	96
80	73	93
100	78	95

The maximizing of gain for cloud computing that is created via a decision tree approach is shown in Fig. 6 and Table 5. For experimental verification, between 1000 and 40,000 mobile nodes are taken into consideration, and the entropy values for each mobile node are calculated. Each mobile node has an entropy value that ranges from 230 to 635 and is defined by index ranges. Additionally, the index ranges are examined using various probability values; thus, the percentage of gain is calculated. It is seen during the comparison case that the proposed method, which uses a decision tree, provides a high gain in comparison to the existing cases. This can be demonstrated using 10,000 mobile nodes with an entropy of 425, where the projected method's gain is 93% and the existing method's gain is 81%. The projected approach with a decision tree offers excellent gain values even for mobile nodes that are constantly increasing, and a gain of about 98% is attained for networks with many computing nodes.

Performance evaluation

The best optimization for real-time cloud computing applications is chosen based on the performance evaluation that is simulated in this section. The smooth operation of mobile cloud computing depends on the integration of numerous distinct algorithms, so it is crucial to have an optimization tool that works well with an analytical working model. It is therefore preferable to distinguish between the status of transmitting and receiving devices, so a comparative case study is carried out using the best iteration cases. All parametric values must also be able to be evaluated by the integrated algorithm with proof-of-concept determination. Thus, the following evaluation is done for two key aspects of cloud computing:

- Case study 1: Conjunction characteristics
- Case study 2: Robustness characteristics

Case study 1: conjunction characteristics

High efficiency is attained if the data that is transmitted and received by the transmitting and receiving devices are converging quickly. If a decision tree is used in a machine learning system, the aforementioned scenario is possible. The fact that each subset's values are independently kept and the complete set is combined during the final loop is a key factor in decision trees' ease of convergence. Therefore, when the values of individual computing cases are stored inside the subset, individual cloud computing convergence increases. It is not very simple to ascertain how a computing process works, thus, index values are picked that provide suitable optimized values. A separate factor is used to remove impure data in order to achieve early convergence. Figure 7 simulates and depicts the convergence characteristics with the best iteration values.

It is evident from Fig. 7 and Table 6 that the decision tree's convergence point occurs significantly more quickly than it does for previous models. The decision tree converges at an early index point that starts from 60 while optimization in the existing model [5] fails to converge at an earlier rate, so 80 iteration periods are provided as the main index converging point. This can be shown using the five best epochs in step sizes of 20, 20, 40, 60, 80, and 100. Because every node in a decision tree converges at an early stage, it is significantly simpler to arrive to predicted solutions overall.

Case study 2: robustness characteristics

The strength of cloud computing nodes is studied in this case study, as well as the tolerance limit under all environmental circumstances. In general, computing nodes have substantially higher installation strengths than they do once the data has been sent. However, with mobile cloud computing, when more data branches are saved in a subset and only specific strengths are assessed, the aforementioned issue can be avoided if decision tree optimization is implemented. As a result, the process' overall efficiency increases to some amount while maintaining the functional aspects of cloud computing. The effectiveness of the suggested and existing methods is shown in Fig. 8.

It is clear from Fig. 8 and Table 7 that the strength of the computing process utilizing a decision tree is significantly larger than the example [5] at hand. Similar epoch values from a prior case study are picked in varied step

