

# Emerging Trends and Challenges in Mobile Edge Computing and Cloud-Aware Mobile Fog Computing

Lead Guest Editor: Muhammad Shiraz

Guest Editors: Suleman Khan, Saba Bashir, and Rashid Khokhar





---

# **Emerging Trends and Challenges in Mobile Edge Computing and Cloud-Aware Mobile Fog Computing**

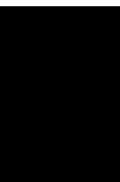
Wireless Communications and Mobile Computing

---

# **Emerging Trends and Challenges in Mobile Edge Computing and Cloud- Aware Mobile Fog Computing**

Lead Guest Editor: Muhammad Shiraz

Guest Editors: Suleman Khan, Saba Bashir, and  
Rashid Khokhar



---



Copyright © 2022 Hindawi Limited. All rights reserved.

This is a special issue published in “Wireless Communications and Mobile Computing.” All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

# Chief Editor

































Zhipeng Cai , USA

## Associate Editors

Ke Guan , China  
Jaime Lloret , Spain  
Maode Ma , Singapore

## Academic Editors

Muhammad Inam Abbasi, Malaysia  
Ghufran Ahmed , Pakistan  
Hamza Mohammed Ridha Al-Khafaji ,  
Iraq  
Abdullah Alamoodi , Malaysia  
Marica Amadeo, Italy  
Sandhya Aneja, USA  
Mohd Dilshad Ansari, India  
Eva Antonino-Daviu , Spain  
Mehmet Emin Aydin, United Kingdom  
Parameshchhari B. D. , India  
Kalapraveen Bagadi , India  
Ashish Bagwari , India  
Dr. Abdul Basit , Pakistan  
Alessandro Bazzi , Italy  
Zdenek Becvar , Czech Republic  
Nabil Benamar , Morocco  
Olivier Berder, France  
Petros S. Bithas, Greece  
Dario Bruneo , Italy  
Jun Cai, Canada  
Xuesong Cai, Denmark  
Gerardo Canfora , Italy  
Rolando Carrasco, United Kingdom  
Vicente Casares-Giner , Spain  
Brijesh Chaurasia, India  
Lin Chen , France  
Xianfu Chen , Finland  
Hui Cheng , United Kingdom  
Hsin-Hung Cho, Taiwan  
Ernestina Cianca , Italy  
Marta Cimitile , Italy  
Riccardo Colella , Italy  
Mario Collotta , Italy  
Massimo Condoluci , Sweden  
Antonino Crivello , Italy  
Antonio De Domenico , France  
Floriano De Rango , Italy




Antonio De la Oliva , Spain  
Margot Deruyck, Belgium  
Liang Dong , USA  
Praveen Kumar Donta, Austria  
Zhuojun Duan, USA  
Mohammed El-Hajjar , United Kingdom  
Oscar Esparza , Spain  
Maria Fazio , Italy  
Mauro Femminella , Italy  
Manuel Fernandez-Veiga , Spain  
Gianluigi Ferrari , Italy  
Luca Foschini , Italy  
Alexandros G. Fragkiadakis , Greece  
Ivan Ganchev , Bulgaria  
Óscar García, Spain  
Manuel García Sánchez , Spain  
L. J. García Villalba , Spain  
Miguel Garcia-Pineda , Spain  
Piedad Garrido , Spain  
Michele Girolami, Italy  
Mariusz Glabowski , Poland  
Carles Gomez , Spain  
Antonio Guerrieri , Italy  
Barbara Guidi , Italy  
Rami Hamdi, Qatar  
Tao Han, USA  
Sherief Hashima , Egypt  
Mahmoud Hassaballah , Egypt  
Yejun He , China  
Yixin He, China  
Andrej Hrovat , Slovenia  
Chunqiang Hu , China  
Xuexian Hu , China  
Zhenghua Huang , China  
Xiaohong Jiang , Japan  
Vicente Julian , Spain  
Rajesh Kaluri , India  
Dimitrios Katsaros, Greece  
Muhammad Asghar Khan, Pakistan  
Rahim Khan , Pakistan  
Ahmed Khattab, Egypt  
Hasan Ali Khattak, Pakistan  
Mario Kolberg , United Kingdom  
Meet Kumari, India  
Wen-Cheng Lai , Taiwan

Jose M. Lanza-Gutierrez, Spain  
Pavlos I. Lazaridis , United Kingdom  
Kim-Hung Le , Vietnam  
Tuan Anh Le , United Kingdom  
Xianfu Lei, China  
Jianfeng Li , China  
Xiangxue Li , China  
Yaguang Lin , China  
Zhi Lin , China  
Liu Liu , China  
Mingqian Liu , China  
Zhi Liu, Japan  
Miguel López-Benítez , United Kingdom  
Chuanwen Luo , China  
Lu Lv, China  
Basem M. ElHalawany , Egypt  
Imadeldin Mahgoub , USA  
Rajesh Manoharan , India  
Davide Mattera , Italy  
Michael McGuire , Canada  
Weizhi Meng , Denmark  
Klaus Moessner , United Kingdom  
Simone Morosi , Italy  
Amrit Mukherjee, Czech Republic  
Shahid Mumtaz , Portugal  
Giovanni Nardini , Italy  
Tuan M. Nguyen , Vietnam  
Petros Nicolitidis , Greece  
Rajendran Parthiban , Malaysia  
Giovanni Pau , Italy  
Matteo Petracca , Italy  
Marco Picone , Italy  
Daniele Pinchera , Italy  
Giuseppe Piro , Italy  
Javier Prieto , Spain  
Umair Rafique, Finland  
Maheswar Rajagopal , India  
Sujan Rajbhandari , United Kingdom  
Rajib Rana, Australia  
Luca Reggiani , Italy  
Daniel G. Reina , Spain  
Bo Rong , Canada  
Mangal Sain , Republic of Korea  
Praneet Saurabh , India

Hans Schotten, Germany  
Patrick Seeling , USA  
Muhammad Shafiq , China  
Zaffar Ahmed Shaikh , Pakistan  
Vishal Sharma , United Kingdom  
Kaize Shi , Australia  
Chakchai So-In, Thailand  
Enrique Stevens-Navarro , Mexico  
Sangeetha Subbaraj , India  
Tien-Wen Sung, Taiwan  
Suhua Tang , Japan  
Pan Tang , China  
Pierre-Martin Tardif , Canada  
Sreenath Reddy Thummaluru, India  
Tran Trung Duy , Vietnam  
Fan-Hsun Tseng, Taiwan  
S Velliangiri , India  
Quoc-Tuan Vien , United Kingdom  
Enrico M. Vitucci , Italy  
Shaohua Wan , China  
Dawei Wang, China  
Huaqun Wang , China  
Pengfei Wang , China  
Dapeng Wu , China  
Huaming Wu , China  
Ding Xu , China  
YAN YAO , China  
Jie Yang, USA  
Long Yang , China  
Qiang Ye , Canada  
Changyan Yi , China  
Ya-Ju Yu , Taiwan  
Marat V. Yuldashev , Finland  
Sherali Zeadally, USA  
Hong-Hai Zhang, USA  
Jiliang Zhang, China  
Lei Zhang, Spain  
Wence Zhang , China  
Yushu Zhang, China  
Kechen Zheng, China  
Fuhui Zhou , USA  
Meiling Zhu, United Kingdom  
Zhengyu Zhu , China

# Contents

## **Load Balancing in Edge Computing Using Integer Linear Programming Based Genetic Algorithm and Multilevel Control Approach**

Rui Zhang , Hong Shu , and Yahya Dorostkar Navaei 


Research Article (22 pages), Article ID 6125246, Volume 2022 (2022)

## **A Novel Link-Network Assignment to Improve the Performance of Mobility Management Protocols in Future Mobile Networks**

Jesús Calle-Cancho , Javier Carmona-Murillo , José-Luis González-Sánchez , and David Cortés-Polo 





Research Article (13 pages), Article ID 7061588, Volume 2022 (2022)

## **Task Offloading and Scheduling Strategy for Intelligent Prosthesis in Mobile Edge Computing Environment**

Ping Qi 




Research Article (13 pages), Article ID 2890473, Volume 2022 (2022)

## **Joint Load Balancing and Offloading Optimization in Multiple Parked Vehicle-Assisted Edge Computing**

Xinyue Hu , Xiaoke Tang , Yantao Yu , Sihai Qiu , and Shiyong Chen 


Research Article (13 pages), Article ID 8943862, Volume 2021 (2021)

## **NOMA and OMA-Based Massive MIMO and Clustering Algorithms for Beyond 5G IoT Networks**

Taj Rahman , Feroz Khan , Inayat Khan , Niamat Ullah , Maha M. Althobaiti , and Fawaz Alassery 

Research Article (12 pages), Article ID 6522089, Volume 2021 (2021)

## **Optimization Strategy of Task Offloading with Wireless and Computing Resource Management in Mobile Edge Computing**

Xintao Wu , Jie Gan , Shiyong Chen , Xu Zhao , and Yucheng Wu 



Research Article (11 pages), Article ID 8288836, Volume 2021 (2021)

## **Data Integrity Time Optimization of a Blockchain IoT Smart Home Network Using Different Consensus and Hash Algorithms**

Ammar Riadh Kairaldein , Nor Fadzilah Abdullah , Asma Abu-Samah , and Rosdiadee Nordin 

Research Article (23 pages), Article ID 4401809, Volume 2021 (2021)

## **A Review of Big Data Resource Management: Using Smart Grid Systems as a Case Study**

Muhammad Fawad Khan, Muhammad Azam, Muhammad Asghar Khan , Fahad Algarni , Mujaddad Ashfaq, Ibtihaj Ahmad, and Insaf Ullah

Review Article (18 pages), Article ID 3740476, Volume 2021 (2021)

## Research Article

# Load Balancing in Edge Computing Using Integer Linear Programming Based Genetic Algorithm and Multilevel Control Approach

Rui Zhang <sup>1</sup>, Hong Shu <sup>1</sup>, and Yahya Dorostkar Navaei <sup>2</sup>

<sup>1</sup>Mathematics and Information Science College of Guiyang University, Guiyang 550005, China

<sup>2</sup>Department of Computer and Technology Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran

Correspondence should be addressed to Hong Shu; yrj999999@sina.com and Yahya Dorostkar Navaei; y.dorostkar@qiau.ac.ir

Received 19 December 2021; Revised 22 April 2022; Accepted 16 June 2022; Published 30 June 2022

Academic Editor: Saba Bashir

Copyright © 2022 Rui Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Due to the proliferation of requests in heterogeneous resources in edge computing, the existence of a large number of tasks and workloads in virtual machines in the edge computing environment is inevitable. Thus, load balancing strives to facilitate an even distribution of workload across available resources. Its purpose is to provide continuous service and to ensure fair load distribution among resources. Load balancing, with the aim of minimizing response time for tasks and improving resource efficiency, tries to do the proper mapping of tasks among virtual machines at a lower cost. Flow scheduling, on the other hand, assigns a task (group of tasks) to computational resources by prioritizing tasks, so that the relationship between them is maintained. Therefore, in this research, a hierarchical control framework for load balancing and assignment of tasks in edge computing services is presented in order to create load balancing. In the proposed method, in the first level, the genetic algorithm receives a set of tasks in a workflow. Genetic algorithm prioritizes and assigns tasks to resources according to time constraints, resource processing power, resource availability, and task cost. For this purpose, the integer linear programming optimization in the evaluation function of the genetic algorithm will be utilized. In the second level, considering the past load distribution in edge resources, we estimate the probability of load distribution among sources according to hidden Markov model (HMM). Finally, in order to optimally map tasks to the virtual machines in each host, we will use game theory with service quality factors as an evaluation function. Previous methods have provided a hierarchical control framework that aims to achieve conflicting goals within a data center, but does not use linear programming. Considering the use of service quality criteria as evaluation function parameters in heuristic and optimization methods in this research, it is expected that the results of this research will improve compared to previous methods.

## 1. Introduction

An edge computing model is possibly the most efficient model if its resources are used efficiently, and this can be achieved by applying and maintaining proper management of edge resources [1]. Resource management is achieved by adopting strong resource scheduling, efficient allocations, and powerful scalability techniques. These resources are provided to customers through a process called virtualization of a software component, hardware, or both, in the form of virtual machines (VMs) [2]. The biggest advantage of edge computing is that a physical machine becomes a merely multipurpose virtual machine for a user [3–5]. The cloud

service provider (CSP) plays an important role in providing services to users, and considering the availability of virtual resources, assigning a task becomes really complex. While submitting user requests, some VMs experience heavy traffic on user tasks, and some of them experience less traffic. As a result, the edge service provider has to deal with unbalanced machines with large differences in user tasks and resource usage [6–9]. The problem with load imbalance is an adverse event on the CSP side that degrades the performance and efficiency of computing resources along with the quality of service (QoS) assurance in the service level agreement (SLA) between the consumer and the edge service provider. In this situation, there is a need for load balancing that has



become a strange yet interesting topic among researchers. Load balancing in edge computing can be used at the physical device level as well as the VM level [10–13]. Load balancing is a process of redistributing workload in a distributed system such as edge computing, which makes sure that no virtual machine is overloaded, while other virtual machines are idle or have less workload. Load balancing tries to accelerate various limiting parameters, such as response time, runtime, and system stability, in order to improve edge performance. This is an optimization method in which scheduling is an NP-hard problem. There are a number of load balancing approaches proposed by researchers, most of which focus on task scheduling, task allocation, resource planning, resource allocation, and resource management [6]. After distributing a balanced load among the hosts of the service provider in the edge environment, there is a need for proper mapping of tasks among virtual machines and scheduling of resources to input tasks in the current host according to different criteria. The process of workflow scheduling refers to the mapping of tasks (a group of tasks) to existing computational resources and the timing of their execution by observing the priorities among tasks so that the relationship between them is maintained. The structure of workflows is often defined as a graph without a circle, like a tree structure. Decisions to map tasks to a resource can be made based on the information contained in a scheduler according to the criteria of the service level agreement (SLA) and the QoS needs of the users [14–17].

Therefore, in order to overcome these challenges, a hierarchical control framework for load balancing and task allocation in edge computing services is presented in this study. In the proposed method, in the first level, a genetic algorithm (GA) has been employed to model the workload [18]. The input of the genetic algorithm in the proposed method includes a set of tasks. Depending on the priorities and the relationship between the tasks, we will model the load distribution among the existing hosts in order to optimize the evaluation criteria.

The evaluation criteria used in this method include time constraints, resource processing power, probability of access to the resource, and the cost of performing tasks in the resource. To optimize the load distribution among resources, due to the existing limitations, the integer linear programming optimization (ILP) in the evaluation function of the genetic algorithm will be used [19]. In the second level, considering the past load distribution in edge resources, we estimate the probability of load distribution among sources according to the hidden Markov model (HMM) [20]. Finally, in order to map the tasks to the virtual machines in each host, we will use the game theory with QoS factors as an evaluation function [21]. Considering the use of QoS criteria as evaluation function parameters in discovery and optimization methods in this research, it is expected that the results of this research will improve compared to previous methods.

The continuation of this article is as follows:

- (i) Prioritizing and assigning tasks to resources using integer linear programming based genetic algorithm by considering time constraints, resource processing power, resource availability, and task cost

- (ii) Estimating the probability of load distribution among sources according to hidden Markov model (HMM) by considering the past load distribution in edge resources
- (iii) Mapping tasks to the virtual machines in each host in optimally manner using game theory with quality of service factors as an evaluation function
- (iv) Evaluating makespan, cost, energy, performance, and reliability of proposed method

In the second part, related works will be reviewed. In the third section, the details of the proposed method will be explained. In the fourth section, the implementation and evaluation of the proposed method will be stated. In the fifth section, the conclusion of the article will be stated.

## 2. Related Works

With the increasing popularity and the advancement of edge computing technology, users are merely trying to access resources that are sufficient to perform the tasks they need and they are only willing to pay for the resources they need. In edge computing, tasks must be distributed in such a way that all available resources have approximately the same number of tasks to execute. One solution that can solve this problem effectively is to schedule tasks with control approaches in order to balance the load. Resource scheduling using a control approach for mapping tasks to virtual machines in the edge is one of the most important issues we face. This resource scheduling approach makes it possible to authorize the execution of a task in a virtual machine in the edge or to migrate a task from one virtual machine to another. The restrictions that the user has on performing tasks on virtual machines must be taken into account during the scheduling based on the desired service quality rules. For instance, tasks may have a specific execution sequence, or virtual machines may be assigned tasks exclusively, and only one task may be executed on the resource at a time, or a specific time limit may be set for the execution of tasks. The importance of load balancing has led to extensive research in this area [22–31]. The following are some of these studies, the main basis of which is timing.

In [32], a job allocation mechanism (JAM) has proposed to reduce battery consumption in the processing of large Internet-physical-social data in mobile edge computing (MCC). In this method battery consumption for processing work has displayed continuously with mobile devices, without an external edge server in a shared architecture MCC environment. In [33], the improved maximum minimum scheduling algorithm (IMMSA) that improves request completion time by using machine learning training as well as requesting size clustering and clustering the productivity percentage of virtual machines has been proposed. In [34], a genetic algorithm (GA)-based optimization technique in the sensor mobile edge computing environment to discern the optimal solution has been proposed. In [35], an improved elitism genetic algorithm (IEGA) for overcoming the task scheduling problem for FC to enhance the quality

of services to users of IoT devices has been suggested. In [36], the two-level load balancing approach in fog computing environment as a multiobjective optimization problem using the elitism-based genetic algorithm (EGA) for minimizing the service time, cost, and energy consumption and thus ensuring the QoS of IoT applications has been presented. In [37], a processing model for the load balancing problem using NSGAI algorithm has presented in which a trade-off between energy consumption and delay in processing workloads in fog has formulated. In [38], an incentive-based bargaining approach which encourages the fog nodes to cooperate among themselves by receiving incentives from the end users benefitting from the cooperation has been proposed. In [39], the hidden Markov chain learning method has been used to cope with this challenge in the IoT ecosystem integrated with the fog computing, to determine the probability of the need for each thing or resource in the near future with the aim of reducing latency and increasing the network use. In [40], a computational model as a game that considers energy consumption and transmission latency as decision parameters for task offloading of IoT applications has been proposed. In [41], an offload and migration-enabled smart gateway for Cloud of Things approach has proposed that employs noncooperative game theory to offload, schedule, and reschedule the computational tasks effectively in the fog-cloud environment.

### 3. Proposed Method

The proposed method aims to provide a hierarchical control framework for the optimal allocation of resources to tasks in order to balance the work load in edge computing center. In this method, the ultimate goal is to provide services that meet QoS needs only by using the necessary resources in the edge infrastructure. In the proposed method, in general, QoS requirements are determined based on resource efficiency, total time of tasks in the edge environment, and energy consumption of resources while performing tasks and cost of performing tasks on resources, which may be different according to task priorities. Therefore, allocating effective and efficient resources contributes to increase in productivity and reduction in completion time, energy, and costs. Edge computing is a solution recently adopted in order to provide services that host tasks in virtual environments. Physical resources are divided into several virtual machines, and these virtual machines are responsible for assigning tasks that work with parts of the capacity of the physical system. Automated management techniques are implemented by network controllers that can control the set of programs run by each server, the volume of requests on different servers, and the capacity allocated to run each program on each server. Therefore, in this research, a two-tier control architecture is presented that deals with the issue of load balance with optimal allocation of resources to tasks as well as controlling task acceptance according to QoS criteria. The general architecture of the proposed method is shown in Figure 1.

As shown in Figure 1, the proposed method provides a two-tier hierarchical control framework for load balancing

and task allocation in edge computing services. In the first level of the proposed method, genetic algorithm has been employed to model the workload [18]. The input of the genetic algorithm in the proposed method includes a set of tasks; then, according to the priorities and the relationship between the tasks, we will model the load distribution among the existing hosts in order to optimize the evaluation criteria. The evaluation criteria used in this method include time constraints, resource processing power, probability of access to the resource, and cost of performing tasks in the resource. Due to the existing limitations, to optimize the load distribution among sources, the correct linear programming optimization in the evaluation function of the genetic algorithm will be employed [19]. In the second level, according to the history of load distribution in edge resources, we estimate the probability of load distribution among sources using hidden Markov model [20]. Finally, in order to map the tasks to the virtual machines in each host, we will use the game theory with QoS factors as an evaluation function [21]. In the continuation of this chapter, we will describe the different parts of the proposed method in more detail.

*3.1. Genetic Algorithm Based on Integer Linear Programming Approach.* In the proposed method, a genetic algorithm based on the integer linear programming (ILP) approach with respect to the dynamic nature of the edge environment is presented. The proposed genetic algorithm looks for an optimal (or near-optimal) solution that starts from the initial population and is generated using the selection of the most appropriate chromosomes based on fitting and mutation operators. In addition to proper encoding for chromosomes as well as proper design for fitting and mutation operators, taking the limitations of the problem that should be consistent with the nature of the optimization problem into account, the proposed genetic algorithm is equipped with a mechanism that examines the feasibility of chromosomes and if the chromosome does not meet the expectations, there is a penalty and the value of the fitness function reduces in order to reduce its selection and survival chances in the next generation. The proposed genetic algorithm aims to look for a solution for scheduling new tasks entering the edge environment using the previous solution for the problem, which allows it to work according to the dynamic scheduling technique.

*3.1.1. Chromosome Encoding.* The proposed genetic algorithm finds an initial set of tasks as input and begins to generate the initial population of chromosomes, each of which represents a possible solution to the optimization problem. Thus, a chromosome is a vector for a gene whose numbers within each gene indicate the source number and possible resource allocation to tasks and a potential scheduling. Figure 2 shows an example of the chromosomes presented in the proposed method.

As shown in Figure 2, the chromosomes in the proposed method include vector genes, and the number of these genes is equal to the number of tasks in the edge environment. In the proposed method, considering the fact that the genetic algorithm uses the previous schedules to present new chromosomes, the number of received tasks and resources are

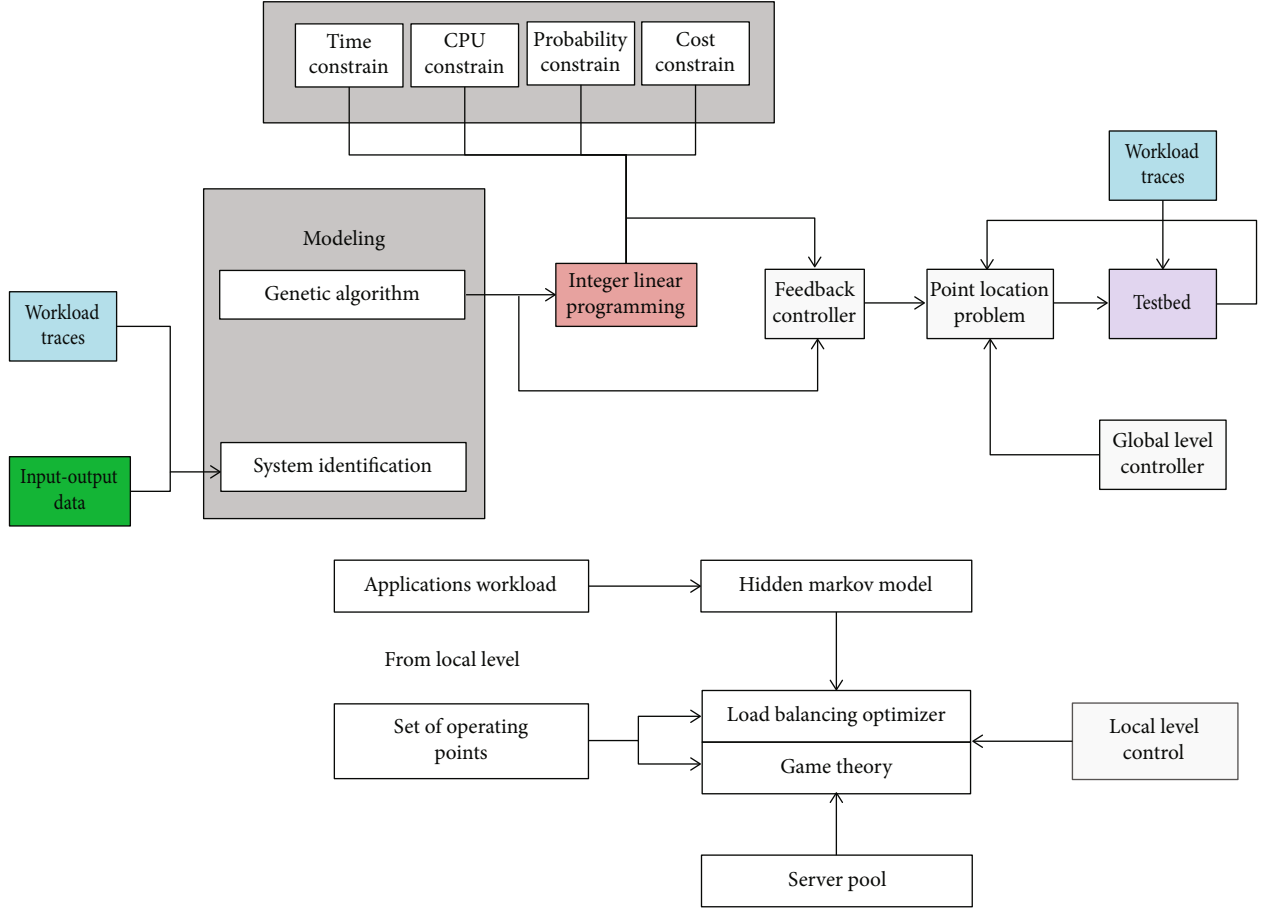


FIGURE 1: General architecture of the proposed method.

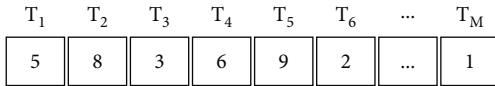


FIGURE 2: Representation of a chromosome in the proposed method.

the same so that in case of any problem for any of the resources, it will not violate error tolerance in the method. The numbers listed in each of the genes serve as a possible source for the task at hand. Initially, each chromosome is considered as a probabilistic scheduling that changes with the application of the fitness function and mutation operators, and finally, the chromosome with the highest amount of proportion is selected as the near-optimal scheduling.

**3.1.2. Fitness Function.** The fitness function for each chromosome is determined according to the objective function, which in the proposed method is a combination of four parameters: completion time, cost, resource efficiency, and probability of allocation. For each possible scheduling (chromosome) of the population, the fitness function is calculated while performing timed tasks on the chromosome at a particular time. The fit function represents the desired parameters such as:

- (i) the amount of time required to complete tasks in resources
- (ii) the cost required to perform tasks
- (iii) the resource efficiency rate
- (iv) the probability of assigning tasks to resources

These parameters are calculated in order not to accumulate charge in a source when performing tasks scheduled on the chromosome at a specific time. Table 1 illustrates the notation for the fit function in the proposed method.

The fitness function for calculating the given chromosomes is calculated based on Equation (1) using the objective function of the proposed model, which is optimized according to the integer linear programming.

$$\min \left( \sum_{i=1}^M \sum_{q=1}^Q X_E q P_i \left( C_{E_{R_i^N R_q^N}} + C_{E_{R_i^S R_q^S}} + C_{E_{R_i^C R_q^C}} \right) T_E + \sum_{j=1}^D \sum_{f=1}^F X_{Mig} q (1 - P_i) \left( C_{T_{R_j^B R_f^B}} + C_{D_{R_j^B R_f^B}} \right) T_T \right)$$

TABLE 1: Notations of the proposed model.

Notation	Explanation
$C_E$	Cost of each task performance
$C_T$	Cost of each task migration
$C_D$	Cost of data transfer
$T_E$	Task execution time
$T_T$	Transfer time and task migration
$T_I$	System idle time
$T_D$	Duty deadline
$X_E$	Task performance decision
$X_{Mig}$	Task migration decision
$D$	The amount of data transferred between virtual machines
$Q$	Request (task)
$F$	Requests for data transfer (migration)
$M$	Virtual machines
$P$	Probability of allocation
$R_N$	Cost according to the number of source cores
$R_S$	Cost according to processing speed
$R_C$	Cost according to memory capacity (cache)
$R_B$	Cost according to network bandwidth

Subject to

$$\begin{aligned}
& \sum_{q=1}^Q X_{E_q} \left( C_{E_{R_q^N}} \right) \leq C_{E_N} \\
& \sum_{q=1}^Q X_{E_q} \left( C_{E_{R_q^S}} \right) \leq C_{E_S} \\
& \sum_{q=1}^Q X_{E_q} \left( C_{E_{R_q^C}} \right) \leq C_{E_C} \\
& \sum_{q=1}^Q X_{Mig_q} \left( C_{E_{R_q^C}} \right) \leq C_{E_B} \\
& \sum_{q=1}^Q X_{Mig_q} \left( C_{D_{R_q^B}} \right) \leq C_{D_B} \\
& \sum_{i=1}^M T_{E_i} + T_{T_i} + T_{I_i} \leq T_{D_i} \\
& \sum_{q=1}^Q X_{E_q} \leq M \\
& \sum_{i=1}^F Q_i (1 - P_i) \leq D_i \\
& \sum_{q=1}^Q P_i = 1 \\
& X_{E_q}, X_{Mig_q} \in \{0, 1\}, q = 1, 2, \dots, Q
\end{aligned} \tag{1}$$

According to the fitness function presented in Equation (1), the appropriate chromosomes are selected from the original population, and the rest of the chromosomes are sent to the fitting and mutation operator in order to diversify the population and produce new superior chromosomes. Each chromosome in the new offspring population is examined to determine whether it is a possible solution to the problem; that is, it minimizes the fitness function and meets the given constraint demands. Impossible chromosomes that violate existing constraints are fined according to the value of the fitness function, so they are less likely to be chosen to reproduce and become new chromosomes. The most appropriate chromosome, which represents a near-optimal scheduling solution, is maintained after each replication step and then sorted by optimality. This process is repeated until the termination condition is met.

**3.1.3. Crossover Operator.** The crossover operator plays an important role for the genetic algorithm to diversify the population and produce new chromosomes. To increase the scope of the search and consequently obtain more possible public solutions, it is necessary for the genetic algorithm to perform the fitting process between two chromosomes (parents) and produce new offspring as a new population. The crossover operator is a random replacement of a number of genes on the first and second chromosomes. The parameter that is important in the crossover operator is called the fitting probability or P-crossover, and regarding the random fitting operator, it can be defined as follows:

$$P - \text{crossover} = \text{round}(k * (G_{\max} - G_{\min})), k \text{ is a rand in } (0, 1), \quad (2)$$

where the  $G_{\max}$  parameter is the maximum number of genes on the chromosome and the  $G_{\min}$  parameter is the minimum number of genes on the chromosome. The parameter  $k$  is considered as a random value in the range of zero and one. The value of the P-crossover parameter is considered as the part of the chromosome that must be exchanged between the first chromosome and the lower chromosome, and the initial and final intersection of this part is called the fitting point. The fitting point may first be selected from the beginning of the chromosome or from any other desired location on it, and the last fitting point is added to the value of the P-crossover parameter. Figure 2 shows the fitting operator.

As shown in Figure 3, during crossover operation, the genes present in P-crossover are swapped on two chromosomes. The part of the genes of the first chromosome that is in the P-crossover is transferred with the same number of genes in the second chromosome so that the new chromosomes have more variety than the previous chromosomes. Chromosome switching is end-to-end so that gene  $i$  from the first chromosome crossed with gene  $i$  from the second chromosome. Thus, in the new generation,  $n$  new chromosomes are produced which are added to the previous chromosomes, and the population after the crossover will be equal to  $2n$ . In this paper, each of the  $T_i$  genes represents a task, and  $R_i$  represents the value of the gene that indicates the source number for each of the solutions.

**3.1.4. Mutation Operator.** The mutation operator plays a key role in generating new populations and diversifying chromosomes and is as important as the fitting operator. This operator, like the mutation operator, is able to increase the scope of search as well as introducing possible solutions and producing new children as a new population. In this operator, the probability parameter is of great importance, which is considered as the point of the chromosome that must mutate. The P-mutate parameter or the probability of mutation can be calculated as follows:

$$P - \text{mutate} = \text{round}(k * (G_{\max} - G_{\min})), k \text{ is a rand in } [0, 1], \quad (3)$$

where the P-mutate parameter indicates the probability of mutation and the  $k$  parameter can be used as a value between zero and one, and even zero or one as the first and last gene on the chromosome. The difference between the fitting and the mutation operator is that the fitting operator replaces several genes on one chromosome with genes on another chromosome, but the mutation operator changes the value of one (or more) genes to produce a new chromosome. Figure 3 shows the mutation operator.

As shown in Figure 4, there are two chromosomes with different genes that change separately at the  $T_2$  and  $T_6$  locations. At  $T_2$ , the value of gene has changed from  $R_{12}$  to  $R_{15}$ . Similarly, the value of gene has mutated from  $R_{16}$  to  $R_{13}$  at  $T_6$ . Gene mutations may have good results. Occasionally it

can have bad results. Nevertheless, gene mutations are essential for maintaining population diversity.

**3.1.5. Selection Operator.** After the fitting and mutation operators perform their tasks, among the new population and the chromosomes produced as the next generation, the selection operator selects the chromosomes that have the highest proportionality function or, in other words, have a near-optimal solution to the scheduling problem in order to balance the load in the edge. In this case, the tasks assigned to virtual machines are examined with the maximum total execution time, the cost of performing tasks in resources and resource efficiency, and the possibility of optimal allocation of tasks to resources, in order to balance the load in the edge environment, and in case tasks in the optimal solution need to be migrated, they are transferred from one virtual machine to another considering optimal evaluation criteria. Therefore, optimizing the performance of the task scheduling algorithm in order to balance the load in the edge environment and transferring some tasks to another virtual machine will also help balance the workload.

**3.2. Markov Hidden Model.** As mentioned, in the proposed method, in order to create a load balance, examining the tasks that have already been assigned to each resource and calculating the probability of assigning tasks to the resources have been determined as one of the evaluation parameters for the fit function in the proposed linear programming model. Resources that have already received more tasks have more potential to upset the balance of load distribution in the edge environment. Therefore, identifying such resources can greatly help to balance the load in the edge environment. In this method, in order to determine the probability of resource distribution, a possible hidden Markov model is used which is described below.

To clarify the general concept of the Markov model process, it can be said that if we divide the time in the Markov model into three periods (past, present, and future), the future time of the system depends not only on its present state but also on this model and its process of determining the probability of the system on the path it has taken in the past. In other words, if the state of the system is known at moments such as  $t_1, t_2, \dots, t_n$ , it can be said that to predict the state of the system in the next moment,  $t_n + 1$ , the state of the system from moment  $t_1$  to moment  $t_n$  must be examined [42].

Markov hidden models can be defined as probabilistic models in which a sequence of probabilities is created by two random processes:

The process of moving between states and the process of propagating an output sequence are characterized by Markov property and output independence. The first model is a Markov model created by a finite set of states, which creates a sequence of states of variables and is provided by the initial state probabilities and the probability of transition between state variables. The second model is characterized by the release of a character from the alphabet specified in each mode, with a probability distribution that depends only on the mode. Transfer sequence mode is a hidden process. This means that variable modes cannot be viewed directly but can

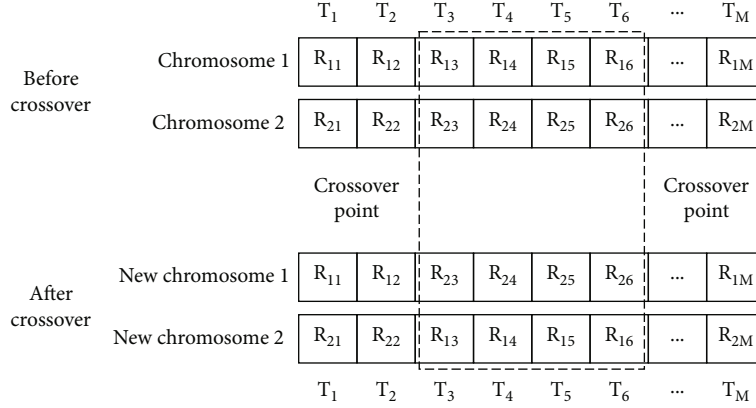


FIGURE 3: An example of a fitting operator on two chromosomes.

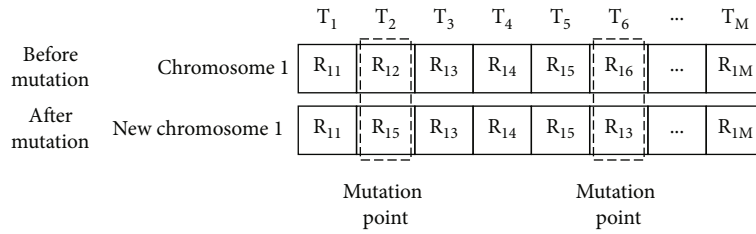


FIGURE 4: An example of population diversity on chromosomes.

be seen through a sequence of published symbols and that is the reason it is named Markov hidden model. Thus, a hidden Markov model is defined by different states, state probabilities, transition probability between states, propagation probabilities, and initial probabilities, and all these make up the architecture of Markov hidden models. The formal definition of hidden Markov models for the proposed method is based on specified pairs  $(S, V, p, A, B)$  with the following elements:

- (i)  $S = \{S_1, S_2, \dots, S_N\}$  is a set of states, where  $N$  is the number of states. The triple sequence pairs  $(S, p, A)$  represent the Markov chain in which the states are hidden and we never see them directly. In the proposed method, the virtual machines inside the hosts are considered as states in the hidden Markov model where the status of each virtual machine is considered hidden, and communication with the hosts containing the virtual machines is established. There may be several virtual machines in each host that provide services independently, but they are generally part of one host, and we can directly observe host states
- (ii)  $V = \{v_1, v_2, \dots, v_{V_M}\}$  are words or a set of symbols that may be published. In the proposed method, tasks are set of symbols that may be propagated and transferred between states. Transferring and migrating tasks between modes are considered as possibilities
- (iii)  $\pi : S \rightarrow [0, 1] = \{\pi_1, \pi_2, \dots, \pi_N\}$  is the initial probability distribution in the states. This indicates the probability of beginning the process in a mode. In the proposed method, for each resource, this proba-

bility is shown as the number of tasks in the queue for one resource compared to the total number of tasks in the edge environment and is considered as the initial probability distribution in the proposed method. It is therefore expected that:

$$\sum_{s \in S} \pi(s) = \sum_{i=1}^N \pi_i = 1. \quad (4)$$

- (iv)  $A = (a_{ij})_{i \in S, j \in S}$  is the probability of transfer and motion between the  $S_i$  and  $S_j$  modes. It is expected that for each  $S_i$  and  $S_j$ ,  $a_{ij} \in [0, 1]$  and for each  $S_j$ ,  $\sum_{i \in S} a_{ij} = 1$ .

In the proposed method, the migration of tasks between two sources  $i$  and  $j$  is shown with  $a_{ij}$ .

- (v)  $B = (b_{ij})_{i \in V, j \in S}$  is the probability of propagation if the symbol  $v_i$  is seen in the state  $S_j$ . In the proposed method, the presence of task  $i$  in source  $j$  is denoted by  $b_{ij}$

Markov hidden models are of great help if you need to model a process in which there is no direct knowledge of the state of the system. The main idea is that Markov hidden model is a “productive” sequel. In general, in this study, we talk about assigning tasks to observable resources since we

can use the hidden Markov model as a generating model that could be used to generate observational sequences. Algorithmically, a sequence of assigning tasks to resources  $O = o_1, \dots, o_T$ , with  $o_t \in V$  can be generated by the hidden Markov model. Two assumptions are made by the model. The first assumption is called Markov and indicates the model memory. It means that the current state depends only on the previous state, and therefore, we have:

$$P(q_t | q_1^{t-1}) = P(q_t | q_{t-1}). \quad (5)$$

The second assumption is the independence of assigning tasks to resources; i.e., the observation of output at time  $t$  depends only on the current state and is independent of previous observations, and we have:

$$P(o_t | o_1^{t-1}, q_1^t) = P(o_t | q_t). \quad (6)$$

The Markov property of a process can also be expressed in mathematical language. Consider a set of random variables  $[X(S), S \geq 0]$  and a set of system states at moments  $t_1, t_2, \dots, t_n$ . If  $X(t)$  follows the Markov process, for all values  $x_1, x_2, \dots, x_n$ , the following relation holds:

$$X(t_n + 1) = \begin{cases} \frac{x_1 + \dots + x_k}{x_n}, & 1 < k \leq n | F_1, \dots, F_k \subseteq t_1 \\ \frac{x_1 + \dots + x_k}{x_n}, & 1 < k \leq n | F_1, \dots, F_k \subseteq t_2, \\ \frac{x_1 + \dots + x_k}{x_n}, & 1 < k \leq n | F_1, \dots, F_k \subseteq t_n \end{cases}$$

$$\begin{aligned} P[X(t_n + 1) \leq x | X(t_n) = x_n, \dots, X(t_2) = x_2, X(t_1) = x_1] \\ = P[X(t_n + a) \leq x | (t_n) = x_n]. \end{aligned} \quad (7)$$

According to this relation, it can be said that the past states of the system can play a part in determining the next state of the system. In Markov Models, the states of system are denoted by  $t$  that can be continuous or discrete. The fact that  $t$  is discrete and can be interpreted in this way:

The behavior of the system is studied only at certain points in time. If  $t$  is discrete,  $X(t)$  is replaced by random variables in the form of  $X_1, X_2, \dots$ , and  $X_n$ . The set of values that  $X(t)$  can choose, by definition, is called the system state. System mode can also be discrete or continuous. Given that the requests sent to the edge environment might be related to each other, the first assumption based on nonindependence is true in the present case, and the past state of the system is examined to determine the possibility of assigning tasks to resources. In the proposed method, the value of the Markov hidden process at  $t + 1$  can be considered as the probability of assigning a new task to a resource with respect to the assignment of previous tasks to that resource. In this case, the system is the resources to which tasks are assigned, and then, the system state changes to a stable state. In the proposed method, according to the assignment of previous tasks in the edge environment, the possibility of assigning tasks in a limited time can be considered for each

resource. Given that our tasks are removed from the edge environment by running on resources, the number of these tasks in the resource queue is reduced, and then, the possibility of allocating resources at a particular time should be considered. This probability should be updated sequentially with the arrival of the new task, and the new tasks should be assigned to the source with the highest probability.

**3.3. Game Theory.** As illustrated in Figure 1, the proposed method uses game theory as a strategy in order to assign tasks to resources. In the previous steps, a near-optimal solution based on a genetic algorithm using a fit function based on integer linear programming was proposed. In this method, the proportionality function considers the constraints related to cost, execution time, and resource efficiency. Output solutions of genetic algorithms based on time, cost, and resource efficiency optimization are presented. However, in these solutions, the status of assigning previous tasks to resources and controlling the load balance between resources has not been considered. Therefore, in the proposed method, the hidden Markov model is used to investigate the distribution of tasks among resources according to the distribution of previous tasks. According to this model, based on the current status of the resource and the status of the assignment of previous tasks, allocation probability is assigned to each resource. Therefore, tasks that are less likely to be distributed should be removed from the generated solutions and then replaced with another solution that is out of the genetic algorithm based on its ranking. Furthermore, the proposed method uses game theory considering that the edge environment is a dynamic multifactor environment and tasks compete to get access to resources, in order to dynamically allocate and control the load balance and service quality parameters. Game theory is responsible for step-by-step review and control of the overall state of the edge system with respect to assigning tasks to resources according to the proposed solution.

Game theory is a mathematical approach for decision making that analyzes competitive situations to determine the optimal actions. In recent years, game theory has been widely used to deal with various problems related to managerial optimization. In particular, the issue of multifactor scheduling with game theory has attracted considerable attention from researchers [43]. In a work using the game-based two-layer scheduling method, with the aim of reducing the completion time of tasks in resources, balancing the total load of machines and energy consumption to achieve real-time data-based optimization for the edge development environment is of great importance [44]. In another work, in order to solve the problem of scheduling edge services by several factors, a model is proposed based on game theory to coordinate the relationships between diverse parameters, according to different strategies of service providers [45].

**3.3.1. Game-Based Scheduling Formulation.** In the proposed method, the scheduling problem can be considered as a non-cooperative game between  $N$  players with complete information in which each machine as a player decides on the next

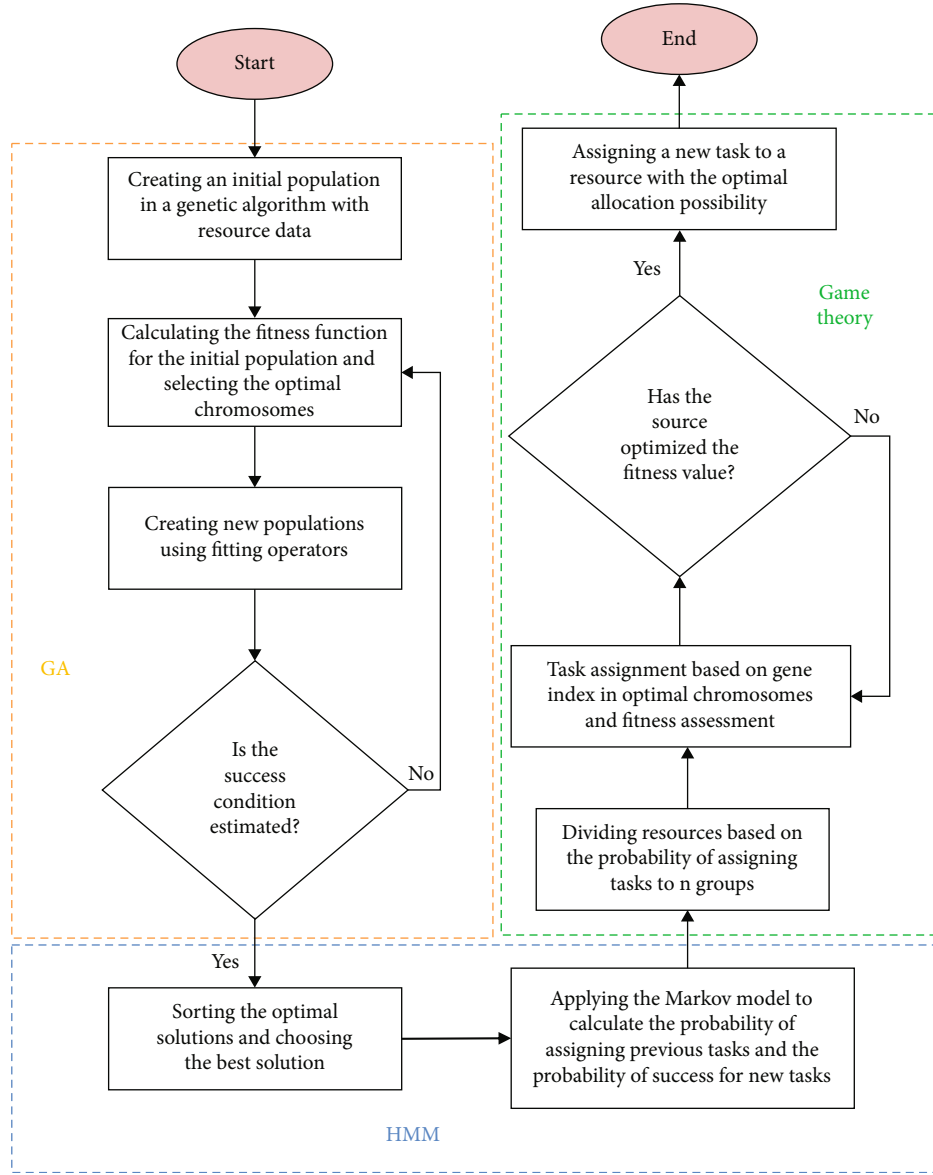


FIGURE 5: Flowchart of the proposed method.

move (performing task  $k$ ). And game strategies (i.e., selecting an appropriate method to process tasks) are selected to achieve their goal (optimization of service quality parameters). This game is defined as a timing game in the edge environment in a triple pair consisting components  $G = \{N, S, U \dots\}$ . To apply game theory to task scheduling and its allocation to resources, we must first define the game to meet the requirements. The game scheduling task variables are defined in detail as follows:

- (i) **Players:** In the game of task scheduling to resources in the proposed method, virtual machines act as players in a dynamic edge environment. All virtual machines are divided into  $n$  groups according to their assignment probability and the machines in each group act as potential players for a game.

Given that the  $M_k$  machine, which belongs to the  $gt$  group machine from the edge environment, is a suitable option for allocation based on the cost solution of the genetic algorithm stage and considering the probability of allocations, the virtual machines in the  $gt$  group of players are candidates for the next stage of the game. They can be candidates because the probability of distribution of this group of virtual machines is high, and if the genetic algorithm is in the proposed solution, it can be selected as the next stage player

- (ii) **Strategies:** In edge environment scheduling, task priorities are determined by the control modules in the edge environment according to whether the task should be performed or transferred in order to balance the edge environment. The overall



TABLE 2: Complexity of each step in proposed method.