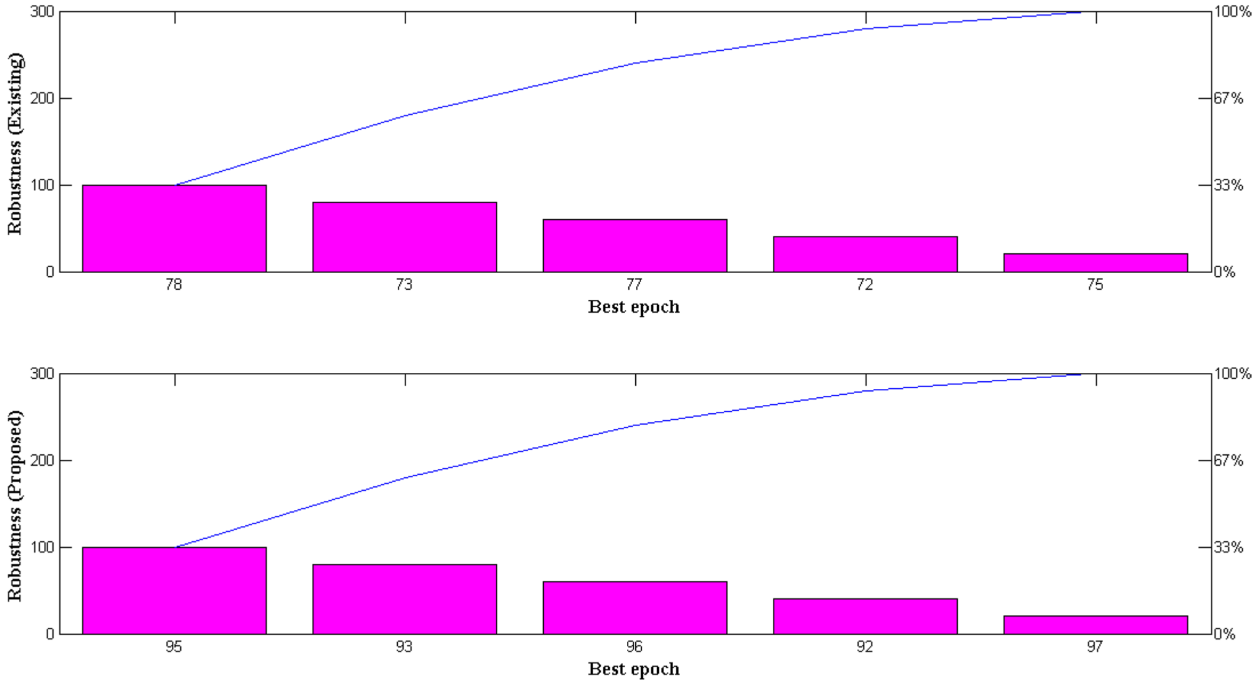


Fig. 8 Flexibility determinations

sizes of 20, in order to validate this property. As data size varies throughout this process, a random variation is seen. Hence, the strength of cloud computing will be lower if less data size is provided. This can be shown using the best epoch values of 60 and 80, where the new method's equivalent robustness is 96 and 93 and the conventional method's is, respectively, 77 and 73.

Conclusion

An efficient simulation model is used to demonstrate how mobile cloud computing might be used to handle various emergency situations. Utilizing energy development variables that are operated with various energy kinds, a new mathematical model is created for both operation and analysis. The analytical representations are used to resolve the SNR values of mobile cloud computing, which are offered as a disadvantage in emergency situations. Using a loop generation technique, the activation function with gain and assigned bandwidth requirements are studied. For selecting the best networked cloud computing nodes, created loops are introduced directly in the suggested method with the aid of the MATLAB toolbox. A theoretical framework and a limited number of implementation categories were created in the analyzed literature, leaving out any mention of steady state mobile cloud computing systems. In order to overcome the aforementioned issue, the projected technique employs a decision tree optimization algorithm with a number of subset features. Since a decision tree method is included, indulgent network information totally removes the impure cloud computing nodes from the network. Additionally, the decision tree's index terms are constructed using entropy values, as a result, the computing model only works when it gets zero decibel points, which shows that all noise has been eliminated from the system. Five different situations are used to test the analytical framework, and comparisons with other approaches are also performed. The comparison case's results demonstrate that the proposed method which uses a decision tree, performs better than the standard procedure. Future applications of the suggested analytical framework for large-scale data processing networks can be made without the need for manual representation adjustments, and the same features can be integrated with extremely efficient optimization algorithms.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No.61862051), UTM Research Fellow (No. 00P27), the Science and Technology Foundation of Guizhou Province (No.ZK[2022]549, No.[2019]1299), the Top-notch Talent Program of of Guizhou province (No.KY[2018]080), the Natural Science Foundation of Education of Guizhou province(No.[2019]203) and the Funds of Qiannan Normal University for Nationalities (No. qnsy2018003, No. qnsy2019rc09, No. qnsy2018JS013, No. qnsyrc201715).