Operator	Complexity
1-scheduling step {	
1-1 Genetic algorithm {	(it = n)
1-1-1 Generate randomly initial population	(c = constant)
1-1-2 Evaluating the initial N chromosome	(n)
1-1-3 Crossover the initial population	(n/2)
1-1-4 Mutation the population	(n)
1-1-5 Select the optimal chromosome}}	(log (n))
2-Load balancing step {	
2-1 Hidden Markov Model {	
2-1-1 Calculate the probability of load in VMs}	(m)
2-2 Game theory {	
2-2-2 Calculate of the objective in each VMs vs Scheduled solutions}}	(m * n)

strategy in the edge environment is to improve run-time, execution cost, and resource efficiency based on load balancing with each player trying to maximize their fitness by choosing the right processes. According to the solution chosen by the genetic algorithm and also according to the number of groups created in the game, a set of n tasks is created that corresponds to n groups of virtual machines. The first task in the edge environment is examined according to the near-optimal solution for the virtual machine in group i which is located in the first gene of the superior chromosome. If other machines in group i have the ability to increase the improvement of the proportion of the first task, the index of the virtual machine in the first gene will be replaced with a more efficient virtual machine in group i. In the edge environment scheduling game, each machine tries to organize its processing queue in a reasonable way so that it can maximize the efficiency of the resource and reduce the time and cost of the task and then by adopting appropriate strategies, tries to minimize the efficiency of other machines in a group that are considered as rivals. Therefore, it should be noted that the fitness of a machine is influenced not only by its chosen strategies but also by the strategies chosen by other machines in the game based on load balancing

- (iii) Fitness function: Fitness function in the game indicates the strength of the players at different stages. Fitness acts as an indicator for machines to choose their strategies. In the proposed method, the repayment of a machine is related to the load index and the amount of productivity of resources and the time and cost of completing the task. As mentioned in the previous section, proportion is directly related to productivity and inversely related to time, cost, and load balance. Therefore, the proportion of the Mk virtual machine in the gt group at stage h of the game is defined in

$$U_k^h(M) = \frac{\eta_k^h}{TC_k^h}. \quad (8)$$

In particular, when a processing task is taken from the Mk machine and assigned to another virtual machine with the same or a higher probability allocation, the Mk permissible machine is defined as zero so that it can participate in the rest of the game. Figure 5 illustrates the proposed method flowchart.

3.4. *Complexity of the Proposed Method.* In order to calculate the complexity of the proposed method, all operators in the proposed method must be considered for the tasks and virtual machines used. The proposed method has two main steps, including scheduling and load balancing. Also, n tasks are assigned to m virtual machines during it iteration of the genetic algorithm. In Table 2, we examine the operators and the complexity of each step in detail, and then, the order of the complexity of the proposed method is given in

$$T(n) = n \times \left( n + \frac{n}{2} + n + \log(n) \right) + (m + (m \times n)) + C,$$

$$T(n) = 2n^2 + \frac{n^2}{2} + (n \times \log(n)) + m + (m \times n) + C,$$

$$T(n) = \frac{7n^2}{2} + (n \times \log(n)) + m + C,$$

$$T(n) \in O(n^2).$$

(9)

As shown in Equation (9), complexity order of proposed method is  $O(n^2)$  that is directly related to the number of input tasks.

#### 4. Proposed Method Implementation

In order to implement the proposed method, random scenarios with 50, 60, 70, 80, 90, and 100 services in the form

TABLE 3: An example of the initial population in the proposed genetic algorithm.

T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	T <sub>4</sub>	T <sub>5</sub>	T <sub>6</sub>	T <sub>7</sub>	T <sub>8</sub>	T <sub>9</sub>	T <sub>10</sub>	T <sub>11</sub>	T <sub>12</sub>	T <sub>13</sub>	T <sub>14</sub>	T <sub>15</sub>	T <sub>16</sub>	T <sub>17</sub>	T <sub>18</sub>	T <sub>19</sub>	T <sub>20</sub>
2	5	5	5	5	6	2	3	3	6	4	3	2	6	2	6	1	1	2	1
1	1	5	4	3	4	3	4	6	1	5	1	6	1	2	5	6	3	5	5
1	6	6	2	2	3	2	1	2	4	4	1	1	2	1	5	2	1	2	1
1	2	1	5	5	4	6	6	6	2	6	6	2	6	2	2	5	4	4	2
1	3	3	5	2	2	1	3	2	4	1	5	4	3	6	2	1	5	3	5
6	2	6	5	1	3	3	3	6	2	3	2	3	4	5	1	5	6	2	2
3	6	6	3	1	5	1	5	5	1	4	4	6	1	2	3	6	5	3	6
5	1	5	4	1	3	1	3	6	4	5	5	2	6	6	1	5	6	5	2
5	2	2	1	2	6	3	5	1	5	2	4	6	2	1	4	6	6	4	3
1	5	6	3	3	1	5	1	4	4	5	5	6	5	1	5	5	4	4	3

of a workflow and 10 virtual machines with different features to perform tasks in the edge environment are defined. The proposed method is simulated in MATLAB software version 2020. Services sent to the edge computing environment are workflows in which one service could be randomly dependent on the others. Therefore, considering the task interdependence, data transfer among services and distribution of services in different virtual machines in terms of geography and priority in providing services, the scheduling of tasks in order to maintain the quality of service and load balance in the edge computing environment is complicated. In the proposed method, when a workflow is introduced, the services are randomly assigned to virtual machines to form the chromosomes in the proposed genetic algorithm. In the proposed method, in the first stage, the genetic algorithm provides a random scheduling by assigning tasks to resources. In the next steps, based on the proposed control framework, in order to achieve the goals of increasing resource efficiency, reducing the total time of tasks in the edge environment, and reducing energy consumption of resources in performing tasks as well as reducing the cost of performing tasks on resources, based on the fitness and mutation operators in the proposed genetic algorithm, this schedule changes and tends towards optimal points. Table 3 shows an example of the initial population in the proposed method.

Accordingly, in the first step of the proposed method, we evaluate the initial population using the fitness function introduced in Equation (1). In this regard, each chromosome according to the priority of the service, the relationship between the services, the time of service based on the number of instructions in each service, the cost of service, and the energy required to run each service in the virtual machine distributed in the environment edge computing is subject to evaluation. Given that the objective function of the proposed genetic algorithm uses a linear optimization problem to improve service quality parameters, the best fitness is given to the chromosome that balances the execution of services in virtual machines. Hence, for each of the chromosomes created in the initial population, a fitness value is calculated, and the value of this fitness function is obtained from the optimality of each gene or the execution of a service in a virtual machine based on the characteristics of that virtual machine. In fact, the fitness of a chromosome is equal to

TABLE 4: Fitness values of the initial population.

Chromosome index	The value of the fitness function
1	0.8215
2	0.8557
3	0.7521
4	0.8492
5	0.7416
6	0.7554
7	0.8153
8	0.8062
9	0.8221
10	0.8036
11	0.7565
12	0.8050
13	0.8715
14	0.8123
15	0.7669
16	0.7583
17	0.8257
18	0.8040
19	0.8565
20	0.8084
21	0.8420
22	0.8075
23	0.7664
24	0.8113
25	0.7921
26	0.7627
27	0.7550
28	0.8111
29	0.7945
30	0.7315

the average fitness of each of the genes in the corresponding virtual machines. Table 4 shows the fitness values of the chromosomes in the initial population.

TABLE 5: Fitness values for population after crossover.

Chromosome index	The value of the fitness function
1	0.8406
2	0.8467
3	0.8189
4	0.8510
5	0.8070
6	0.8215
7	0.8119
8	0.7967
9	0.8213
10	0.8609
11	0.8094
12	0.8244
13	0.7785
14	0.8749
15	0.8133
16	0.8355
17	0.8162
18	0.8049
19	0.8537
20	0.8144
21	0.8082
22	0.8084
23	0.7967
24	0.8119
25	0.8018
26	0.8738
27	0.7912
28	0.8659
29	0.8376
30	0.7991

TABLE 6: Fitness values for population after mutation.

Chromosome index	The value of the fitness function
1	0.8211
2	0.8486
3	0.8112
4	0.8125
5	0.8040
6	0.8084
7	0.8039
8	0.8111
9	0.8221
10	0.8113
11	0.8113
12	0.8123
13	0.8153
14	0.8050
15	0.8186
16	0.8266
17	0.8112
18	0.8558
19	0.8125
20	0.8030
21	0.8065
22	0.8075
23	0.8078
24	0.8123
25	0.8541
26	0.8123
27	0.8110
28	0.8040
29	0.8053
30	0.8153

As shown in Table 4, the value of the fitness function for each of the chromosomes in the initial population is calculated according to the average fitness for each of the chromosomes. The fitting operator is applied on the initial population in order to change the new population in order to improve the optimization and increase the value of the fitness function and to diversify the new population compared to the initial population. In the proposed method, the fitness probability value is considered 0.6, which is selected according to the previous methods and can be changed. Given that the chromosomes in the initial population have 50 genes, 30 genes from chromosome  $i$  and the remaining 20 genes from chromosome  $i + 1$  are selected, and a new population is created based on this crossover operator. Now, in this step, the evaluation is done using the proposed fitness function on the new population. Table 5 shows the values of the fitness function for the chromosomes in the new population.

As shown in Table 5, the values of the chromosome after crossover operator are calculated according to the proposed

fitness function. It can be seen that about 75% of the chromosomes, after crossover, have improved compared to the initial population. In the next step, in order to create diversity and improvement in the new population, the mutation operator has been applied. The mutation operator randomly selects one or more genes on some chromosomes in a population and replaces their values with allowable values. In fact, the mutation operator takes one or more services from the virtual machines to which they are assigned and then assigns them to other virtual machines. In this case, the chromosomes will change, and of course, there will be a change in the fitness of chromosomes. Table 6 shows fitness values for mutant chromosomes.

As shown in Table 6, the values of the fitness function for the mutant population are calculated. By looking closely at the values of the mutant population fitness function, it can be seen that approximately 50% of the chromosomes that were mutated showed improvement and the remaining 50% had the opposite results. Figure 6 shows the fitness

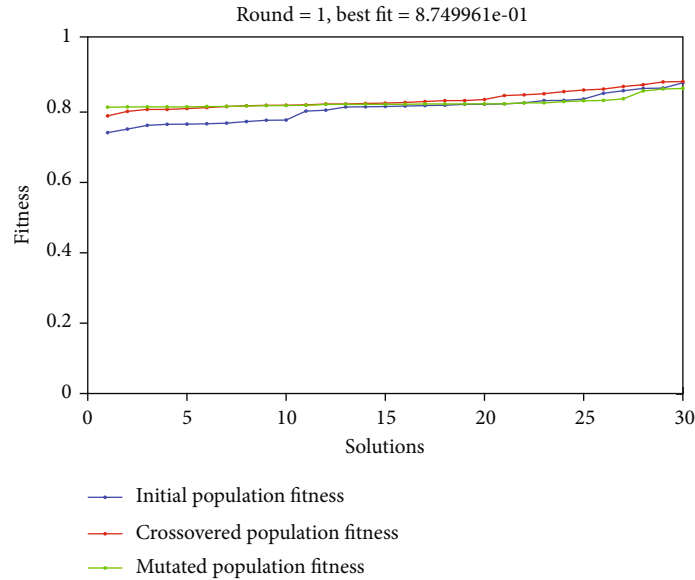


FIGURE 6: Fitness function values in the first stage of the genetic algorithm.

improvement graphs for the initial population, the new population after crossover, and the mutation in the first stage of the genetic algorithm.

As shown in Figure 6, the values of the fitness function for the initial population are shown in blue, the crossovered population in red, and the mutated population in green. The crossovered population is generally higher than the other two populations, and this indicates load balancing and service quality factors in scheduling solutions in a crossovered population. The values of the fitted and mutated populations are mostly oriented towards improving the fitness of the original population which indicates the optimization of scheduling based on the proposed method. In the next step, according to the selection operator, those chromosomes that are more fitness than the average proportion of the total chromosomes are selected as the expert population and the rest of the chromosomes are removed from the initial population. Fitness and mutation operators are applied to the new expert population, respectively, in order to find the best population among the scheduling solutions. The final population is the most optimal scheduling solution that, in addition to load balancing, takes service quality factors into account. Figure 7 shows examples of the replication process of a genetic algorithm.

As shown in Figure 7, the values of the fitness function are gradually optimized in order to get scheduling solutions, and finally, the best population with the fitness function of 97.4 is selected as the proposed scheduling based on the genetic algorithm. After this stage, according to the proposed method, control solutions for load balancing and service quality factors are presented in two steps, Markov model and game theory.

*4.1. Implementing the Control Framework.* After assigning services to virtual machines, the queue of these virtual machines can be more crowded than the others considering

that powerful virtual machines may receive more tasks due to the fit function, and obviously, it will improve the balance of load distribution among virtual machines and require more time to perform all services as well which could affect service quality factors. Therefore, in order to overcome this challenge, the hidden Markov model has been applied. The proposed hidden Markov model, which is part of the proposed control framework, calculates the probability of allocations according to the allocation of services for each of the virtual machines. Based on this allocation probability, the new service will be allocated with respect to the fitness function in the genetic algorithm, the number of pending services and the number of services assigned to the genetic algorithm output population, the performance of the virtual machine in the data set, and the allocation probability of the Markov model to the most suitable virtual machine. Table 7 shows the probability of each virtual machine when allocating new services based on the Markov model.

As presented in Table 7, the Markov model calculates an allocation probability for each virtual machine, and based on this allocation probability, it is possible to understand how the previous services were distributed and how full the queue of each processor is. During each round of service allocation to virtual machines, the allocation probability is updated, and each service is assigned to the virtual machine with the highest allocation probability. Therefore, the new services are directed toward the most appropriate virtual machines considering the allocation probability and the population provided by the genetic algorithm. In order to determine the status of service quality factors during each step of assigning the service to a virtual machine in the proposed method, the game pattern is used as another component of the hierarchical control framework. Due to the fact that a task completion deadline is defined for each service and computing task in the edge computing environment, in order to help increase the reliability of the scheduling

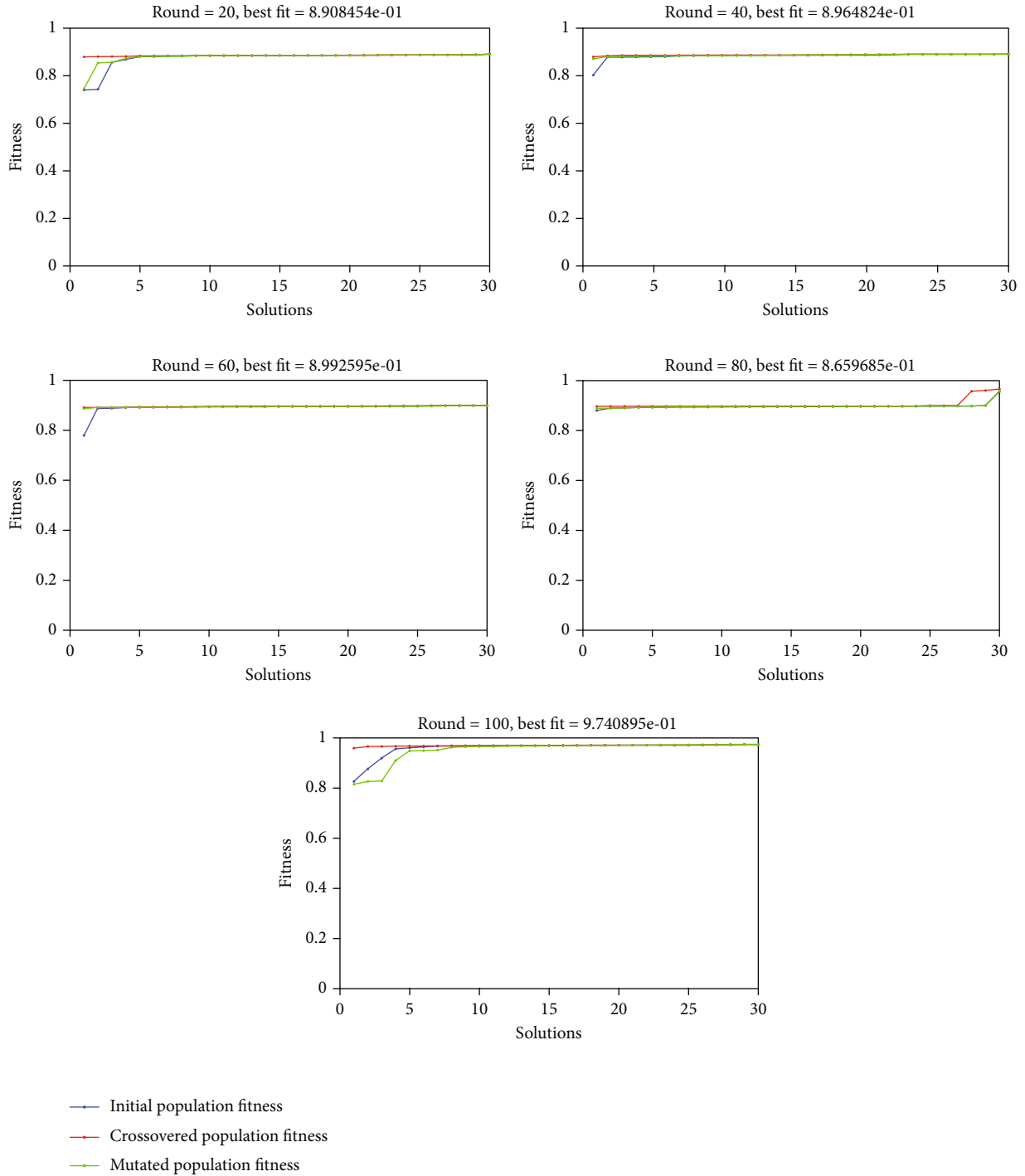


FIGURE 7: Process of optimization of the fitness function in the genetic algorithm.

method in the edge environment, in addition to calculations related to completion time, cost, and energy consumption in the edge computing environment, the possibility of performing a service in a virtual machine is considered to prevent the expiration of the deadline for service. In case the virtual machines are not able to run the service within the specified time, the scheduling operation is performed again, and the service is assigned to another virtual machine. Table 8 shows

the game process for a round of service allocation to virtual machines.

As shown in Table 8, the allocation of services to virtual machines is presented according to the fitness function for the genetic algorithm, the probability of allocations, and the optimal amount of each virtual machine in the multifactor game. Services that cannot be processed by any virtual machine are also identified. In the following, we will evaluate the proposed method.

TABLE 7: Probability of allocation to virtual machines.

Virtual machine	Allocation probability
1	0.5227
2	0.1250
3	0.4053
4	0.1439
5	0.1098
6	0.1856
7	0.2689
8	0.1780
9	0.4696
10	0.8534

TABLE 8: Game theory strategy in the proposed hierarchical control framework.

Virtual machine number	Game theory results		Out of deadline services index
	Strategy A	Strategy B	
1	0.374	0.626	3
2	0.749	0.251	2, 7, 25
3	0.6	0.4	3, 2, 7
4	0.571	0.429	—
5	0.444	0.556	25
6	0.455	0.545	7, 25
7	0.462	0.538	10
8	0.467	0.533	—
9	0.471	0.529	10
10	0.474	0.526	3

*4.2. Performance Evaluation for the Proposed Method.* After implementing the proposed method in the form of a hierarchical control framework consisting of linear optimization, in order to create scheduling plans based on time, cost, and energy factors using genetic algorithms and calculating the probability of allocating services to virtual machines using hidden Markov model and load balance control and service quality factors as well as feasibility of performing each task and providing services in virtual machines based on deadline by dynamic multifactor theory of the game, we evaluate the performance of the proposed method in order to provide the improvement obtained from the combination of the above methods, and then, we evaluate the quality of the proposed method. Due to the importance of load balancing and optimal scheduling in performing services and computational tasks for users in the edge computing environment and in order to comply with service quality factors, various evaluation criteria have been introduced to measure the improvement of scheduling methods in publications. Considering the mentioned goals for the research, the proposed method has been investigated in terms of service completion time in virtual machines (makespan), the total cost of services in virtual machines, the energy required to run all services in the computing center and data storage in edge computing environment, the performance efficiency of vir-

tual machines in edge computing environment, and the reliability of the proposed scheduling approach.

One of the main criteria for evaluating scheduling approaches in the edge computing environment is the time to complete services in virtual machines, known as makespan. Calculating this criterion requires calculating time from the moment of entering a task to the moment of completing it which consists of waiting time in order to access the virtual machine, the time required in order to transfer data in the network in case of need for transferring data from another service, the execution time, and the time required to perform a computational task or to provide services. Figure 8 shows the service completion time graph in virtual machines in the scenarios of proposed method.

As shown in Figure 8, the time required to complete services in virtual machines increases with the number of services. What can be seen is that increasing the number of services does not have much effect on the completion time of all tasks in virtual machines. Due to the near-optimal scheduling in the proposed method, the time required to complete the services with increasing their number increases appropriately, which indicates the efficiency of the proposed method in controlling the load and scheduling of tasks. Also, this indicates that there is not a long waiting time to get to the virtual machine. We can also observe that long services require a minimum runtime due to the fact that powerful virtual machines are responsible for long services.

Another criterion that has been evaluated in the proposed method is the cost of running services in virtual machines. The cost of performing services includes the cost of transferring information between two tasks in virtual machine network and the cost of performing a task or service on a virtual machine. Therefore, it is obvious that if one service is dependent on another service, the cost of performing it will be higher than other tasks. Figure 9 shows the cost of performing tasks in scenarios in virtual machines.

As shown in Figure 9, the cost of tasks and services entering the edge environment varies depending on the relationship between the services and some tasks, and services that require the transfer of information between virtual machines in the edge, via the network, will cost much more. Now, according to the proposed scheduling method, the total cost of performing tasks in scenarios in virtual machines is shown in Figure 10.

As shown in Figure 10, in the proposed method, the cost for executing and transferring data between virtual machines has increased via increasing the number of tasks. To understand the cost reduction in the proposed method, we can consider that the cost of 100 tasks using a random scheduling method such as the initial population of the genetic algorithm would be around 9000\$, while in the proposed method, this cost is at most of \$ 2,500, which shows a significant improvement. The proposed method has scheduled tasks in near optimally manner among virtual machines, and the cost reduction is achieved by using cheaper virtual machines for tasks that require communication.

Another criterion used to evaluate the hierarchical control framework mentioned in the proposed method is the energy consumption to perform computational tasks and

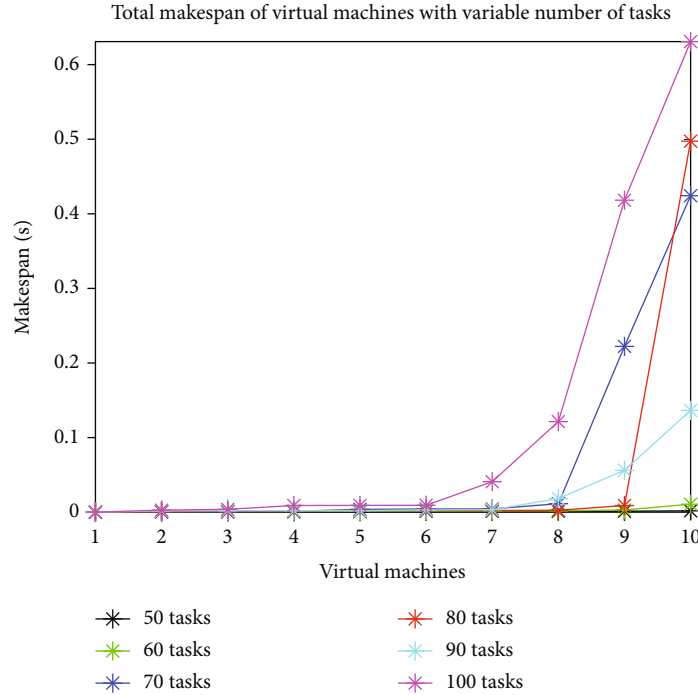


FIGURE 8: Service completion time in scenarios in virtual machines.

provide services in virtual machines. In order to perform tasks and provide services as well as computing and transferring data over a network between virtual machines, we need energy, and naturally, the presence of more communication in the workflow requires more energy. Figure 11 shows the energy consumption in scenarios in the proposed method.

As shown in Figure 11, the energy required to perform tasks in virtual machines increases with the number of tasks and the communication between tasks. The same procedure is used in the proposed method, but it can be seen that increasing the number of tasks does not impose unreasonable energy consumption on data centers.

Another criterion that has been evaluated in the proposed method is the performance of virtual machines during edge tasks execution. Virtual machine performance means the maximum use of virtual machines in performing computational tasks and providing services. Figure 12 represents the performance for virtual machines according to the scenarios in the proposed method.

As shown in Figure 12, in the proposed method, most virtual machines have high performance that indicates the distribution of uniform loads among the virtual machines in the edge computing environment. According to the Figure 12, the performance of virtual machines for different scenarios is almost close to each other and near to optimal. Proper scheduling of tasks to virtual machines with fair distribution of load among virtual machines has prevented virtual machines from being idle and losing cycles.

The last criterion considered in the proposed method is reliability. The proposed method relies on predicting the load probability in virtual machines based on the Markov model and avoids sending tasks to inappropriate virtual

machines. It can be prevent the loss of deadline to perform tasks. This allows virtual machines to receive tasks that they are able to perform on their deadline. Therefore, the reliability of the proposed method will be high, and the tasks will be performed on the deadline. Figure 13 shows the reliability of the proposed method in the scenarios.

As can be seen from Figure 13, the reliability of the proposed method is close to each other in different scenarios. Proper distribution of load among virtual machines and considering the deadline for performing tasks based on the probability of load distribution in virtual machines has resulted in the proposed method of executing tasks with high reliability.

*4.3. Comparison of the Proposed Method with Previous Methods.* After implementing and evaluating the proposed method in order to measure the validity of its performance, we compare it with previous methods in the field of load balancing in the edge environment. Due to the importance of load balancing and creating appropriate schedules in order to benefit from service quality factors in edge computing, many articles have been presented, and many researchers have shown interest in this field. Therefore, the proposed method can be compared with the previous methods [14, 18, 32] based on the main criterion in scheduling methods, namely, the completion time of tasks in edge environment resources, which is considered and evaluated in many articles. Figure 14 illustrates a comparison of the proposed method with the methods available in the publications from the makespan criterion point of view.

As shown in Figure 14, the proposed method has performed better than other methods in publications in

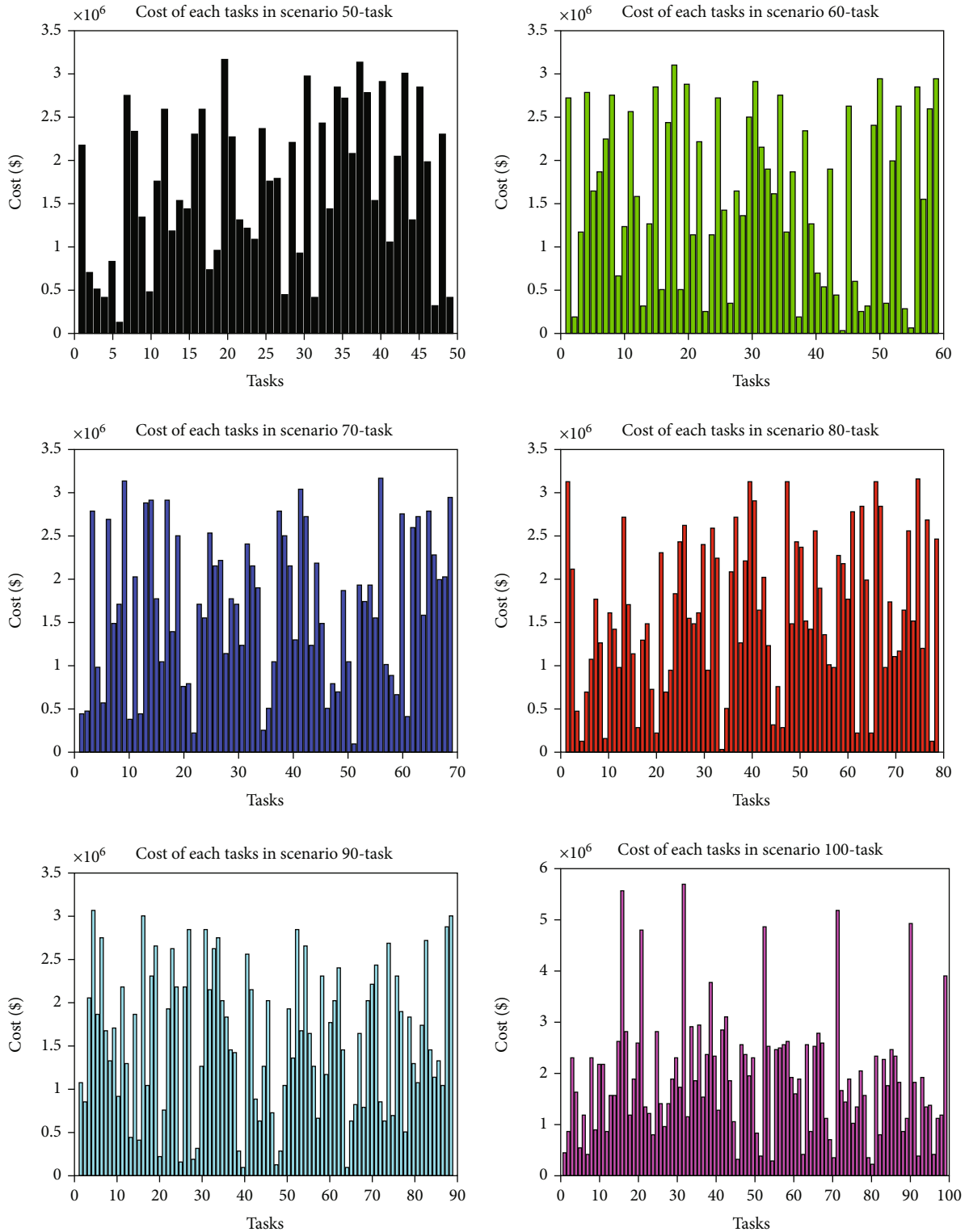


FIGURE 9: Cost of each tasks in scenarios.

allocating services to virtual machines. It has also reduced the service completion time in virtual machines. Therefore, the proposed method has a lower value for the makespan criterion compared to the previous methods and has well complied with service quality factors.

Another criterion added in this paper to compare with previous methods is the performance of the scheduling approach in the fog environment. The proposed method achieves good performance by assigning optimal tasks to virtual machines suitable for different scenarios (as shown in



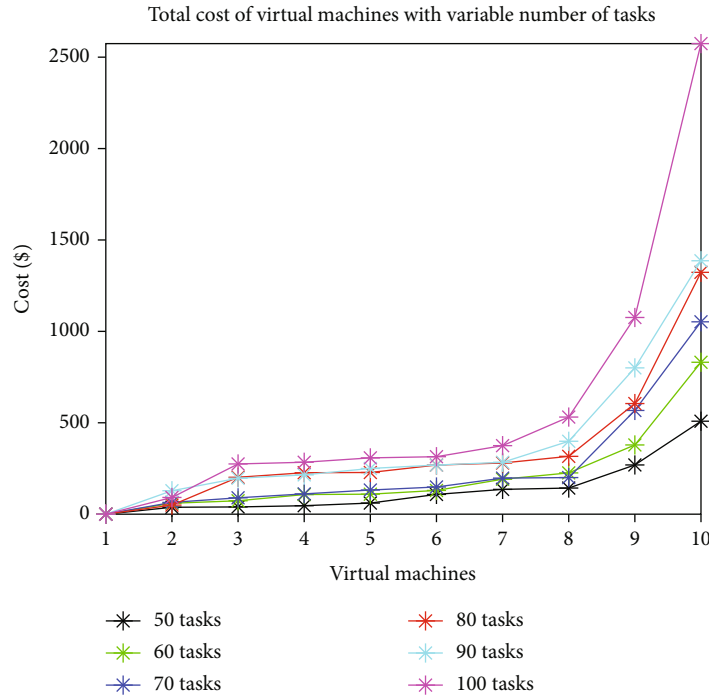


FIGURE 10: Total cost of performing tasks in the proposed method.

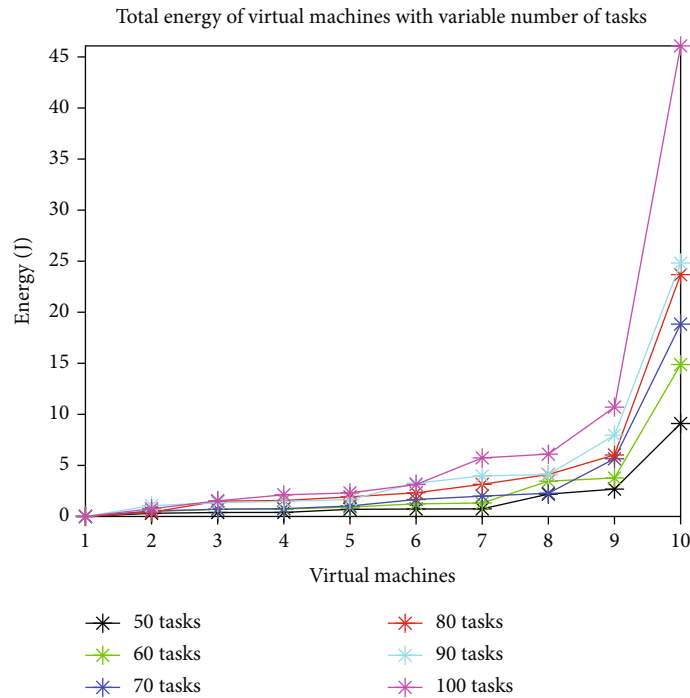


FIGURE 11: Energy consumption in scenarios in the proposed method.

Figure 12). Figure 15 shows a comparison of the performance of the proposed method with previous methods [14, 32, 46].

As shown in Figure 15, the performance of the proposed method is better than other existing methods. The proposed method by applying the control step to adjust and balance

the load in the cloud environment and the use of linear programming in the genetic algorithm has been able to assign optimal tasks to the resources so that the performance of resources in the fog-cloud environment is near-optimal.

Another important issue in load balancing in a cloud environment is the cost of performing tasks on virtual

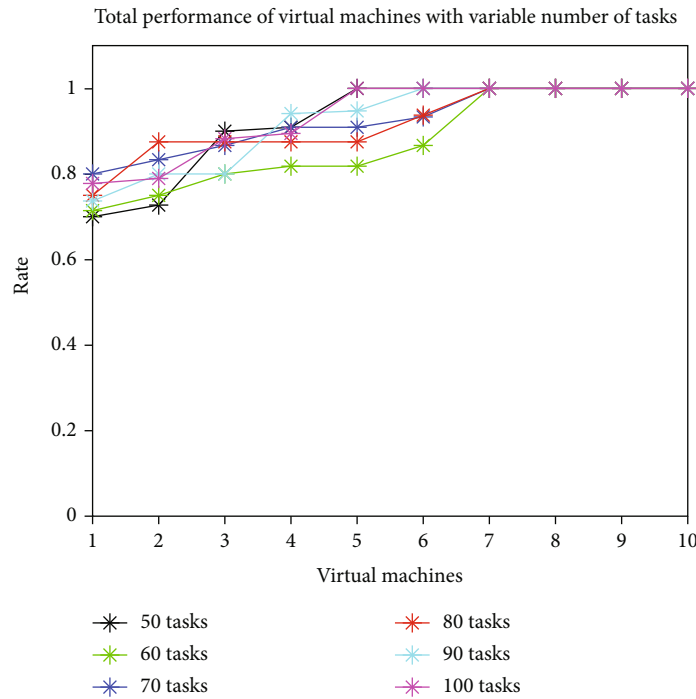


FIGURE 12: Virtual machine efficiency in the proposed method.

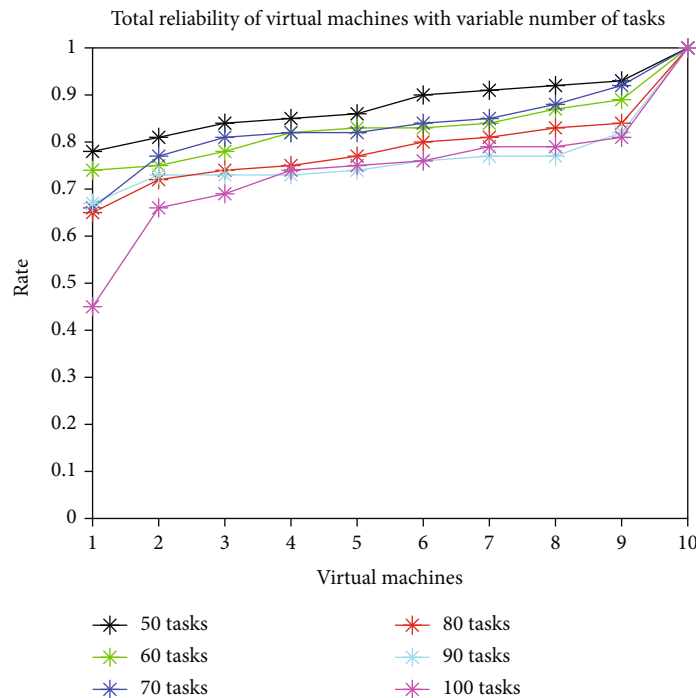


FIGURE 13: Reliability value of task executing in scenarios in proposed method.

machines. In this paper, in order to show the improvement of the proposed method in reducing the costs associated with performing tasks in virtual machines, we compare the results of the proposed method with other existing methods [47, 48]. Figure 16 shows a comparison of the cost of performing tasks in virtual machines.

According to Figure 16, it can be seen that the proposed method has a lower cost of performing tasks than other existing methods. Considering that in the proposed method, one of the main parameters in the fitness function of the genetic algorithm has been the cost of performing tasks, the proposed method by optimizing this parameter has

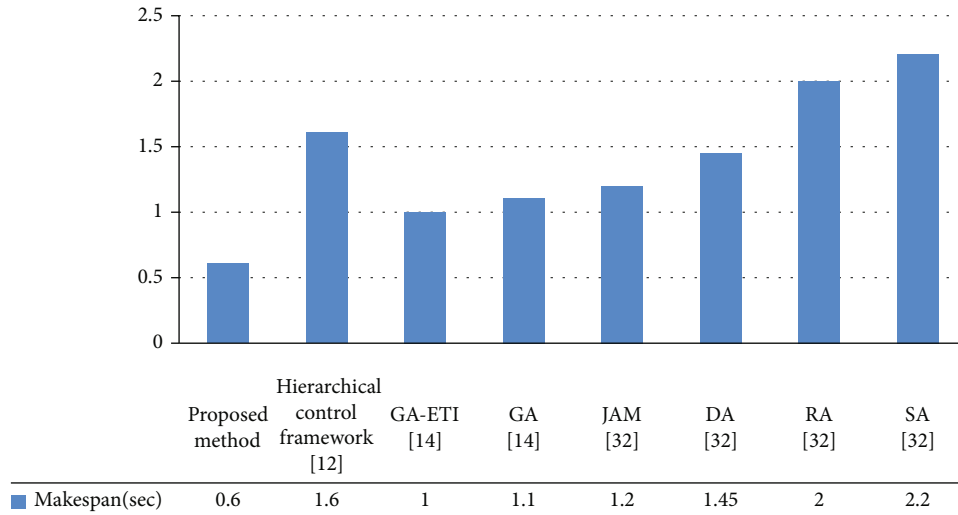


FIGURE 14: Comparison of the makespan in proposed method with previous methods.

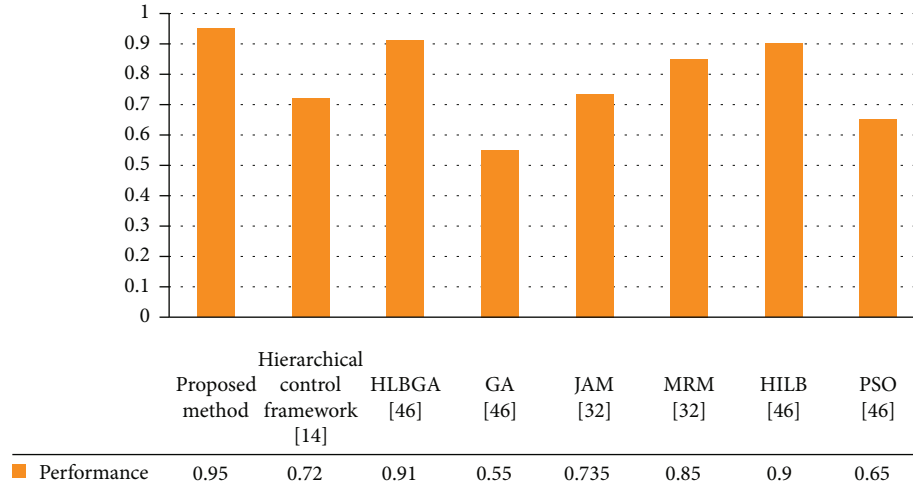


FIGURE 15: Comparison of the performance in proposed method with previous methods.

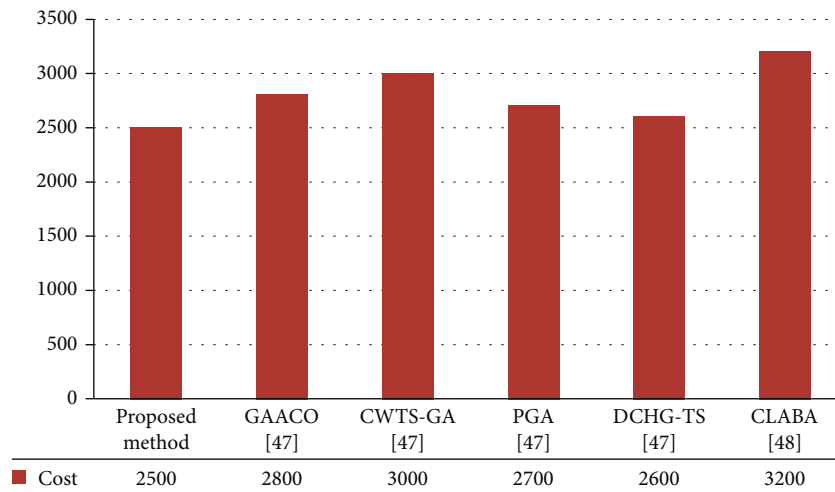


FIGURE 16: Comparison of the performance in proposed method with previous methods.

allocated tasks to resources at a reasonable cost. For this reason, the overall cost of performing the tasks is optimized compared to previous methods.

## 5. Conclusion

In this research, in order to balance the load in the edge computing environment, a hierarchical control framework for assigning tasks in edge computing services is presented. In the proposed method in the first stage, genetic algorithm is used to model the workload. The input of the genetic algorithm in the proposed method includes a set of tasks that are evaluated according to the priorities and the relationship between the tasks in order to model the load distribution among the existing hosts as well as optimizing the criteria. The evaluation criteria applied in this method include time constraints, resource processing power, probability of access to the resource, and the cost of performing tasks in the resource. In order to optimize the load distribution among sources, considering the existing limitations, the integer linear programming optimization in the evaluation function of the genetic algorithm has been used. In the second stage, according to the past load distribution in edge resources, the probability of load distribution among sources is calculated according to the hidden Markov model. Finally, in order to map the tasks to the virtual machines in each host, the game theory with QoS factors has been used as an evaluation function. The results of the experiments show that the proposed method, in comparison with other methods available in publications, in addition to providing balanced service allocations to virtual machines, has reduced the service completion time as well. Therefore, the proposed method has minimized the makespan compared to the previous methods and has well complied with service quality factors.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] V. Lander, H. Sastry, S. Katti, S. V. Vepa, and S. V. Shenoy, "Identity cloud service authorization model," US Patent 10454940B2, 2019.
- [2] J. B. Rajkumar Buyya and A. Goscinski, *Cloud computing principles and paradigms*, John Wiley & Sons, 2019.
- [3] M. Alouane and H. El Bakkali, "Virtualization in cloud computing: NoHype vs HyperWall new approach," in *2016 International Conference on Electrical and Information Technologies (ICEIT)*, pp. 49–54, Tangiers, Morocco, 2016.
- [4] P. Kumar and R. Kumar, "Issues and challenges of load balancing techniques in cloud computing: a survey," *ACM Computing Surveys*, vol. 51, no. 6, pp. 1–35, 2019.
- [5] S. Einy, C. Oz, and Y. D. Navaei, "The anomaly- and signature-based IDS for network security using hybrid inference systems," *Mathematical Problems in Engineering*, vol. 2021, Article ID 6639714, 10 pages, 2021.
- [6] S. Afzal and G. Kavitha, "Load balancing in cloud computing—a hierarchical taxonomical classification," *Journal of Cloud Computing*, vol. 8, no. 1, p. 22, 2019.
- [7] M. Chiregi and N. J. Navimipour, "A new method for trust and reputation evaluation in the cloud environments using the recommendations of opinion leaders' entities and removing the effect of troll entities," *Computers in Human Behavior*, vol. 60, pp. 280–292, 2016.
- [8] Y. D. Navaei and M. Afzali, "A survey on product recommendation system in e-commerce," *International Journal of Computer & Information Technologies*, vol. 2014, 2014.
- [9] Y. D. Navaei and M. Afzali, "Dihedral product recommendation system for E-commerce using data mining applications," *International Journal of Computer & Information Technologies*, vol. 3, pp. 610–631, 2015.
- [10] S. K. Mishra, B. Sahoo, and P. P. Parida, "Load balancing in cloud computing: a big picture," *Journal of King Saud University-Computer and Information Sciences*, 2018.
- [11] Y. Nanehkaran, Z. Licaï, J. Chen et al., "Anomaly detection in heart disease using a density-based unsupervised approach," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 6913043, 14 pages, 2022.
- [12] Z. Peng, M. S. Jabloo, Y. D. Navaei et al., "An improved energy-aware routing protocol using multiobjective particular swarm optimization algorithm," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 6677961, 16 pages, 2021.
- [13] S. Einy, C. Oz, and Y. D. Navaei, "Network intrusion detection system based on the combination of multiobjective particle swarm algorithm-based feature selection and fast-learning network," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 6648351, 12 pages, 2021.
- [14] N. Leontiou, D. Dechouniotis, S. Denazis, and S. Papavassiliou, "A hierarchical control framework of load balancing and resource allocation of cloud computing services," *Computers & Electrical Engineering*, vol. 67, pp. 235–251, 2018.
- [15] Z. Peng, M. Rastgari, Y. D. Navaei et al., "TCDABCF: a trust-based community detection using artificial bee colony by feature fusion," *Mathematical Problems in Engineering*, vol. 2021, Article ID 6675759, 19 pages, 2021.
- [16] S. Einy, C. Oz, and Y. D. Navaei, "IoT cloud-based framework for face spoofing detection with deep multicolor feature learning model," *Journal of Sensors*, vol. 2021, Article ID 5047808, 18 pages, 2021.
- [17] S. Jamali and Y. D. Navaei, "A two-level product recommender for E-commerce sites by using sequential pattern analysis," *International Journal of Integrated Engineering*, vol. 8, no. 1, 2016.
- [18] I. Casas, J. Taheri, R. Ranjan, L. Wang, and A. Y. Zomaya, "GA-ETI: an enhanced genetic algorithm for the scheduling of scientific workflows in cloud environments," *Journal of Computational Science*, vol. 26, pp. 318–331, 2018.
- [19] H. Ibrahim, R. O. Aburukba, and K. El-Fakih, "An integer linear programming model and adaptive genetic algorithm approach to minimize energy consumption of cloud computing data centers," *Computers & Electrical Engineering*, vol. 67, pp. 551–565, 2018.