Authors' contributions

Conceptualization by Tao Hai, Dan Wang; Methodology by Dayang Jawawi and Haripasath Manoharan; Software by Ye Lu and Jincheng Zhou; formal analysis by Dawang Jawani and Ebuka Ibeke investigation by Tao Hai and Dan Wang; Resources and data collection by Jincheng Zhou, Shitharth Selvarajan; Writing by: Hariprasath Manoharan, Dan Wang, Ye Lu and Tao Hai; Validation by: Ebuka Ibeke and Jincheng Zhou; Funding Acquisition by Shitharth Selvarajan. The author(s) read and approved the final manuscript.

Funding
The project was supported by the This work was supported by the National Natural Science Foundation of China (No.61862051), the Science and Technology Foundation of Guizhou Province (No.ZK[2022]549, No.[2019]1299), the Top-notch Talent Program of of Guizhou province (No.KY[2018]080), UTM Research Fellow (No. 00P27).

Availability of data and materials
The supporting data can be provided on request.

Declarations

Ethics approval and consent to participate
The research has consent for Ethical Approval and Consent to participate.

Consent for publication
Consent has been granted by all authors and there is no conflict.

Competing interests
The authors declare no competing interests.

Received: 6 September 2022 Accepted: 29 September 2022

References

1. Garetto M, Leonardi E (2007) Analysis of random mobility models with partial differential equations. IEEE Trans Mob Comput 6:1204–1217. https://doi.org/10.1109/TMC.2007.1023
2. Rathee G, Garg S, Kaddoum G, et al (2020) A trusted social network using hypothetical mathematical model and decision-based scheme. IEEE Access 4223–4232. https://doi.org/10.1109/ACCESS.2020.3048077
3. Hatami-Marbini A, Varzgani N, Sajadi SM, Kamali A (2022) An emergency medical services system design using mathematical modeling and simulation-based optimization approaches. Decis Anal J 3:100059. https://doi.org/10.1016/j.dajour.2022.100059
4. Bruno R, Conti M, Gregori E (2002) Optimization of efficiency and energy consumption in p-persistent CSMA-based wireless LANs. IEEE Trans Mob Comput 1:10–31. https://doi.org/10.1109/TMC.2002.1011056
5. Ramasamy V, Gomathy B, Sarkar JL et al (2020) EMC2: An emergency management system using mobile cloud computing. IET Networks 9:64–73. https://doi.org/10.1049/iet-net.2019.0014
6. Kim C, Dudin A, Dudin S, Dudina O (2021) Mathematical model of operation of a cell of a mobile communication network with adaptive modulation schemes and handover of mobile users. IEEE Access 9:106933–106946. https://doi.org/10.1109/ACCESS.2021.3100561
7. Lamb ZW, Agrawal DP (2019) Analysis of mobile edge computing for vehicular networks. Sensors, Switzerland, p 19. https://doi.org/10.3390/s19061303
8. Dou Y, Ho YH, Deng Y, Chan HCB (2021) Mobile intercloud system for edge cloud computing. Wirel Commun Mob Comput 2021. https://doi.org/10.1155/2021/5549470
9. Gandotra P, Jha RK (2016) Device-to-device communication in cellular networks: a survey. J Netw Comput Appl 71:99–117
10. Mitropoulos S, Mitsis C, Valacheas P, Douligeris C (2021) An online emergency medical management information system using mobile computing. Appl Comput Informatics. https://doi.org/10.1108/aci-03-2021-0069
11. Chennam KK, Aluvalu R, Shitharth S (2021) An Authentication Model with High Security for Cloud Database', Architectural Wireless Networks Solutions and Security Issues, Lecture notes in network and systems. Springer. 196(1):13–26. https://doi.org/10.1007/978-981-16-0386-0_2