- [20] W. Wei, X. Fan, H. Song, X. Fan, and J. Yang, "Imperfect information dynamic Stackelberg game based resource allocation using hidden Markov for cloud computing," *IEEE Transactions on Services Computing*, vol. 11, no. 1, pp. 78–89, 2018.
- [21] T. Halabi, M. Bellaiche, and A. Abusitta, "Toward secure resource allocation in mobile cloud computing: a matching game," in *2019 International Conference on Computing, Networking and Communications (ICNC)*, pp. 370–374, Honolulu, HI, USA, 2019.
- [22] O. Elzeki, M. Reshad, and M. Elsoud, "Improved max-min algorithm in cloud computing," *International Journal of Computer Applications*, vol. 50, no. 12, pp. 22–27, 2012.
- [23] P. Samal and P. Mishra, "Analysis of variants in round robin algorithms for load balancing in cloud computing," *International Journal of computer science and Information Technologies*, vol. 4, no. 3, pp. 416–419, 2013.
- [24] V. W. Thawari, S. D. Babar, and N. A. Dhawas, "An efficient data locality driven task scheduling algorithm for cloud computing," *International Journal in Multidisciplinary and Academic Research*, vol. 1, no. 3, 2012.
- [25] S. Sharma, S. Singh, and M. Sharma, "Performance analysis of load balancing algorithms," *World Academy of Science, Engineering and Technology*, vol. 38, no. 3, pp. 269–272, 2008.
- [26] D. Agarwal and S. Jain, "Efficient optimal algorithm of task scheduling in cloud computing environment," 2014, <https://arxiv.org/abs/1404.2076>.
- [27] K. Mahajan, A. Makroo, and D. Dahiya, "Round robin with server affinity: a VM load balancing algorithm for cloud based infrastructure," *Journal of Information Processing Systems*, vol. 9, no. 3, pp. 379–394, 2013.
- [28] G. Joshi and S. Verma, "Load balancing approach in cloud computing using improvised genetic algorithm: a soft computing approach," *International Journal of Computers and Applications*, vol. 122, no. 9, pp. 24–28, 2015.
- [29] C. T. Joseph, K. Chandrasekaran, and R. Cyriac, "A novel family genetic approach for virtual machine allocation," *Procedia Computer Science*, vol. 46, pp. 558–565, 2015.
- [30] S. A. Hamad and F. A. Omara, "Genetic-based task scheduling algorithm in cloud computing environment," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 4, pp. 550–556, 2016.
- [31] T. Wang, Z. Liu, Y. Chen, Y. Xu, and X. Dai, "Load balancing task scheduling based on genetic algorithm in cloud computing," in *2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing*, pp. 146–152, Dalian, China, 2014.
- [32] G. Yi, H. W. Kim, J. H. Park, and Y. S. Jeong, "Job allocation mechanism for battery consumption minimization of cyber-physical-social big data processing based on mobile cloud computing," *IEEE Access*, vol. 6, pp. 21769–21777, 2018.
- [33] M. Kalra and S. Singh, "A review of metaheuristic scheduling techniques in cloud computing," *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 275–295, 2015.
- [34] S. Chakraborty and K. Mazumdar, "Sustainable task offloading decision using genetic algorithm in sensor mobile edge computing," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 4, pp. 1552–1568, 2022.
- [35] M. Abdel-Basset, R. Mohamed, R. K. Chakraborty, and M. J. Ryan, "IEGA: an improved elitism-based genetic algorithm for task scheduling problem in fog computing," *International Journal of Intelligent Systems*, vol. 36, no. 9, pp. 4592–4631, 2021.
- [36] B. Natesha and R. M. R. Guddeti, "Adopting elitism-based genetic algorithm for minimizing multi-objective problems of IoT service placement in fog computing environment," *Journal of Network and Computer Applications*, vol. 178, article 102972, 2021.
- [37] M. Abbasi, E. Mohammadi Pasand, and M. R. Khosravi, "Workload allocation in iot-fog-cloud architecture using a multi-objective genetic algorithm," *Journal of Grid Computing*, vol. 18, no. 1, pp. 43–56, 2020.
- [38] S. Kashyap, S. K. Singh, A. Rouniyar, R. Saxena, and A. Kumar, "Load balancing and resource allocation in fog-assisted 5G networks: an incentive-based game theoretic approach," 2022, <https://arxiv.org/abs/2202.05128>.
- [39] S. Kalantary, J. Akbari Torkestani, and A. Shahidinejad, "Resource discovery in the internet of things integrated with fog computing using Markov learning model," *The Journal of Supercomputing*, vol. 77, no. 12, pp. 13806–13827, 2021.
- [40] A. Mebrek and A. Yassine, "Intelligent resource allocation and task offloading model for IoT applications in fog networks: a game-theoretic approach," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2021, pp. 1–15, 2021.
- [41] S. Balasubramanian and T. Meyyappan, "Game theory based offload and migration-enabled smart gateway for cloud of things in fog computing," in *Computing in Engineering and Technology*, pp. 253–266, Springer, 2020.
- [42] M. Franzese and A. Iuliano, "Hidden markov models," in *Encyclopedia of Bioinformatics and Computational Biology*, pp. 753–762, Academic Press, Oxford, UK, 2019.
- [43] J. Xiao, W. Zhang, S. Zhang, and X. Zhuang, "Game theory-based multi-task scheduling in cloud manufacturing using an extended biogeography-based optimization algorithm," *Concurrent Engineering*, vol. 27, no. 4, pp. 314–330, 2019.
- [44] Y. Zhang, J. Wang, and Y. Liu, "Game theory based real-time multi-objective flexible job shop scheduling considering environmental impact," *Journal of Cleaner Production*, vol. 167, pp. 665–679, 2017.
- [45] Y. Zhang, J. Wang, S. Liu, and C. Qian, "Game theory based real-time shop floor scheduling strategy and method for cloud manufacturing," *International Journal of Intelligent Systems*, vol. 32, no. 4, pp. 437–463, 2017.
- [46] W. Saber, W. Moussa, A. M. Ghuniem, and R. Rizk, "Hybrid load balance based on genetic algorithm in cloud environment," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 3, p. 2477, 2021.
- [47] A. Iranmanesh and H. R. Naji, "DCHG-TS: a deadline-constrained and cost-effective hybrid genetic algorithm for scientific workflow scheduling in cloud computing," *Cluster Computing*, vol. 24, no. 2, pp. 667–681, 2021.
- [48] A. Kishor, R. Niyogi, and B. Veeravalli, "A game-theoretic approach for cost-aware load balancing in distributed systems," *Future Generation Computer Systems*, vol. 109, pp. 29–44, 2020.

## Research Article

# A Novel Link-Network Assignment to Improve the Performance of Mobility Management Protocols in Future Mobile Networks

Jesús Calle-Cancho <sup>1</sup>, Javier Carmona-Murillo <sup>1</sup>, José-Luis González-Sánchez <sup>2</sup>,  
and David Cortés-Polo <sup>1</sup>

<sup>1</sup>Department of Computing and Telematics Engineering, University of Extremadura, 10003 Cáceres, Spain

<sup>2</sup>Research, Technological Innovation and Supercomputing Center of Extremadura (CénitS), 10071 Cáceres, Spain

Correspondence should be addressed to Jesús Calle-Cancho; [jesus.calle@cenits.es](mailto:jesus.calle@cenits.es)

Received 22 September 2021; Accepted 24 February 2022; Published 5 April 2022

Academic Editor: Saba Bashir

Copyright © 2022 Jesús Calle-Cancho et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

5G is expected to support new services and applications that will change the user experience and will drive to a new business landscape. Moreover, most of the services will require optimum connectivity and seamless mobility in heterogeneous networks. To cope with these challenges, network mobility management and network densification are envisioned to be key factors in the emerging 5G architectures. By deploying a large number of cells, 5G architectures can provide users with high throughput and flexible access services, an improvement of the network scalability and optimized network coverage. However, with this densification, seamless mobility support can lead to significant increasing in signaling overhead due to frequent handovers. In this context, network operators need to efficiently plan the deployment of the base stations taking into account the mobility management of the users, and the service degradation that this mobility process could cause. This article aims to optimize the assignment of the base stations to the access routers in the mobile network to improve the network performance. The results obtained show that our proposed Link-Network Assignment algorithm based on clustering techniques achieve significant gains in terms of signaling and data forwarding costs. These simulation results demonstrate that the proposed algorithm can successfully reduce signaling cost and packet delivery cost by up to 56% and 5%, respectively, on average, compared with baseline algorithms.

## 1. Introduction

5G is the evolution of mobile networks which provides higher rates in the transmission, analyzes and manipulates massive amounts of data and applications quickly, and manages the network resources efficiently than ever before.

In this sense, 4G was developed to provide mobile broadband communications to the users; whereas, 5G is conceived as a key technology because it combines entities, communications, and control technologies [1]. 5G is appro-

priated to support enhanced mobile broadband communications, ultra-reliable low-latency communications (URLLC), and massive machine-type communications (mMTC) [2, 3].

The 3GPP architecture implemented in 4G manages the mobility from a centralized scheme [4]. These Centralized Mobility Management (CMM) protocols [5] introduce a mobility anchor that is responsible for data forwarding and signaling traffic management. In the 4G architecture, these nodes are the packet gateway (PGW), which acts as mobility anchor; the serving gateway (SGW); and the

evolved Packet Data Gateway (ePDG) which implements the signaling agent functions and eNode B (eNB) which provides the radio access network. However, some issues affect centralized mobility management protocols, such as nonoptimal routing, scalability problems, and centralized anchors [4, 6].

The architecture in 5G defines a set of functions to handle mobility. These network functions (NF) are defined using a service-based architecture and allow to manage the mobility deploying a distributed scheme [7]. Distributed Mobility Management (DMM) protocols [8, 9] propose to locate closer to the user the mobility anchors to achieve a flatter network. In the 5G architecture, the entities involved in the nodes' mobility are the Access Mobility Manager (AMF) and the Session Management Function (SMF) which assist and manage the control plane of the communication. The User Plane Functions (UPF) is also involved in the 5G architecture, being the data forwarding entity. Then, UPF and SMF replace the entities SGW and PGW deployed in 4G. In this architecture, UPFs are closer to the users acting as mobility anchor towards the access network [10]. In particular, the 5G access network can be a radio access network or any non-3GPP access network, such as WLAN.

The UPF has a key role in the deployment of the Multi-access Edge Computing (MEC) in a 5G network, some specific implementations can include this element as part of the MEC [11]. The MEC approach is implemented in a cloud datacenter located at the edge of the mobile network and distributes computation and storage capabilities reducing communications distance and, consequently, the delay between the mobile network and end-users. This approach also improves the maintenance of the user connectivity, implementing new network functions that provide L3 (network level) mobility support by maintaining active communications when the mobile user performs its movement through the mobility domain.

This new 5G architecture comprising of new radio and core network looks to fulfil the requirements on higher bandwidth and reliability, lower latency, an increment of the network efficiency, and a much higher network densification. To achieve this, operators must plan the resources efficiently to improve the network performance. In this sense, the signaling overhead introduced to manage the user mobility between different access networks and also to handle the IP mobility management increase the latency of the communication.

The main contributions of this paper are summarized as follows:

- (a) The optimization problem is formulated to minimize the impact of base stations assignment to access routers in terms of signaling and data forwarding costs associated with the mobility management protocols
- (b) A novel Link-Network Assignment algorithm is proposed for planning future mobile networks. This algorithm collects information from the access network topology and examines the base stations distribution

in order to perform an appropriate assignment

- (c) A performance evaluation is conducted to show the benefits of the proposed algorithm in terms of well-known mobility costs

This work proposes a mechanism to plan the most appropriate association between base stations and the access nodes to reduce the signaling of the network and improve the performance of mobility protocols in 4G and 5G.

The rest of this paper is organized as follows. Section II describes the background of the work, presenting the most representative mobility management solutions and the analyzed problem. In Section III, the proposed system model and the Link-Network Assignment problem are formulated. In Section IV, the metrics to evaluate the performance of the mobility management protocols are discussed. The considered simulation setup and experimental results are presented and discussed in Section V. Lastly, the conclusions and future works are summarized in Section VI.

## 2. Background

The densification of the network is introduced to address the huge service demands in 5G. The next-generation radio access network (RAN) will be a mixture of various types of RANs macrocells base stations (BSs), femtocell BSs, picocell BSs, and WiFi Access Points. The infrastructure that interconnects those radio access networks with the Internet must provide ubiquitous device connectivity providing seamless mobility support at the network layer.

As seen before, many protocols deploy IP mobility management in the network to provide seamless mobility support. This section presents an overview of the two most representative approaches to mobility management, Proxy Mobile IPv6 (PMIPv6) and Distributed Mobility Management (DMM).

*2.1. Centralized Mobility Management.* PMIPv6 is the most characteristic centralized mobility management protocol [12]. In this approach, a single mobility anchor manages the signaling and traffic of the Mobile Nodes (MNs). PMIPv6 manages mobility by introducing the Mobile Access Gateway (MAG) and the Local Mobility Anchor (LMA), which usually are deployed in the SGW and PGW, respectively, in the 4G architecture. The main role of the MAG is to manage the mobility signaling for an MN that is attached to it, establishing a tunnel with the LMA. The LMA ensures the MN address remains reachable when it performs the movement across the mobility domain. The generic architecture of PMIPv6 is shown in Figure 1.

When an MN connects to a network domain, its traffic is anchored to the LMA and is encapsulated in a tunnel between the LMA and MAG. When the MN performs a handover from MAG-1 to MAG-2, as shown in Figure 1, the binding is updated at the LMA using Proxy Binding Update (PBU) and Proxy Binding Acknowledgement (PBA) messages. Then, a new tunnel is established between

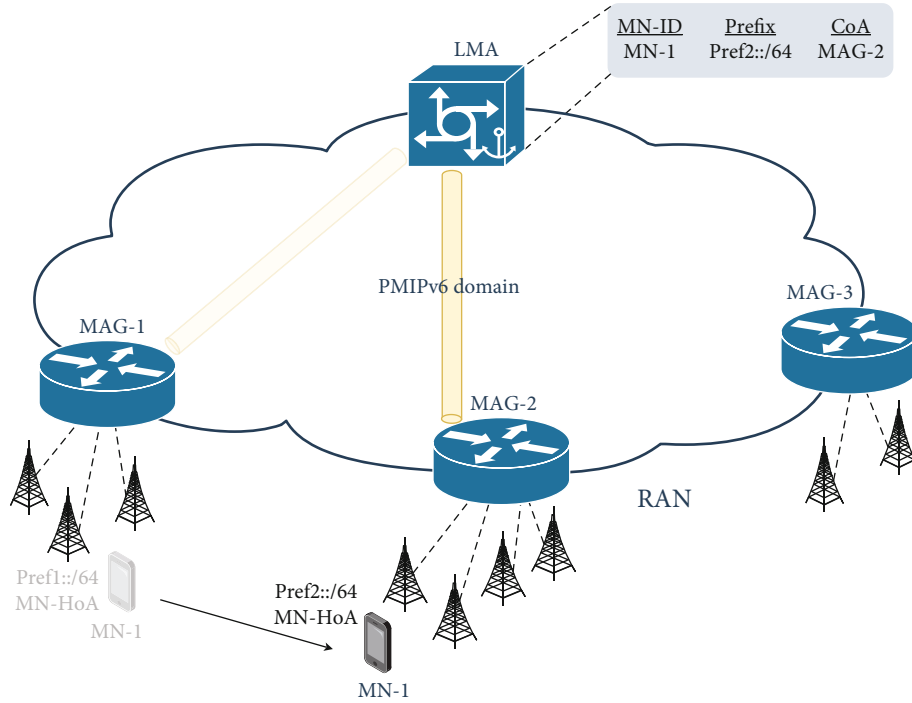


FIGURE 1: Centralized Mobility Management solution: Proxy Mobile IPv6.

MAG-2 and the LMA to keep the ongoing IP traffic flows active.

**2.2. Distributed Mobility Management.** Centralized approaches have several problems and limitations that have been identified in [8, 9]: nonoptimal routing, signaling overhead, and scalability and reliability issues. These limitations have recently propelled the emergence of Distributed Mobility Management. In this approach, the mobility anchors are located and distributed closer to the user, reducing the traffic bottlenecks that affect mobile networks at this time.

A representative proposal of a DMM is Network-Based DMM (NB-DMM) [8]. NB-DMM is a network-based DMM approach, as PMIPv6. This means that the MN does not need to participate in the process of signaling issues related to mobility. Therefore, it is not necessary to update the mobile node’s protocol stack. In this protocol, the mobility management functionalities are moved to the Access Routers (ARs) or an entity with similar features in 5G to anchor the traffic closer to the MN. These entities, called mobility capable access router (MAR), distribute the control and data plane mobility functions along the edge of the access network. The generic architecture of NB-DMM is depicted in Figure 2.

In NB-DMM, when an MN connects to a network domain, its traffic is anchored at the serving MAR.

However, when the MN performs a handover from MAR-2 to MAR-3, as shown in Figure 2, the data traffic of the session is tunneled between the serving MAR (MAR-3) and the anchoring MAR (MAR-2). Thus, upon a handover, the new MAR needs the IP addresses of each previous MAR with active MN’s sessions. These addresses are obtained from the mobility database. Then, the new MAR

notify to each previous MAR, by sending a PBU message, in order to update the location of the MN. Each anchoring MAR replies by a PBA.

**2.3. Link-Network Assignment Problem.** Recently, the academia and the industry have analyzed the different assignment problems in 5G where the principal one is the user assignment to a Base Station in heterogeneous networks [13]. These solutions focus on determining the users (devices) belonging to each base station, addressing the issues of user association in the network, and studying parameters related to physical and data link layers [14]. However, neither solution discusses the association between the stations and the mobile access network, analyzing and improving the behavior of IP mobility management protocols.

Many works often improve some objective in the wireless network like energy [15]. In this work, power consumption is analyzed in a multi-connectivity environment, in which devices are associated with multiple radio access technologies, simultaneously. In [16], the goal is to achieve load balancing in the association between users and base stations taking advantage of the effectiveness in offloading users provided by 5G. Other authors focus on optimizing user association to achieve proportional fairness function among different users of the network [17]. All these solutions also do not consider the association between base stations and the access network, nor the improvement of IP mobility management in 5G environments.

Otherwise, unlike our proposal, works directly related to mobility management protocols [4–9] do not provide performance improvements through the association between the access network and base stations. The solutions based



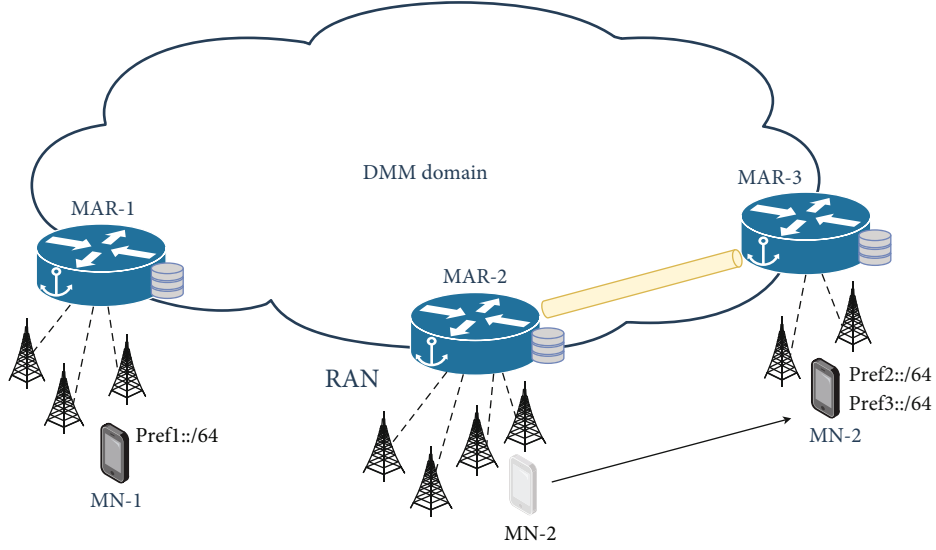


FIGURE 2: Network architecture of the Distributed Mobility Management solution.

on mobility management in 5G-enabled networks generally focus on protocols [18].

Other works study the ability provided by network slicing to assign different sliced radio access networks to various core slices [19–21]. However, these proposals concern requirements for allocating a 5G network slice, without taking into consideration the mobility management protocols.

In this paper, a Link-Network Assignment problem is proposed to improve the performance of mobility management protocols in 5G. To the best of our knowledge, this assignment problem and its solution for future mobile networks analyzing mobility management protocols have not yet been proposed before. All analyzed solutions have taken into account base stations and users to improve the communication. In this approach, the optimization minimizes the signaling of the mobility management protocol, and the latency introduced when a handover is produced. Both of them can be a key aspect to improve in ultrareliable and low-latency communications proposed by the 5G standard. In general, mobility management solutions aim to balance the signaling overhead generated during the movement process with the packet delivery cost caused by the suboptimal routing imposed by the protocols when the user is roaming among different networks. In these cases, the decrease in one of the costs impacts negatively on the other and vice versa. It is worth noting that our solution does not require the modification of any mobility management protocol involved in the communication.

### 3. System Model and Problem Formulation

In this section, the system model is introduced in order to define the optimization problem to minimize the impact of base stations assignment to access nodes, without loss of per-

formance in terms of packet delivery ( $P_{cost}$ ) and signalling ( $C_u$ ) costs associated with mobility management protocols.

**3.1. Mobility Domain and Access Network.** Let a given access network be represented as an undirected graph  $G = (V, E)$ , where  $V$  and  $E$  denote the sets of nodes and links (edges), respectively. Let  $K \subseteq V$  be the set of access routers that give access to mobile users through a set of base stations  $B$ , where each base station is denoted by  $b_i (1 \leq i \leq |B|)$ . This set  $B$  provides full coverage to a geographical area under consideration and each location is given by  $\{L_{b_i}\}_{b_i \in B}$ , where  $L_{b_i} \in \mathbb{R}^2$  represents the bidimensional space where base stations will be located.

**3.2. Base Station Assignment and Mobile Nodes Support.** Each access router  $\{k_j\}_{j \in K}$  serves a given number of base stations  $B_k \subseteq B$  within a network domain. The access routers are defined as the first-hop routers, which can be taken as the link between physical and network levels. Furthermore,  $N$  denotes the set of mobile nodes which moves around the network where each mobile node is defined by  $N_j (1 \leq j \leq |N|)$ , and it is attached to base station  $b_i \in B$ .

Moreover, let us assume that each base station  $b_i$  is linked to a mobility domain access router, as shown in Figure 3, and each access router  $k_j$  manages a set of base stations.

**3.3. Link-Network Assignment Problem.** The optimal assignment between the access network and base stations set is defined as the following optimization problem:

$$\text{Min } F = \sum_{x_{mr}^{ps}} \sum_{m \in B} \sum_{k \in K} \sum_{p \in B} T C x_{mr}^{ps}, \quad (1)$$

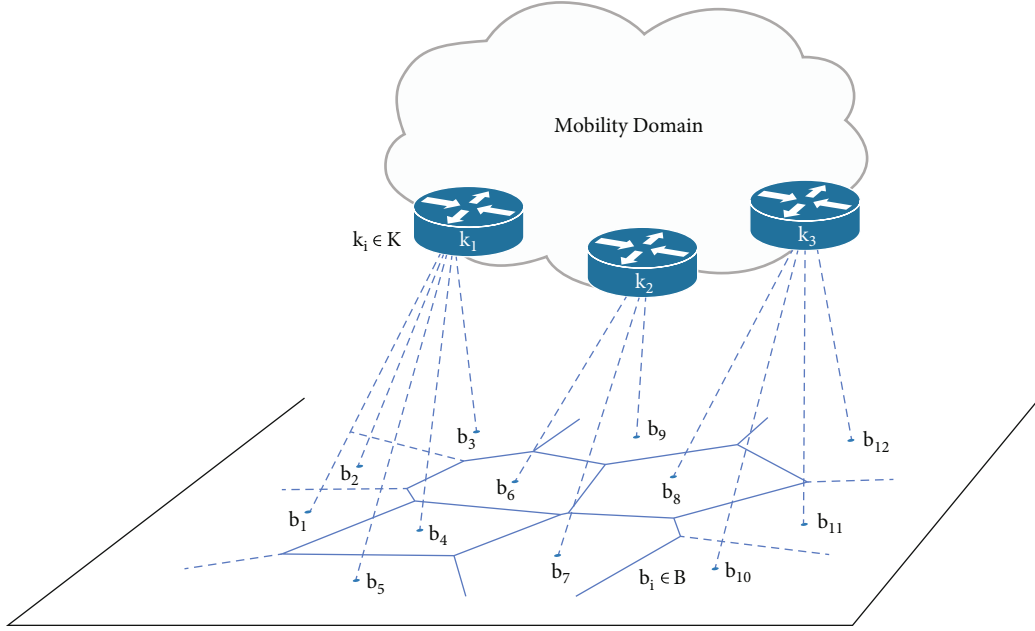


FIGURE 3: Base stations assignment to mobility domain.

subject to:

$$\begin{aligned}
 & \sum_{r \in K} \sum_{p \in B} \sum_{s \in K} x_{mr}^{ps} + \sum_{m' \in B} \sum_{r' \in K} \sum_{s' \in K} x_{m'r'}^{p's'} = 1, \\
 & \forall m = p' \in B, m = p' = 1, \dots, B \\
 & \sum_{m \in B} \sum_{p \in B} \sum_{s \in K} x_{mr}^{ps} + \sum_{m' \in B} \sum_{r' \in K} \sum_{p' \in B} x_{m'r'}^{p's'} \leq th_j, \\
 & \forall r = s' = j \in K, r = s' = j = 1, \dots, K \\
 & x_{mr}^{ps} \in \{0, 1\}, \forall m \in B, r \in K, p \in B, s \in K.
 \end{aligned} \tag{2}$$

Being  $x_{mr}^{ps}$  a binary decision variable defined as:

$$x_{mr}^{ps} = \begin{cases} 1 & \text{if base station } m \text{ is assigned to the access} \\ & \text{router } r \text{ and base station } p \text{ is assigned to} \\ & \text{the access router } s. \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

For each assignment, the total cost  $TC$  is defined as the sum of the main parameters related to mobility management protocols, signalling cost, and packet delivery cost:

$$TC = C_u(\cdot) + P_{cost}(\cdot) \tag{4}$$

Signalling cost considers the traffic load in bytes generated by signalling messages when a layer three handover process occurs in order to maintain the active sessions of each mobile node (MN). Packet delivery cost, calculated in bytes,

measures the cost to forward the data packet in the network. It depends on the size of the data messages and the number of hops needed to forward packets to the MN. Both measurements will be explained in deep in Section 4. Constraint 2 indicates that a base station ( $m = p' \in B$ ) is assigned to a single access router, and constraint 3 is related to the balance of base stations between the different access routers. The assumption is that a given access router ( $r = s' \in K$ ) cannot serve more than a specific number of base stations, determined by a threshold ( $th_j$ ).

**3.4. Link-Network Assignment Algorithm.** The above model computes of the optimal assignment between the physical level and the access network. In this context, the problem size depends on the number of base stations and the nodes of the access network. When the problem size is reduced, for instance, 15 base stations and 3 routers, an optimal solution is found in less than a second. Nevertheless, if the number of base stations increases and large-scale topologies of the access network are used, the time complexity increases exponentially. The optimal bind between base stations and the access network has become a challenge that has to be addressed for future mobile network operators. For this reason, a new strategy is proposed to solve this problem at a weak polynomial time. Thus, Algorithm 1 defines the link-network assignment algorithm exhaustively.

The proposed algorithm collects the information from the access network topology and examines the base stations distribution in order to perform an appropriate assignment. It is composed of three steps that are defined as follows:

- (a) *First step.* A set of data observations (base stations set  $B = \{b_1, \dots, b_{|B|}\}$ ) are classified into a specific

```

Input: Base stations set  $B = \{b_1, \dots, b_{|B|}\}$ , access routers set  $K = \{k_1, \dots, k_{|K|}\}$  and network topology NT
Output: Dictionary DictAssoc with final relation between base stations clusters (centers) and access routers
▷first step
1:  $C = \text{k-means++}(B, |K|)$ ;
   ▷second step
2:  $D_{CKM} = \text{getEuclideanDistances}(C)$ ;
3:  $D_{AR} = \text{allShortestPath}(NT, K)$ ;
   ▷third step
4: For  $c_i \in C$  do
5:    $M_C[i] = \sqrt{c_{ix}^2 + c_{iy}^2}$ 
6: End for
7: For  $k_i \in K$  do
8:    $CC_{AR}[j] = |K| / \sum_{k_j \in K} D_{AR}[k_i][k_j]$ 
9: End for
   ▷fourth step
10:  $MC = \text{argmax}(M_C)$ ;  $cc = \text{argmin}(CC_{AR})$ 
11: For  $j = 1$  to  $|K|$  do
12:    $d_c[j] = D_{CKM}[MC][j]$ ;
13:    $d_{ar}[j] = D_{AR}[cc][j]$ ;
14: End for
15:  $d_c = \text{sorted}(d_c)$ ;  $d_{ar} = \text{sorted}(d_{ar})$ 
16: For  $i = 1$  to  $|K|$  do
17:    $\text{DictAssoc}\{“d_c[i]”\} = d_{ar}[i]$ ;
18: End for
19: Return DictAssoc;

```

ALGORITHM 1: Link-network assignment algorithm.

number of  $|K|$  clusters, matching the number of access routers using a widely known unsupervised algorithm called k-means++ algorithm [22]. This algorithm presents an evolution for centroids initialization in order to improve the computational time that can become exponential if the original algorithm is used. Thus k-means++ minimizes the distance between observations and centroids using Euclidean distance, and it includes an initialization method based on a uniform random variable. This method allows a proper set of initial cluster centroids to be obtained

- (b) *Second step.* For each centroid  $c_i$ , the distances  $D_{CKM}(i, j)$  between  $c_i$  and all others ( $c_j \in C$ ) are calculated using Euclidean distance. Then, the distance matrix of topology access nodes  $D_{AR}$  is built: for each topology access router  $k_i$ , the distance with the others ( $k_j \in K$ ) is computed through Dijkstra algorithm
- (c) *Third step.* Our algorithm selects the highest centroid modulus value. Each centroid  $c_i$  in a bi-dimensional space is defined as  $c_i = [c_{ix}, c_{iy}]$ . Thus, the Euclidean norm or modulus of each centroid can be calculated as the square root of the sum of the squared centroid values
- (a) Moreover, for each access router  $k_i \in K$ , closeness centrality [23] is computed according to

Equation (5) in order to select the node with minimum value.

(b)

$$CC_{AR}(k_i) = \frac{|K|}{\sum_{k_j \in K} D_{AR}(k_i, k_j)} \quad (5)$$

- (d) *Fourth step.* Then, the algorithm performs the first association between the highest centroid modulus value  $MC$  and the access router with minimum value of closeness centrality  $cc$ . Once this is done, vectors distance  $d_c$  and  $d_{ar}$  are extracted;  $d_c$  indicates the distance between  $MC$  and all others; and  $d_{ar}$  stores the distance between  $cc$  and all others. Finally, both vectors are sorted in ascending order and the final association is performed

Figure 4 shows an example of the proposed method in operation. In this case, the access network topology consists of three routers. Therefore, the base stations set is clustered into three groups through the k-means++ algorithm. Then, the proposed Algorithm 1 performs the final assignment, as described above.

#### 4. Performance Metrics

The performance metrics, described in this section, evaluate both NB-DMM and PMIPv6 approaches by means of

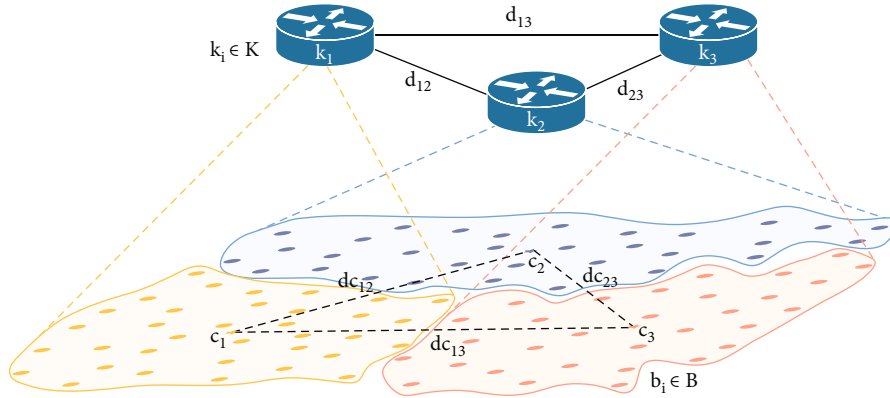


FIGURE 4: Example of proposed link-network assignment algorithm for three access routers.

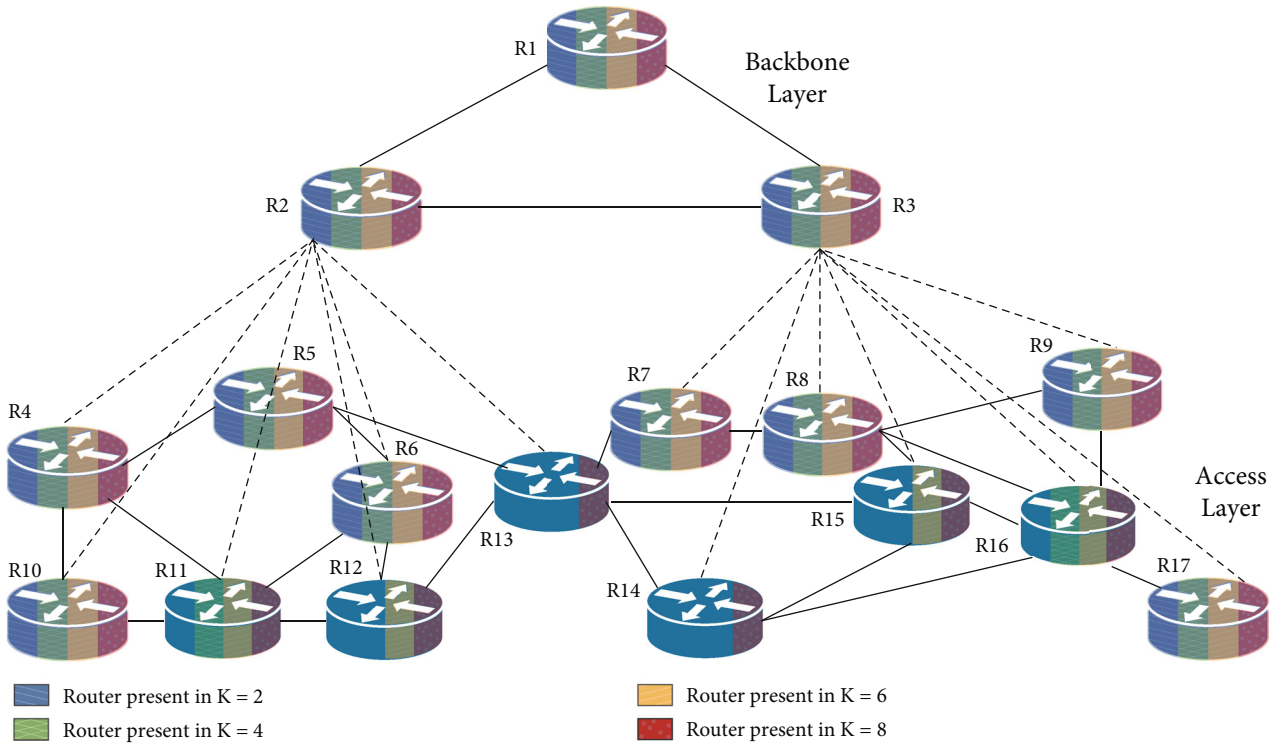


FIGURE 5: The city network topology used in the simulations.

 TABLE 1: Functions of the nodes of the network topology for  $K = 8$ .

	R1	R2–R9	R10–R17
PMIPv6 approach	LMA	Router	MAG
NB-DMM approach	Router	Router	MAR

assessing the signaling cost, the packet delivery cost, and the signaling delay.

The packet transmission cost in IP networks is proportional to the number of hops between source and destination nodes. Thus, the transmission cost of a packet (signaling or data) between nodes  $X$  and  $Y$  can be expressed as  $C(\cdot) = Si ze_{Packet} d_{X-Y}$ .

**4.1. Control Plane.** The mobility support comprises the process of maintaining the MN's sessions while users move through the mobility domain. For this purpose, mobility management protocols are responsible for this process and use signaling messages between the mobility agents.

In order to evaluate the control plane, an important metric is the accumulative traffic overhead in bytes on exchanging signaling messages during the communication session of the MN. Thus, the total cost of signaling for a mobility session is expressed as  $C_u(\cdot)$ , where  $(\cdot)$  is one of the analyzed approaches (PMIPv6 or NB-DMM). This cost is directly proportional to the size of the control messages and the distance in number of hops in each handover during the time interval that the MN communication remains active. In

TABLE 2: Simulation parameters.

Parameter	Value
Total number of routers	17
Number of access routers (K)	2,4,6,8
Simulation scenario	$10 \times 10 \text{ km}^2$
Velocity of mobile users	1–20 m/s
Number of mobile users	200
Average rate of Poisson process ( $\lambda$ )	0.01
Duration of a session ( $\mu$ )	10
Flow rate of a request	1.5–10 Mbps
Confidence interval	95%
Number of iterations	500

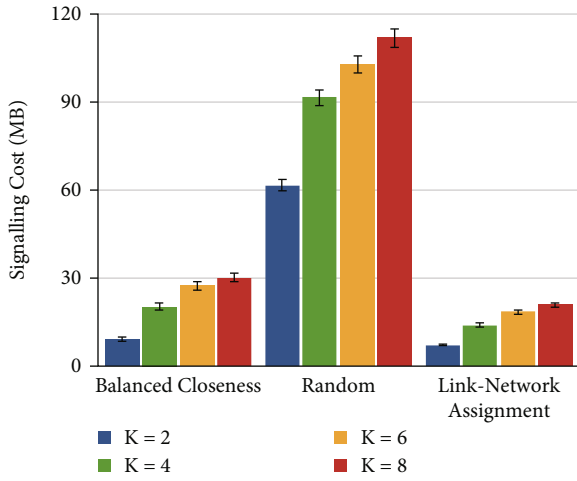


FIGURE 6: Control plane evaluation of centralized mobility protocol using different assignment algorithms.

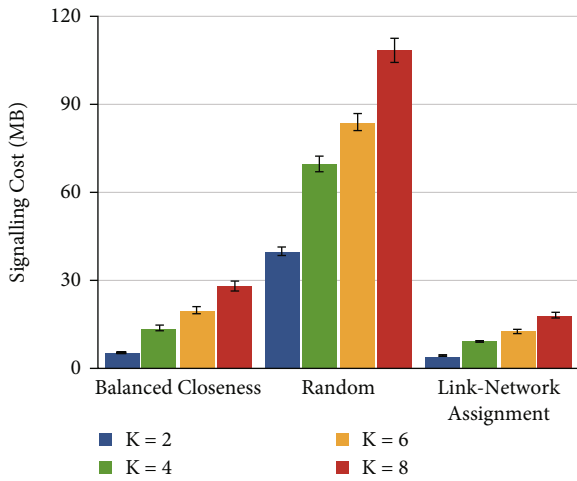


FIGURE 7: Control plane evaluation of distributed mobility protocol using different assignment algorithms.

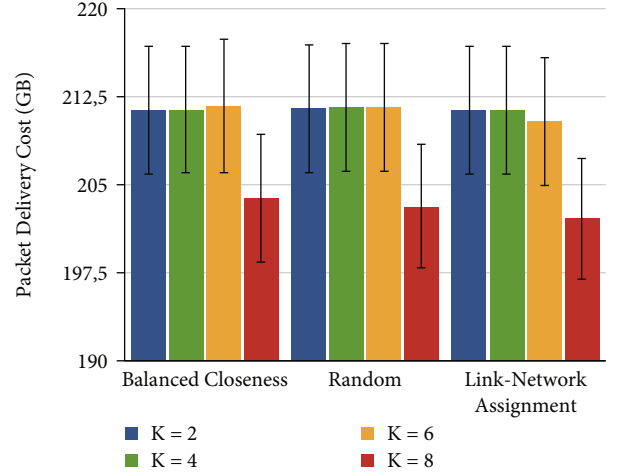


FIGURE 8: Data plane evaluation of centralized mobility protocol using different assignment algorithms.

PMIPv6 ( $C_u(PMIP)$ ), the registration update is needed with the mobility anchor (LMA). On the other hand, in DMM ( $C_u(NB-DMM)$ ), the serving MAG ( $SMAR$ ) retrieves information of the previous MAR's ( $PMAR_i$ ) with an active session, and it establishes IP-IP tunnels with them.

Hence, the following expressions represent the signaling cost for both solutions:

$$\begin{aligned}
 C_u(PMIP) &= 2s_u h_{MAG-LMA} \cdot \\
 C_u(NB-DMM) &= 2s_u + 2s_u \sum_{i=1}^{nActiveMAR-1} (h_{PMAR_i-SMAR}).
 \end{aligned} \tag{6}$$

Where  $s_u$  is the size of the PBU message and  $nActiveMAR$  is the number of MAR with an active session anchored for a particular MN.

**4.2. Data Plane.** Regarding the data plane, one of the metrics that have a major impact on the overall performance of the network is the packet delivery cost ( $P_{cost}(\cdot)$ ). Apart from the signaling related to the mobility management process, data packets have to be forwarded from the CN (Correspondent Node) to the MN and vice versa. In CMM solutions, the data is first routed to the centralized anchor, causing a suboptimal routing and a single point of failure in the network. In order to address these problems, new DMM solutions have been designed. Therefore, the packet delivery cost for a session is proportional to the size of the data messages and the number of hops needed to forward packets to the MN.

In PMIPv6, the packets are routed to the mobile user via the LMA through a tunnel that encapsulates the data packets. In NB-DMM protocols, when the handover process occurs, the traffic in the new location will be routed directly to the peer (direct mode), whereas the remaining connections will be tunneled to the user's corresponding anchoring

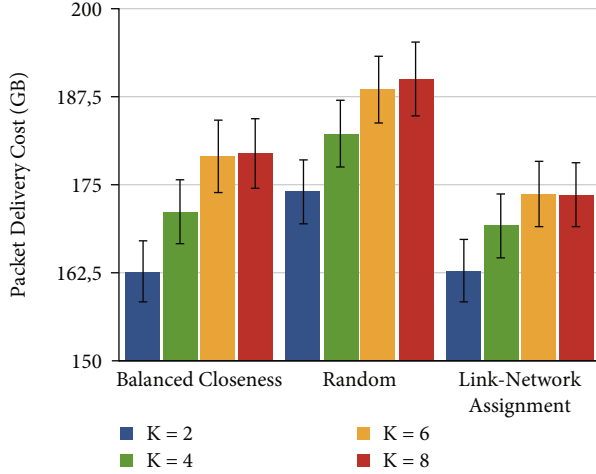


FIGURE 9: Data plane evaluation of distributed mobility protocol using different assignment algorithms.

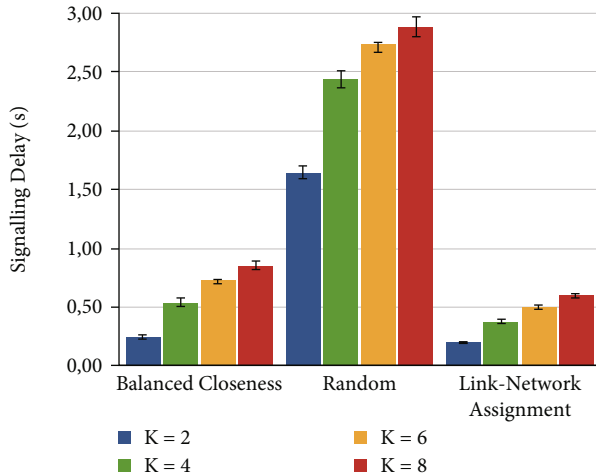


FIGURE 10: Average signalling delay evaluation of centralized mobility management protocol using different assignment algorithms.

MAR, and then routed to the destination peer (indirect mode).

Thus, the expressions that represent the cost are as follows:

$$P_{cost}(PMIP) = [s_d h_{CN-LMA} + (s_t + s_d) h_{LMA-MAG} + s_d h_{MAG-MN}] N_{p/s}$$

$$P_{cost}(NB-DMM) = [P_n P_{cost}(direct) + P_h P_{cost}(indirect)] N_{p/s} \quad (7)$$

Where  $N_{p/s}$  is the packet transmission rate per active flow,  $s_d$  is the size of these data messages and  $s_t$  is the average size of the tunnel header. Moreover,  $P_n$  and  $P_h$  are, respectively, the probabilities that the traffic is new or it is hand-over traffic.  $P_{cost}(direct)$  and  $P_{cost}(indirect)$  are the units of cost of delivering one packet in the direct and indirect

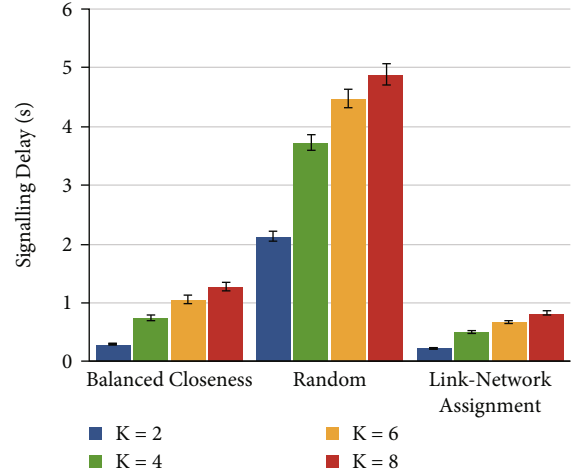


FIGURE 11: Average signalling delay of distributed mobility management protocol using different assignment algorithms.

modes of DMM, respectively. Then these costs are expressed as follows:

$$P_{cost}(direct) = s_d h_{CN-SMAR} + s_d h_{SMAR-MN}$$

$$P_{cost}(indirect) = (s_t + s_d) h_{PMAR-SMAR} + s_d h_{SMAR-MN} + s_d h_{CN-PMAR} \quad (8)$$

**4.3. Signaling Delay.** As seen before, the L3 handover requires a control message between different entities to maintain the established communications while the MN is moving across the network. These control messages introduce a delay in the communication that deteriorates the low-latency communications promoted in 5G. Assuming that packets are transmitted in a first-come-first-served manner, the signaling packets in the network can be transmitted only after all the packets before it has been transmitted. In this analysis, the propagation latency of the transmission medium is not considered.

Consequently, the signaling delay  $\delta(\cdot)$  for both solutions is summarized in the following expressions:

$$\delta(PMIP) = \frac{2s_u + (2s_u h_{MAG-LMA})}{B_w}, \quad (9)$$

$$\delta(NB-DMM) = \frac{2s_u + (2s_u n_{ActiveMAR})}{B_w},$$

being  $B_w$  is the mean bandwidth of the links.

## 5. Results

This section aims at providing insights into the impact of several mobility costs on the overall network performance and evaluating the proposed algorithm using different topologies. These topologies are based on city network topologies [24] by varying the number of access routers ( $K = \{2, 4, 6, 8\}$ ). Figure 5 shows the network topology with eight access

TABLE 3: Evaluation of mobility protocols using different assignment algorithms between base stations and access routers.

Centralized mobility protocol-PMIPv6					
Algorithm analysis		$ K  = 2$	$ K  = 4$	$ K  = 6$	$ K  = 8$
	$C_u(MB)$	$9.35 \pm 0.59$	$20.35 \pm 1.24$	$27.43 \pm 1.36$	$30.23 \pm 1.42$
Balanced closeness	$P_{cost}(MB)$	$211380.49 \pm 5401.64$	$211409.27 \pm 5403.60$	$211722.41 \pm 5632.99$	$203830.82 \pm 5437.97$
	$T_D(s)$	$12.47 \pm 0.78$	$27.14 \pm 1.65$	$36.55 \pm 1.78$	$42.78 \pm 1.78$
	$C_u(MB)$	$61.76 \pm 2.06$	$91.57 \pm 2.62$	$102.92 \pm 2.91$	$111.93 \pm 3.06$
Random	$P_{cost}(MB)$	$211518.15 \pm 5405.48$	$211596.41 \pm 5413.47$	$211638.96 \pm 5413.47$	$203178.27 \pm 5235.02$
	$T_D(s)$	$82.35 \pm 2.75$	$122.10 \pm 3.401$	$137.17 \pm 3.9$	$144.35 \pm 3.98$
	$C_u(MB)$	$7.46 \pm 0.40$	$14.21 \pm 0.52$	$18.74 \pm 0.65$	$21.03 \pm 0.70$
Link-network assignment	$P_{cost}(MB)$	$211375.93 \pm 5401.83$	$211393.59 \pm 5402.16$	$210442.95 \pm 5432.32$	$202162.96 \pm 5123.51$
	$T_D(s)$	$9.95 \pm 0.53$	$18.95 \pm 0.70$	$25.11 \pm 0.84$	$29.87 \pm 1.00$
Distributed mobility protocol-NB-DMM					
Algorithm analysis		$ K  = 2$	$ K  = 4$	$ K  = 6$	$ K  = 8$
	$C_u(MB)$	$5.60 \pm 0.39$	$13.84 \pm 0.94$	$19.88 \pm 1.18$	$28.27 \pm 1.59$
Balanced closeness	$P_{cost}(MB)$	$162737.46 \pm 4284.03$	$171150.97 \pm 4523.40$	$179118.25 \pm 5096.63$	$179529.07 \pm 4982.56$
	$T_D(s)$	$14.92 \pm 1.03$	$36.91 \pm 2.50$	$53.01 \pm 3.14$	$63.53 \pm 3.45$
	$C_u(MB)$	$39.97 \pm 1.49$	$69.88 \pm 2.58$	$83.96 \pm 2.99$	$108.53 \pm 4.14$
Random	$P_{cost}(MB)$	$174055.86 \pm 4540.51$	$182291.82 \pm 4728.63$	$188613.12 \pm 4729.96$	$189995.44 \pm 5231.66$
	$T_D(s)$	$106.60 \pm 3.97$	$186.35 \pm 6.88$	$223.89 \pm 7.98$	$244.27 \pm 9.15$
	$C_u(MB)$	$4.36 \pm 0.26$	$9.41 \pm 0.43$	$12.79 \pm 0.56$	$18.29 \pm 0.80$
Link-network assignment	$P_{cost}(MB)$	$162933.91 \pm 4390.05$	$169223.20 \pm 4490.39$	$173769.60 \pm 4661.60$	$173569.57 \pm 4547.01$
	$T_D(s)$	$11.62 \pm 0.69$	$25.08 \pm 1.15$	$34.09 \pm 1.43$	$41.31 \pm 1.76$

routers (R10–R17). The other topologies ( $K = \{2, 4, 6\}$ ) used in our simulations are based on this. For example, for  $K = 2$ , R11–R16 network nodes are removed from the access layer. Accordingly, for  $K = 4$ , the topology consists of R1–R11 and R16–R17 nodes. Finally, for  $K = 6$ , R13–R14 nodes are removed from the topology. Note that, each network node plays a determining role listed in Table 1.

Several topologies have been selected in order to provide more reliable results, avoiding the misleading performance of centralized or distributed protocols. The traffic and mobility parameters used in the simulations, as well as the numerical results of mobility costs, are presented next.

We run a MonteCarlo simulation of 500 iterations providing the average values and improving the accuracy of the results with a confidence interval of 95%. The proposed association algorithm is tested through simulations using Python with NetworkX and SciPy libraries [25], among others.

The simulation scenario is a square region of  $10 \times 10$  km<sup>2</sup> of area, where the base stations are distributed according to a Poisson Point Process (PPP) whose intensity ( $\lambda_{BS}$ ) coincides with the average number of BS ( $N_{BS}$ ) per unit area ( $A$ ) [26] and is obtained as  $\lambda_{BS} = N_{BS}/A$ . Moreover, the BS coverage areas are modeled as Poisson-Voronoi tessellation on the bidimensional plane where each mobile user is connected to the closest BS.

User mobility is defined by a Random Waypoint with a uniformly distributed velocity between 1 and 20 m/s. Each simulation consists of 200 mobile users who move across the mobility domain by connecting to different Base Stations. These mobile users manage a set of sessions during the simulation time. It is assumed that each mobile user receives incoming sessions following a Poisson process with an average rate of  $\lambda = 0.01$ . Moreover, the duration of a session is exponentially distributed with parameter  $\mu = 10$  [27]. It is also assumed that the flow rate requirement of a request varies from 1500 Kbps to 10 Mbps (e.g. video streams) [28]. The parameters that have been used in the simulations are presented in Table 2.