12. Han S, Ma D, Kang C, et al (2022) Optimization of Mobile Edge Computing Offloading Model for Distributed Wireless Sensor Devices. J Sensors 2022. https://doi.org/10.1155/2022/9047737
13. Nanda S, Panigrahi CR, Pati B (2020) Emergency management systems using mobile cloud computing: a survey. Int J Commun Syst 1–20. https://doi.org/10.1002/dac.4619
14. Poulymenopoulou M, Malamateniou F, Vassilacopoulos G (2012) Emergency healthcare process automation using mobile computing and cloud services. J Med Syst 36:3233–3241. https://doi.org/10.1007/s10916-011-9814-y
15. Aluvalu R, Uma Maheswari V, Chennam KK, Shitharth S (2021) Data security in cloud computing using abe-based access control, architectural wireless networks solutions and security issues, lecture notes in network and systems. Springer 196(1):47–62. https://doi.org/10.1007/978-981-16-0386-0_4
16. Devi BT, Shitharth S (2020) An Appraisal over Intrusion Detection systems in cloud computing security attacks, 2nd International Conference on Innovative Mechanisms for Industry applications, IEEE Explore, ICIMIA. p 122. https://doi.org/10.1109/ICIMIA48430.2020.9074924
17. Shabbir M, Shabbir A, Iwendi C, Javed AR, Rizwan M, Herencsar N, Lin JC (2021) Enhancing security of health information using modular encryption standard in mobile cloud computing. IEEE Access 1–1. https://doi.org/10.1109/ACCESS.2021.3049564
18. Anajemba J, Yue T, Iwendi C, Alenezi M, Mittal Mohit (2020). Optimal cooperative offloading scheme for energy efficient multi-access edge computation. IEEE Access 1–1. https://doi.org/10.1109/ACCESS.2020.2980196
19. Sirajuddin M, Rupa C, Iwendi C, Biamba C (2021) TBSMR: a Trust-Based Secure Multipath Routing Protocol for Enhancing the QoS of the Mobile Ad Hoc Network. Sec Commun Netw. https://doi.org/10.1155/2021/5521713
20. Priya S, Bhattacharya S, Reddy P, Somayaji S, Lakshman K, Kaluri R, Hussien A, Gadekallu T (2020) Load balancing of energy cloud using wind driven and firefly algorithms in internet of everything. J Parallel Distrib Comput 142. https://doi.org/10.1016/j.jpdc.2020.02.010
21. Wang T, Quan Y, Shen X, Gadekallu T, Wang W, Dev K (2021) A privacy-enhanced retrieval technology for the cloud-assisted internet of things. IEEE Trans Ind Inform 1–1. https://doi.org/10.1109/TII.2021.3103547
22. Gadekallu T, Pham QV, Nguyen DC, Maddikunta P, Deepa N, Prabadevi B, Pathirana P, Zhao J, Hwang W (2021) Blockchain for edge of things: applications, opportunities, and challenges. IEEE Internet Things J 9:964. https://doi.org/10.1109/jiot.2021.3119639
23. Muralidharan C, Mohamed Sirajudeen Y, Shitharth S, Alhebaishi N, Mosli RH, Alhelou HH (2022) Three-phase service level agreements and trust management model for monitoring and managing the services by trusted cloud broker, IET Commun. pp 1–12. https://doi.org/10.1049/cmu2.12484
24. Bhageerath Chakravorthy G, Aditya Vardhan R, Karthik Shetty K, Mahesh K, Shitharth S (2021) Handling tactful data in cloud using Pkg encryption technique' 4th smart city symposium. pp 338–43. https://doi.org/10.1049/icp.2022.0366
25. Ding Z, Yang R (2018) Modeling and analysis for mobile computing systems based on petri nets: a survey. IEEE Access 6:68038–68056. https://doi.org/10.1109/ACCESS.2018.2878807
26. Bashiri M, Rezanezhad M, Tavakkoli-Moghaddam R, Hasanzadeh H (2018) Mathematical modeling for a p-mobile hub location problem in a dynamic environment by a genetic algorithm. Appl Math Model 54:151–169. https://doi.org/10.1016/j.apm.2017.09.032

Publisher's Note