The performance of the proposed algorithm is evaluated over this network scenario by calculating the performance metrics analyzed before signaling ( $C_u(\cdot)$ ) and packet delivery cost ( $P_{cost}(\cdot)$ ). Moreover, to investigate how the link-network assignment affects the performance of the mobility management protocols, a set of simulations have been conducted over the aforementioned centralized and distributed mobility management protocols (PMIPv6 and NB-DMM, respectively) using different network topologies ( $K = \{2, 4, 6, 8\}$ ).

Figures 6 and 7 show the accumulated signaling cost for all connections generated during the simulation to provide a comparison of this metric as a function of the number of access routers for CMM and DMM protocols, respectively.

In these evaluations, the performance of the Link-Network Assignment algorithm proposed obtains better results compared with the other approaches analyzed, The random algorithm and the Balanced Closeness approach. The first one assigns each base station randomly to the access routers of the access network while the second one selects  $K$  base stations and computes the nearest  $N = \lfloor B \rfloor / \lfloor K \rfloor$  base stations to build  $\lfloor K \rfloor$  groups to associate the access routers.

The DMM approach locates the distributed mobility anchors closer to mobile users with the aim of generating a flatter network. These anchors are responsible for managing signaling traffic. Due to the distribution of the nodes, DMM reduces in 20%, on average, the traffic bottlenecks that affect to the CMM approaches, providing performance improvements in the control plane. Moreover, the signaling cost is directly proportional to the  $\lfloor K \rfloor$  value for both analyzed protocols. Thus, concerning the mobility management protocols, both solutions demonstrate a clear trend as the number of access routers increases. The proposed Link-Network Algorithm improves in both mobility management protocols reducing the signaling. In PMIP, the proposed algorithm reduces around 25% compared with Balanced Closeness and around 85% the random proposal. The last result is expected because the assignment of the base stations to the access router is a critical decision that can increase the signaling of the protocol, as shown in the analysis. In NB-DMM, the proposed algorithm reduces the signaling cost in around 30% compared with Balanced Closeness and around 86% with random assignment.

Figures 8 and 9 show the performance of the data plane. As could be observed, the packet delivery cost is greater in PMIP approach than in NB-DMM. This is produced because PMIP introduces the LMA that serves as an anchor to the MN. This produces suboptimal routing in the network and increase the packet delivery cost, and  $P_{cost}$  because a tunneling mechanism is introduced to forward data packets. In PMIP, the improvement applying the algorithm is not relevant due to this suboptimal routing, but in NB-DMM, the improvement of the proposed algorithm is around a 2% on average using Balanced Closeness and around a 7% with the random analysis. With these results, it can be concluded that the proposal, even improving the signaling of the mobility management protocols, can improve in a lesser extent the packet delivery protocol.

We can also see in Figures 8 and 9, when the number of access routers is increased, using the assignment algorithms based on clustering techniques, the  $P_{cost}$  begins to decrease with respect to lower  $\lfloor K \rfloor$  values.

Figures 10 and 11 show the impact of the signaling in the delay introduced in the network. As could be observed the average delay introduced by the signaling decrease using the proposed Link-Network Algorithm in both scenarios, using PMIP and NB-DMM.

The average bandwidth used to obtain this measurement is around 3 Mbps for each mobile node. Using PMIP as the mobility management protocol, the improvement is around 28% when the Balanced Closeness algorithm is used and

around 83% in the random proposal. As could be observed, this measurement is deeply connected with the signaling cost. In NB-DMM, the improvement is around 31% using the Balanced Closeness algorithm and around 85% using a random assignment.

All these numerical results are summarized in Table 3, which reflects the average and error of accumulated costs during all performed simulations. As shown in this Table 3, the overall delay introduced in the simulations is described. Consequently, with the results presented in Figures 10 and 11, the measurement coincides and presents a big delay introduced when the number of handovers is increased.

The results demonstrate that our proposed algorithm minimizes the impact on the total mobility cost and reduces the delay introduced by the signaling of the handovers. The benefits obtained are more significant in DMM when Link-Network assignment is used.

## 6. Conclusions and Future Works

This paper proposes a new way to improve the mobility management protocols that can affect positively to the latency of the network in 5G. The Link-Network Assignment Algorithm improves the mobility management protocols, centralized and distributed, reducing the signaling to manage the handovers, and maintain the reachability of the mobile node on the Internet. This mechanism also improves the packet delivery cost, especially in distributed mobility management protocols and the delay introduced by the signaling of the mobility management protocols, which will improve 5G's URLLC. The proposed algorithm has been compared favorably with others in terms of mobility costs (signaling cost, data packet delivery cost, and signaling delay), allowing the overall mobile network performance to be evaluated. Obtained results demonstrate that the LNA algorithm can successfully reduce the signaling cost by up to 86% compared with the baseline algorithm without penalizing the packet delivery cost that is also improved by up to 7%. This reduction in both metrics is one of the main contributions of this work. With these results, and taking into account, the expected increment of traffic expected for future mobile networks, our proposed mechanism offers significant gains for network operators to plan deployments for improving network performance for mobile users.

Future research in this direction would involve testing assignment algorithms based on other clustering mechanisms on different access network topologies. Moreover, some initialization techniques to find optimal centroids will need to be implemented in order to minimize the impact of base station assignment on the access network.

## Notations

$G$ :	Undirected graph of the access network
$V$ :	Set of network nodes
$E$ :	Set of network links
$B$ :	Set of base stations



$K$ :	Set of access routers. Each access router is denoted by $\{k_j\}_{j \in K}$
$N$ :	Set of mobile nodes which around the network
$x_{mr}^{ps}$ :	The binary decision variable equal to 1 if base station $m$ is assigned to the access router $r$ and $p$ is assigned to $s$ , 0 otherwise
$TC$ :	Sum of the signalling cost and packet delivery cost
$C_u(\cdot)$ :	Signalling cost
$P_{cost}(\cdot)$ :	Packet delivery cost
$\delta(\cdot)$ :	Signalling delay
$NT$ :	Network topology
$C$ :	Set of all centroids calculated by k-means++ algorithm from the set of base stations ( $B$ )
$D_{CKM}$ :	Matrix of distances between centroids
$D_{AR}$ :	The distance matrix of topology access nodes
$M_c$ :	This array stores the Euclidean norm of all centroids
$CC_{AR}$ :	This array stores the closeness centrality of each access routers
$MC$ :	The highest centroid modulus value
$cc$ :	The minimum value of closeness centrality
$d_c$ :	Distance vector that indicates the distance between $MC$ and all others
$d_{ar}$ :	Distance vector that stores the distance between $cc$ and all others
$DictAssoc$ :	Dictionary with the final relation between base stations and access routers
$h_{X-Y}$ :	Hop distance between $X$ and $Y$ nodes
$s_u$ :	Size of the Proxy Binding Update message
$nActiveMAR$ :	Number of Mobility Access Routers with an active session anchored for a particular mobile node
$s_d$ :	Size of the data messages
$s_t$ :	Size of the tunnel header
$N_{p/s}$ :	Packet transmission rate per active flow
$B_w$ :	The mean bandwidth of the links.

## Data Availability

The data supporting this work are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## Acknowledgments

This research was funded in part by the Spanish Ministry of Science and Innovation, grant number PID2020-112545RB-C54, the Regional Government of Extremadura, Spain, grant number GR21097, and the Regional Ministry of Economy and Infrastructure of the Junta de Extremadura under project IB18003. The authors are grateful to Research, Technological Innovation and Supercomputing Center of

Extremadura (CénitS) for allowing us to use their supercomputing facilities (LUSITANIA II).

## References

- [1] R. Atat, L. Liu, H. Chen, J. Wu, H. Li, and Y. Yi, "Enabling cyber-physical communication in 5G cellular networks: challenges, spatial spectrum sensing, and cyber-security," *IET Cyber-Physical Systems: Theory & Applications*, vol. 2, pp. 49–54, 2017.
- [2] A. Gupta and R. K. A. Jha, "Survey of 5G Network: architecture and emerging technologies," *IEEE Access*, vol. 3, pp. 1206–1232, 2015.
- [3] J. Navarro-Ortiz, P. Romero-Diaz, S. Sendra, P. Ameigeiras, J. J. Ramos-Munoz, and J. M. Lopez-Soler, "Survey on 5G usage scenarios and traffic models," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 905–929, 2020.
- [4] J. C. Zuniga, C. J. Bernardos, A. de la Oliva, T. Melia, R. Costa, and A. Reznik, "Distributed mobility management: a standards landscape," *IEEE Communications Magazine*, vol. 51, no. 3, pp. 80–87, 2013.
- [5] K. Kong, W. Lee, Y. Han, M. Shin, and H. You, "Mobility management for all-IP mobile networks: mobile IPv6 vs. proxy mobile IPv6," *IEEE Wireless Communications*, vol. 15, no. 2, pp. 36–45, 2008.
- [6] M. Balfaqih, Z. Balfaqih, V. Shepelev, S. A. Alharbi, and W. A. Jabbar, "An analytical framework for distributed and centralized mobility management protocols," *Journal of Ambient Intelligence and Humanized Computing*, 2020.
- [7] M. M. Sajjad, D. Jayalath, and C. J. Bernardos, "A comprehensive review of enhancements and prospects of fast handovers for mobile IPv6 protocol," *IEEE Access*, vol. 7, pp. 4948–4978, 2019.
- [8] S. Jeon, S. Figueiredo, R. L. Aguiar, and H. Choo, "Distributed mobility management for the future mobile networks," *IEEE Access*, vol. 5, pp. 11423–11436, 2017.
- [9] J. Carmona-Murillo, V. Friderikos, and J. L. González-Sánchez, "A hybrid DMM solution and trade-off analysis for future wireless networks," *Computer Networks*, vol. 133, 2018.
- [10] D. Chandramouli, R. Liebhart, and J. Pirskanen, *5G for the Connected World*, Wiley, 2019.
- [11] S. Kekki, W. Featherstone, Y. Fang et al., "MEC in 5G networks," *ETSI White Paper*, vol. 28, no. 28, pp. 1–28, 2018.
- [12] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, and B. Patil, *Proxy Mobile IPv6*, 2008.
- [13] E. A. Esquivel-Mendiola, H. Galeana-Zapién, and E. Aldana-Bobadilla, "Clustering-based approach to base station assignment in IoT cellular systems," *International Congress of Teleomatics and Computing*, vol. 1053, pp. 268–2283, 2019.
- [14] Z. Kaleem and K. Chang, "Public safety priority-based user association for load balancing and interference reduction in PS-LTE systems," *IEEE Access*, vol. 4, pp. 9775–9785, 2016.
- [15] M. Saimler and S. Coleri, "Multi-connectivity based uplink/downlink decoupled energy efficient user association in 5G heterogenous CRAN," *IEEE Communications Letters*, vol. 24, no. 4, pp. 858–862, 2020.
- [16] T. M. Shami, D. Grace, A. Burr, and J. S. Vardakas, "Load balancing and control with interference mitigation in 5G heterogeneous networks," *Journal on Wireless Communications and Networking*, vol. 2019, article 177, 2019.

- [17] K. Bakht, F. Jameel, Z. Ali et al., "Power allocation and user assignment scheme for beyond 5G heterogeneous networks," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 2472783, 11 pages, 2019.
- [18] N. Aljeri and A. Boukerche, "Mobility management in 5G-enabled vehicular networks: models, protocols, and classification," *ACM Comput. Survival*, vol. 53, no. 5, pp. 1–35, 2020.
- [19] D. Sattar and A. Matrawy, "Optimal slice allocation in 5G core networks," *IEEE Networking Letters*, vol. 1, no. 2, pp. 48–51, 2019.
- [20] N. Van Huynh, D. Thai Hoang, D. N. Nguyen, and E. Dutkiewicz, "Optimal and fast real-time resource slicing with deep dueling neural networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1455–1470, 2019.
- [21] Z. Kotulski, T. W. Nowak, M. Sepczuk, and M. A. Tunia, "5G networks: types of isolation and their parameters in RAN and CN slices," *Computer Networks, Volume*, vol. 171, 2020.
- [22] D. Arthur and S. Vassilvitskii, "The advantages of careful seeding," *Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms*, 2007.
- [23] C. Linton and Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, 1978.
- [24] G. Zheng, A. Tsiopoulos, and V. Friderikos, "Optimal VNF chains management for proactive caching," *IEEE Transactions on Wireless Communications*, vol. 17, no. 10, pp. 6735–6748, 2018.
- [25] A. Hagberg, D. Schult, and P. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proceedings of the 7th Python in Science Conference*, Pasadena United State, August 2008.
- [26] M. Di Renzo, A. Zappone, T. T. Lam, and M. Debbah, "System-level modeling and optimization of the energy efficiency in cellular networks—a stochastic geometry framework," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2539–2556, 2018.
- [27] M. K. Murtadha, N. K. Noordin, B. M. Ali, and F. Hashim, "Design and evaluation of distributed and dynamic mobility management approach based on PMIPv6 and MIH protocols," *Wireless Networks*, vol. 21, pp. 1–17, 2015.
- [28] G. Zheng, C. Wang, V. Friderikos, and M. Dohler, "High mobility multi modal e-health services," in *IEEE International Conference on Communications*, Kansas City, MO, USA, May 2018.

## Research Article

# Task Offloading and Scheduling Strategy for Intelligent Prosthesis in Mobile Edge Computing Environment

Ping Qi 

*Department of Mathematics and Computer Science, Tongling University, Tongling 244061, China*

Correspondence should be addressed to Ping Qi; [qiping929@tlu.edu.cn](mailto:qiping929@tlu.edu.cn)

Received 5 October 2021; Revised 19 October 2021; Accepted 29 November 2021; Published 7 January 2022

Academic Editor: Muhammad Shiraz

Copyright © 2022 Ping Qi. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Traditional intent recognition algorithms of intelligent prosthesis often use deep learning technology. However, deep learning's high accuracy comes at the expense of high computational and energy consumption requirements. Mobile edge computing is a viable solution to meet the high computation and real-time execution requirements of deep learning algorithm on mobile device. In this paper, we consider the computation offloading problem of multiple heterogeneous edge servers in intelligent prosthesis scenario. Firstly, we present the problem definition and the detail design of MEC-based task offloading model for deep neural network. Then, considering the mobility of amputees, the mobility-aware energy consumption model and latency model are proposed. By deploying the deep learning-based motion intent recognition algorithm on intelligent prosthesis in a real-world MEC environment, the effectiveness of the task offloading and scheduling strategy is demonstrated. The experimental results show that the proposed algorithms can always find the optimal task offloading and scheduling decision.

## 1. Introduction

According to statistics, the number of disabled people in China has topped 85 million in 2020 [1]. Lower limb amputation is a major cause of disability; millions of transfemoral amputees are suffering from difficulty in moving, which accounts for approximately seventy of the total number of disabled persons [1]. With the progress of science and technology, the scientists concentrate on the research and the maintenance of rehabilitation equipment, appliances, and other aids for disabled people. The intelligent prosthesis is the only way to compensate or restore the motor function, which can enable transfemoral amputees to perform diverse daily activities.

However, even though the movement of the lower limbs is the most basic human movement, many disabled people are still different to accomplish some simple tasks through the prosthetic leg. Meanwhile, using a passive prosthesis may significantly impair the walking symmetry and metabolic energy efficiency of transfemoral amputees. Therefore, the premise of using intelligent prosthesis is that some appropriate sensors and intent recognition algorithms

should be selected to obtain movement information. Then, the intelligent prosthesis can automatically calibrate the torque according to the analysis signals (such as biomechanical signal, surface electromyographic signal, and sEMG) perceived by sensors, which makes amputee's moving process more stable and natural [2].

As shown in Figure 1, the hierarchical control strategy of intelligent prosthesis is made up of three layers: perception layer, transformation layer, and execution layer. The perception layer recognizes amputee's motion intent by activity mode and context recognition. The transformation layer constantly adjusts the control strategy by comprehending human motion intent. Then, the above control strategy is used to actuate the intelligent prosthesis in execution layer. Obviously, as the motion intent recognition influences the effectiveness of the perception layer, the intent recognition algorithm with high performance and low latency is of the utmost importance.

Unfortunately, the traditional intent recognition methods often involve high complexity algorithms, such as template matching, convolutional neural network (CNN), and sensor fusion [3, 4]. Although there are many existing

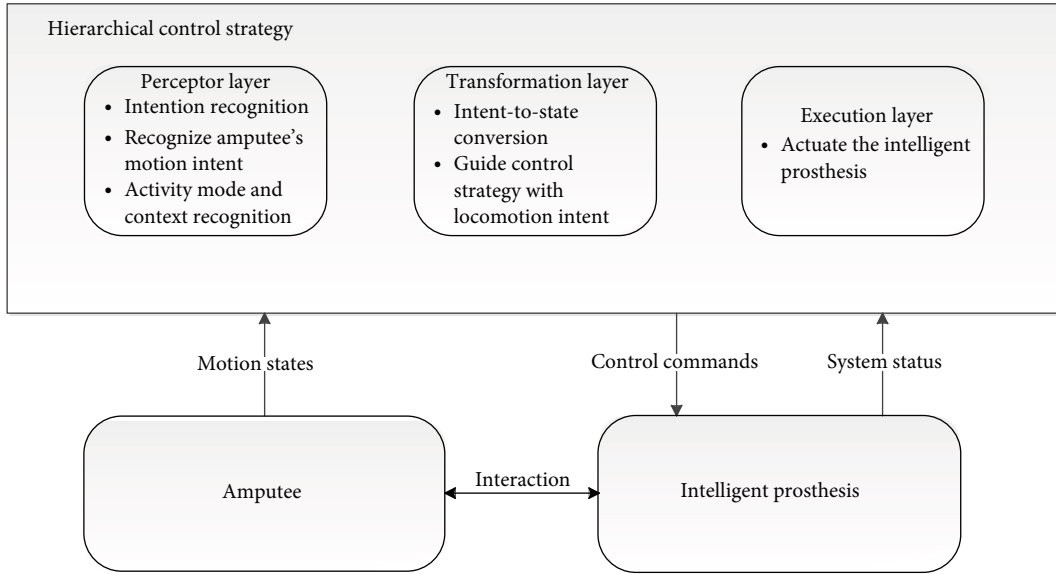


FIGURE 1: Hierarchical control strategy of intelligent prosthesis.

works focusing on reducing the computational complexity [5–9]. It is still difficult to achieve the purpose of real-time intent recognition. In the meantime, the computation and communication overheads of neural network model are affected by the model size, which bring a large amount of energy consumption and execution time. Therefore, the intelligent prosthesis, which is limited by their computing power and battery capacity, requires intensive computation and energy resources to provide services.

Recently, mobile edge computing (MEC) has become a promising solution which supports computation-intensive tasks. MEC is an efficient method to overcome the challenge by offloading some latency-sensitive tasks or computation-intensive to nearby edge servers through wireless communication [10, 11]. Then, the edge servers execute some of the received computation tasks and transmit the rest to resource-rich cloud infrastructures by low-latency connection. At last, the edge servers or the cloud server transmit the computation results to the mobile device. In this paper, the intelligent prosthesis can be represented as the mobile device in an MEC system.

To take full advantage of the mobile edge computing, an effective collaboration between the intelligent prosthesis, the edge servers, and the cloud is an essential problem. In this paper, we focus on solving latency and energy-constrained challenges, in which mobile processors share multiple heterogeneous edge servers. By deploying the deep learning-based motion intent recognition algorithm on the intelligent prosthesis in MEC environment, we demonstrate the effectiveness of the proposed task offloading strategy in reducing the latency and energy consumption.

The structure of this paper is as follows: Section 2 describes the related works on the intent recognition algorithm for intelligent prosthesis and some existing studies for applying AI technology in the MEC environment. MEC-based offloading model and problem definition are given in Section 3. The proposed algorithms are described

in detail in Section 4. The experimental results are discussed in Section 5. Finally, the conclusion of this paper is provided in Section 6.

## 2. Related Work

The motion states, which are conducted by the lower limb, have an inherent regularity. One single gait cycle includes two phases: a stance phase and a swing phase. As shown in Figure 2, the swing phase consists of a steady state and a transitional state. The steady state begins with a toe-off and ends with a heel strike. The transitional state also begins with a toe-off. Then, the foot rises from the flat ground, and the transitional state ends when the heel touches the stair or the ramp [12]. Various pattern recognition and machine learning algorithms are used to analyze the regularity, locomotion modes, and transition modes within a single gait cycle.

Huang et al. [5] propose a motion intent recognition algorithm based on neuromuscular-mechanical fusion. Electromyographic (EMG) signals and ground reaction forces are used as the input data to a phase-dependent pattern classifier. In this study, six locomotion modes and five transitions can be recognized, and the recognition accuracy reaches 95.2%. Liu et al. [6] study the effectiveness of applying three different adaptive pattern classifiers based on surface electromyography and mechanical sensors. Under a variety of different terrains, the proposed algorithm predicts amputee's motion intent with a rate of 95.8%. During recent years, CNN has been widely used in different smart systems, such as smart health, smart logistics, and smart agriculture. Su et al. [7] put the inertial measurement units (IMUs) on the healthy leg of amputees. They design a CNN structure to automatically learn the features from the sensor signals without any prior knowledge. Idowu et al. [8] present an integrated deep learning model (deep neural networks, DNN) for motor intention recognition of multiclass signals.

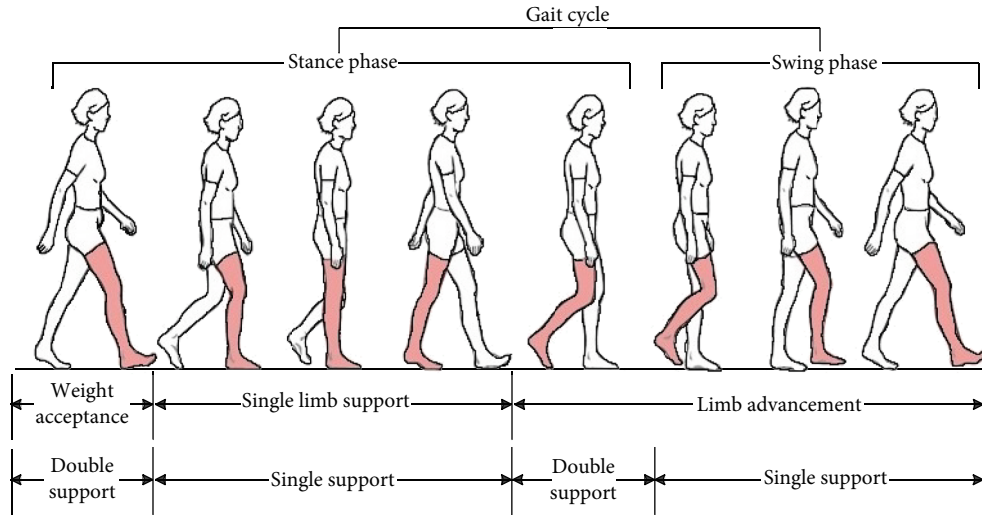


FIGURE 2: Illustration of stance phase and swing phase.

In this paper, the related features with different frequencies are introduced as input representation to the proposed deep learning model, and the recognition accuracy can reach a high level (average performance of 99.01%). Gautam et al. [9] propose a low-complex deep learning framework for sEMG-based movement recognition. The method of two-stage pipeline compression (input data compression and data-driven weight sharing) is designed to reduce the number of memories, energy consumption, and execution time.

Deep learning inference and training require substantial computation resources to run quickly; a common approach is to leverage cloud computing [13, 14]. However, sending data to the cloud for inference or training may incur additional queuing and propagation delays from the network and cannot satisfy strict end-to-end low-latency requirements. Cloud-based intent recognition algorithms need to be further modified to run on computation constrained devices, such as a Raspberry Pi or an Arduino.

In summary, the previous studies usually use mechanical, sEMG, or IMU sensors for data collection, which are embedded on the intelligent prosthesis or the healthy leg. Then, various machine learning algorithms, such as hidden Markov model, SVM, or deep learning, are selected as the classifier to recognize the motion intent of the amputees. In particular, the deep-learning-based intent recognition algorithms have the distinct advantage of executing the data-driven feature extraction without any prefeature extraction. However, the deep learning algorithm's high accuracy and real-time implementation for the intelligent prosthesis comes at the expense of high computational, memory, and energy consumption requirements for both the training and inference phases.

To meet the computational requirements of deep learning, edge computing is a viable solution to meet the latency and energy consumption challenges. There are many existing works focusing on applying deep learning technology in the MEC environment. Li et al. [15] present a deep learning model coinference framework. To reduce the execution time, they optimize the CNN partitioning and right-sizing

based on the available bandwidth and the on-demand manner. Li et al. [16] propose a joint accuracy-and latency-aware execution framework named JALAD to minimize the edge computation latency, cloud computation latency, and transmission latency. Gao et al. [17] propose the Edge4Sys system to reduce the computation load of the edge server in a MEC-based UAV (Unmanned aerial vehicle) delivery scenario. Tariq et al. [18] present the FogNetSim++ environment, which covers the network aspects such as latency, transmission range, scheduling, mobility, and heterogeneous mobile devices. Asad et al. [19] propose a fog simulation framework named xFogSim to support latency-sensitive applications. It has a very efficient task distribution algorithm that can choose the best computation resource depending on the cost, availability, and latency. Syed et al. [20] present a fog computing framework to simulate the vehicle-assisted computing environment, which allow researchers to incorporate their own scheduling policies to simulate a realistic environment. Table 1 lists the comparison of these works.

However, it is not suitable for applying the above works directly to the intelligent prosthesis scenario. One major challenge is that the above works do not consider the mobility of the mobile device, which is an important property of the intelligent prosthesis. The second challenge is accommodating the high resource requirements of deep learning on less powerful intelligent prosthesis with mobile processors. Most offloading and scheduling strategy still transmit a large amount of data to the remote servers. The data processing time by deep learning model is large; the cooperation between the mobile device and the remote servers should be employed in a real-world MEC-based system to reduce the energy consumption and latency.

### 3. MEC-Based Task Offloading Model

The motion intent recognition algorithm need to be quickly processed to detect and return a response. However, the complexity of computing tasks brings a big burden to the mobile processors which are limited by their computation

TABLE 1: Comparison of existing work on edge computing.

Research	Proposed solutions	Key metrics	Edge devices	What is to be offloaded
Li et al. [15]	A deep learning model coinference framework	Latency, communication size	Devices with cameras	Computer vision algorithms
Li et al. [16]	A joint accuracy-and latency-aware execution framework	Latency, accuracy	Devices with cameras	Computer vision algorithms
Gao et al. [17]	Edge4Sys system to reduce the computation load of the edge server in a MEC-based UAV delivery scenario	Latency, energy, accuracy	UAV	DNN-based feature extraction and classification
Tariq et al. [18]	A fog simulator, covers the network, transmission range, heterogeneous mobile devices, and mobility feature	Latency, transmission range, mobility	IoT devices	Typical IoT tasks
Asad et al. [19]	A fog simulation framework to support latency-sensitive applications	Latency, energy, accuracy	IoT devices	Typical IoT tasks
Syed et al. [20]	A fog computing framework to simulate the vehicle-assisted computing environment	Latency, energy consumption, communication size, memory	Vehicles	Compute-intensive tasks
The proposed algorithm	MEC-based system to reduce the latency and energy consumption	Latency, energy consumption	Intelligent prosthesis	DNN-based intent recognition tasks

resource and battery capacity. Meanwhile, sending data to the cloud for inference or training may incur propagation delays from the network. This method cannot satisfy real-time requirements of applications because of the unsteady character of the mobile network.

In the mobile edge computing environment, the edge devices provide computation abilities close to the mobile devices. Unfortunately, this potential solution of moving the computation and data from the mobile devices to the edge still has its limitations. While edge's computation resources are substantial, they are also limited when compared to the cloud. Therefore, the edge servers should coordinate with the mobile device, the cloud, and other edge servers to ensure a good performance. Each task will be decided to be processed locally or offloaded to the cloud or the edge.

As can be seen from Figure 3, we present the detail design of the MEC-based task offloading model for deep neural networks in the intelligent prosthesis scenario. The execution process is mainly divided by the following steps: (1) the sensor data are preprocessed by the mobile processors. We deploy the intent recognition algorithm [7] on the intelligent prosthesis which is designed specifically for embedded devices. (2) CNN model partitioning. As shown in Figure 4, the CNN is partitioned into layers, some layers are executed on the mobile processors, and some layers are offloaded to the edge or the cloud according to the task offloading and scheduling strategy; (3) when the amputees are moving, mobility-aware task offloading strategy (such as task migration and task deferred execution) is considered to guarantee the service continuity; (4) the computation results are transmitted to intelligent prosthesis.

**3.1. System Model.** In this section, we consider the computation offloading problem of multiple heterogeneous edge servers in MEC environment. The original data, which is

preprocessed by the mobile processor, is continuously offloaded to the edge servers and the cloud based on the task offloading strategy. As can be seen in Figure 3, there are  $m$  edge servers,  $S = \{S_1, S_2, \dots, S_m\}$ , which have been deployed in area  $A$ . The intelligent prosthesis (mobile device) in use by the amputee is walking through area  $A$ , the moving path can be represented as a set of continuous position coordinates,  $\{c_1, c_2, \dots, c_i, \dots\}$ ,  $c_i = (x_i, y_i)$ , where  $c_i$  is the  $i$ th position coordinates of the moving path,  $x_i$  and  $y_i$  are defined as the  $x$ -coordinate and  $y$ -coordinate of  $c_i$ , respectively. We assume that the moving path is predetermined, and the speed is  $v$  (m/s).

The edge server  $S_i$  can be represented by a quadruple,  $S_i = \{f_i, BW_i, Loc_i, Dis_i\}$ , where  $f_i$ ,  $Loc_i$ ,  $BW_i$ ,  $Dis_i$  are used to indicate the computational capability, position coordinate, transmission bandwidth, and maximum communication range of  $S_i$ , respectively. For each MEC server, when the distance between a mobile device and  $S_i$  is more than the maximum communication range  $Dis_i$ , they can not communicate with each other. Due to the amputees are walking round in area  $A$ , increasing distance between the mobile device and the edge server will decrease the communication rate. The instantaneous transmission rate between  $S_i$  and the mobile device can be calculated by

$$R_{\text{ins}} = BW_i \log_2(1 + f_{\text{SNR}}(d)), d \leq Dis_i, \quad (1)$$

where  $R_{\text{ins}}$  is the instantaneous transmission rate,  $d$  is the distance between  $S_i$  and the mobile device, and  $f_{\text{SNR}}(d)$  is the signal-to-noise ratio (SNR) [21]. The mobile device can be represented by a triple,  $MD = \{f_m, P, Loc_m\}$ , where  $f_m$  is the computational capability of the mobile processor,  $P$  is the energy consumption, and  $Loc_m$  is the position coordinate of the intelligent prosthesis. The cloud server is represented by a binary group,  $CS = \{f_c, R_c\}$ , where  $f_c$  is the

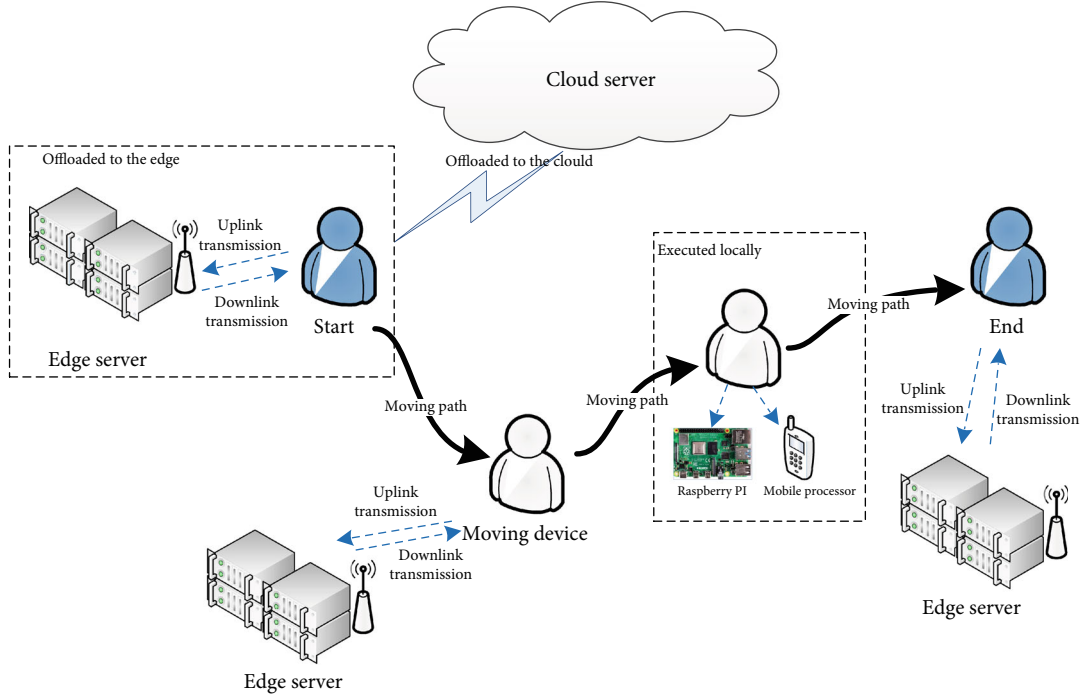


FIGURE 3: Overview of the MEC-based task offloading model.

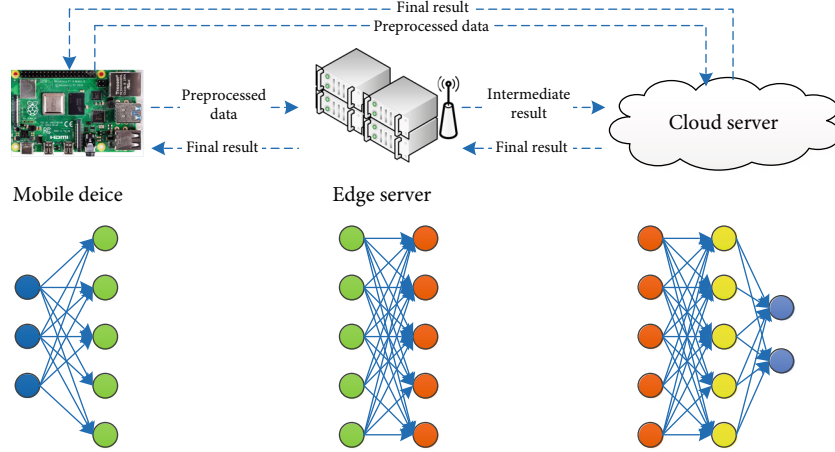


FIGURE 4: Deep learning inference with edge computing.

computational capability of the cloud server and  $R_c$  is the maximum data transfer rate.

In the computation offloading process, task execution spans the mobile device, the edge servers, and the cloud. The original data should be transmitted to the remote servers, and the execution results will be transmitted back to the mobile device. It is a tradeoff between the benefit of remote execution and the cost of data transmission. When the computation task is performed locally, the execution time and the energy consumption are only generated by the mobile device. Therefore, computation offloading happens only if the time and energy consumption of task offloading is better than the local execution. The relevant definitions are given below. Suppose that the uplink transmission power, downlink transmission power, idle power, and execution power of the mobile device be  $p_{up}$ ,  $p_{down}$ ,  $p_{idle}$ , and  $p_{exec}$ , respectively. Accordingly, the energy consumption can be represented by a quadruple,  $P = \{p_{up}, p_{down}, p_{idle}, p_{exec}\}$ .

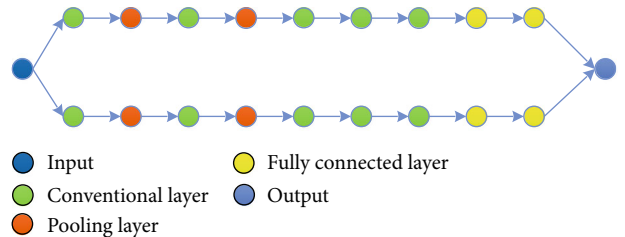


FIGURE 5: Workflow instance and abstract workflow DAG graph.

TABLE 2: Summary of key symbols in Section 3.

Symbol	Description
$T$	$T = \{t_1, \dots, t_i\}$ is the set of tasks in mobile device.
$IN_i, OUT_i$	The input and output data size of task $t_i$ .
$l_i$	The CPU cycles frequency that is required to process task $t_i$
$f_c, f_m, f_i$	The computational capability of the cloud server, mobile device, and edge server $S_i$
$P_{tra}, P_{rec}, P_{idle}, P_{exec}$	The transmitting, receiving, idle, and execution powers of mobile device
$Loc_i, BW_i, Dis_i$	The position coordinate, transmission bandwidth, and maximum communication range of edge server $S_i$
$Loc_m$	The position coordinate of the intelligent prosthesis
$f_{SNR}(d)$	Signal-to-noise ratio (SNR)
$R_{ins}$	Instantaneous transmission rate between edge server and mobile device
$T_{es}, T_c, T_m$	The total execution time of the edge servers, cloud server, and mobile device.
$E_{es}, E_c, E_m$	The total energy consumption when the tasks are offloaded to the edge servers, cloud server, or computed locally.
$R_{migr}$	The instantaneous transmission rate between two edge servers.

The most important step toward computation offloading is partitioning, which divides the motion intent recognition algorithm into several parts that can be performed on different platforms. As shown in Figure 5, according to the structure of CNN [7], the neural network is partitioned into several layers. One layer, which can only be computed by one computation resource, is defined as the metatask. The metatask cannot be partitioned, and the set of metatasks can be represented by a directed acyclic graph (DAG),  $DAG = (T, E)$ , where  $T$  is a set of nodes,  $T = \{t_i | t_i = (IN_i, OUT_i, l_i)\}$ .  $IN_i$ ,  $OUT_i$ , and  $l_i$  are used to indicate the input data size, output data size, and CPU cycle frequency of task  $t_i$ , respectively.  $E$  is a set of edges,  $E = \{(t_{pre}, t_{succ}) | t_{pre}, t_{succ} \in T\}$ , where  $t_{pre}$  is the predecessor task and  $t_{succ}$  is the successor task. The topology relationships between different tasks can be described by  $E$ .

As can be seen in Figure 5, the convolutional layer and fully connected layer are the most commonly used neural network layers. Their computation load can be calculated by

$$l_c = (2 \times C_{in} \times KH \times KW [-1]) \times H_{out} \times W_{out} \times C_{out}, \quad (2)$$

$$l_{FC} = (2 \times I [-1]) \times O, \quad (3)$$

where  $H_{out}$  and  $W_{out}$  are defined as the output feature map size and  $C_{out}$  and  $C_{in}$  are the number of output channels and input channels.  $KH$  and  $KW$  are the size of the convolution kernel.  $I$  and  $O$  are the number of input and output neurons.

Summary of key symbols used in this section can be found in Table 2.

### 3.2. Energy Consumption and Latency Model

**3.2.1. Local Execution Cost.** When the task  $t_i$  is performed locally, the energy consumption is generated by the execution processes. We assume that  $n_m$  is the total number of tasks performed by the mobile device; the total execution time  $T_m$  and energy consumption  $E_m$  can be estimated by

$$\begin{cases} T_m = \sum_{i=1}^{n_m} \frac{l_i}{f_m}, \\ E_m = P_{exec} \times \sum_{i=1}^{n_m} \frac{l_i}{f_m}. \end{cases} \quad (4)$$

**3.2.2. Offloading Cost.** When the task  $t_i$  is offloaded to the cloud server, the latency  $T_c$  is consisted by the transmission latency and the execution latency, and the energy consumption  $E_c$  is consisted by the transmission energy and the execution energy. Suppose that  $n_c$  is the total number of tasks offloaded to the cloud,  $T_c$  and  $E_c$  can be calculated by

$$\begin{cases} T_c = T_{up} + T_{down} + T_{exec} = \sum_{i=1}^{n_c} \frac{IN_i}{R_c} + \sum_{i=1}^{n_c} \frac{OUT_i}{R_c} + \sum_{i=1}^{n_c} \frac{l_i}{f_c}, \\ E_c = E_{up} + E_{down} + E_{idle} = P_{tra} \times \sum_{i=1}^{n_c} \frac{IN_k}{R_c} + P_{rec} \times \sum_{i=1}^{n_c} \frac{OUT_k}{R_c} + P_{idle} \times \sum_{i=1}^{n_c} \frac{l_i}{f_c}, \end{cases} \quad (5)$$

where  $T_{up}$ ,  $T_{down}$ , and  $T_{idle}$  are defined as the uplink transmission latency, downlink transmission latency, and execution latency, respectively.  $E_{up}$  and  $E_{down}$  are defined as the uplink transmission energy and the downlink transmission energy.  $E_{idle}$  is the idle energy consumption of the mobile device when the task is computed by the cloud server or edge server.

When the mobile device offloads computation to the edge, it is important to guarantee the service continuity. However, as motioned above, increasing distance between the mobile device and the edge server will decrease the communication rate. If the amputee is moving out of the communication range of the edge server, the task offloading will be failed. As shown in Figure 6, there exist four kinds of situations: normal offloading, offloading failure, task migration, and task deferred execution.



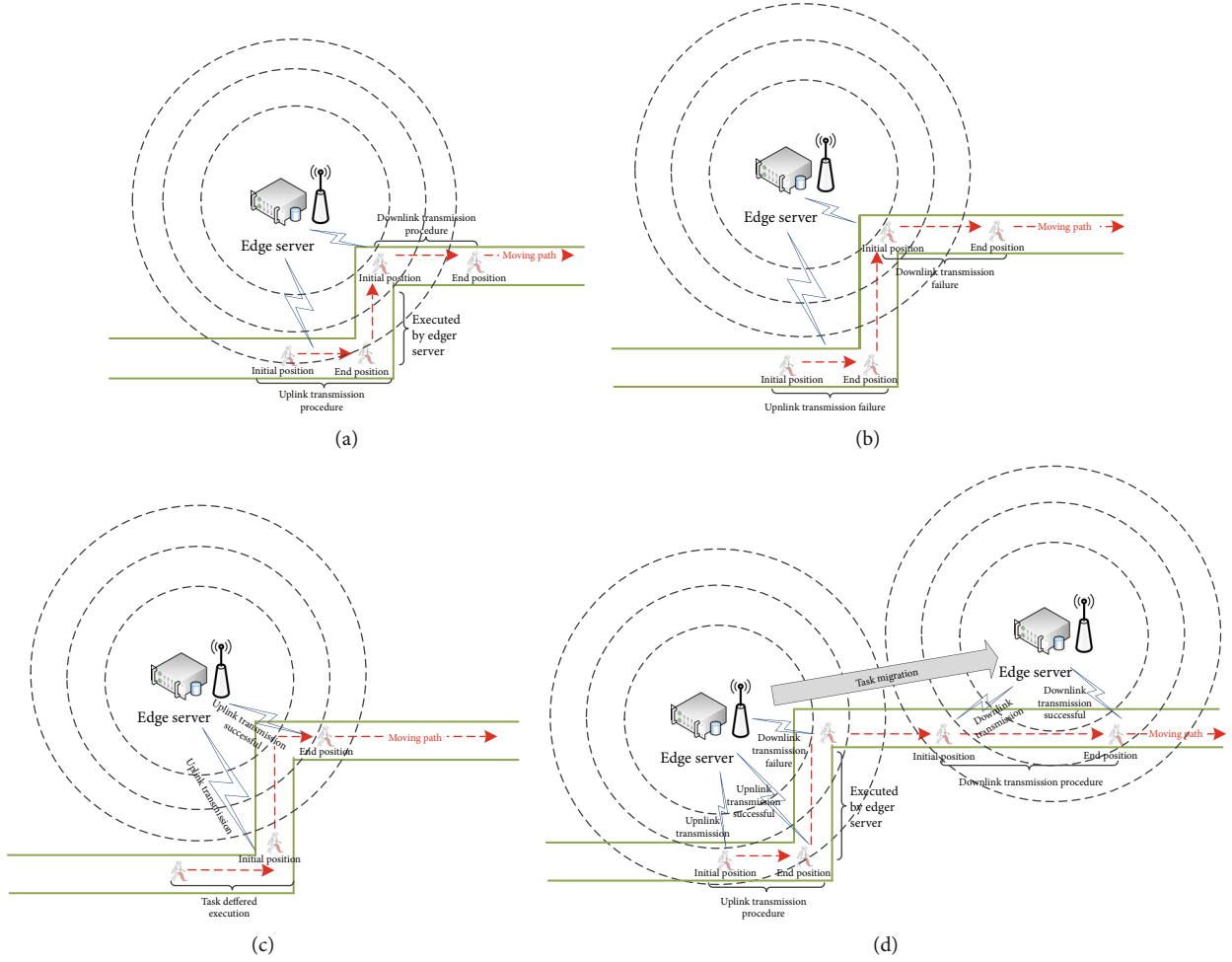


FIGURE 6: Four kinds of situations: normal offloading, offloading failure, task migration, and task deferred execution.

**3.2.3. Normal Offloading.** When the amputee is in the communication range of the edge server, the task can be normally offloaded. As shown in Figure 6(a), the amputee is moving from the initial position (denoted by  $c_{\text{inti}}$ ) to the end position (denoted by  $c_{\text{end}}$ ) in the uplink transmission procedure of task  $t_i$ . Suppose that  $R(S_j, MD)$  is the instantaneous transmission rate between the edge server  $S_j$  and the mobile device,  $d(Loc_j, Loc_m)$  is the distance between them. We assume that the instantaneous transmission rate remains unchanged when the moving distance is no more than one meter. Therefore, the size of data transferred within one meter can be calculated by

$$\text{Data}_{\text{meter}} = R(S_j, MD) \times \frac{1}{v}, \quad (6)$$

where  $R(S_j, MD)$  can be calculated by Formula (1).

When the summation of the transmitted data (denoted by  $\text{Data}_{\text{tra}}$ ) is equal to the input data size of  $t_i$ , the uplink transmission procedure is finished. Then, the distance  $d(c_{\text{inti}}, c_{\text{end}})$  between  $c_{\text{inti}}$  and  $c_{\text{end}}$  can be calculated by

$$\begin{cases} \text{Data}_{\text{tra}} = \sum_{k=1}^{d(c_{\text{inti}}, c_{\text{end}})} \text{Data}_{\text{meter}}, \\ \arg \min_{d(c_{\text{inti}}, c_{\text{end}})} |\text{Data}_{\text{tra}} - \text{IN}_i|, \text{Data}_{\text{tra}} \geq \text{IN}_i. \end{cases} \quad (7)$$

Accordingly, the end position of the mobile device can be estimated along the preset path according to the initial position. Similarly, in the downlink transmission procedure of task  $t_i$ , the transmission time and the end position can be calculated by the same way.

**3.2.4. Offloading Failure.** As shown in Figure 6(b), there is no edge server satisfying the communication conditions along the moving path. The task cannot be offloaded to the edge.

**3.2.5. Task Deferred Execution.** As shown in Figure 6(c), the task cannot be offloaded to the edge server in the uplink transmission procedure. However, there are several servers satisfying the communication conditions along the moving path. ‘‘Task deferred execution’’ is used to wait for the nearest edge server along the moving path.  $T_{\text{delay}}$  is the deferred time.

**3.2.6. Task Migration.** As shown in Figure 6(d), although the mobile device cannot connect to the edge in the downlink transmission procedure, there are several servers satisfying the communication conditions along the moving path. The computation results should be migrated from the current edge server to the target edge server to make sure the task execution result will be delivered successfully back to the mobile device. Suppose that the computation capability of the current edge server and the target edge server are  $f_{\text{curr}}$  and  $f_{\text{tgt}}$ ,  $R_{\text{migr}}$  is the transmission rate between them. Then, the task  $t_i$  is performed by the current edge server, and the computation results  $\text{OUT}_i$  is migrated to the target edge server. The migration time is  $T_{\text{migr}} = \text{OUT}_i / R_{\text{migr}}$ .

In conclusion, when the mobile device offloads the task  $t_i$  to the edge, the latency  $T_{\text{es}}$  is consisted by the transmission latency (including uplink transmission time and downlink transmission time), execution latency, migration latency, and deferred execution latency. Suppose that  $n_{\text{es}}$  is the total number of tasks offloaded to the edge,  $T_{\text{es}}$  and  $E_{\text{es}}$  can be calculated by

$$\begin{cases} T_{\text{es}} = \sum_{i=1}^{n_{\text{es}}} T_{\text{up}} + \sum_{i=1}^{n_{\text{es}}} T_{\text{down}} + \sum_{i=1}^{n_{\text{es}}} T_{\text{delay}} + \sum_{i=1}^{n_{\text{es}}} T_{\text{migr}} + \sum_{i=1}^{n_{\text{es}}} T_{\text{exec}}, \\ E_{\text{es}} = p_{\text{up}} \times \sum_{i=1}^{n_{\text{es}}} T_{\text{up}} + p_{\text{down}} \times \sum_{i=1}^{n_{\text{es}}} T_{\text{down}} + p_{\text{idle}} \times \left( \sum_{i=1}^{n_{\text{es}}} T_{\text{delay}} + \sum_{i=1}^{n_{\text{es}}} T_{\text{migr}} + \sum_{i=1}^{n_{\text{es}}} T_{\text{exec}} \right), \end{cases} \quad (8)$$

where  $T_{\text{up}}$ ,  $T_{\text{down}}$ ,  $T_{\text{exec}}$ ,  $T_{\text{delay}}$ , and  $T_{\text{migr}}$  are defined as the uplink transmission time, downlink transmission time, execution time, migration time, and deferred time (if the task can be normally offloaded,  $T_{\text{delay}} = 0$ ,  $T_{\text{migr}} = 0$ ).  $T_{\text{up}}$ ,  $T_{\text{down}}$ , and  $T_{\text{exec}}$  can be calculated according to the initial location of the mobile device and the current edge server.

## 4. Proposed Algorithms

To reduce the latency and energy consumption, we propose algorithms to determine whether the methods to be executed remotely or locally with deadline constraint of each task. The proposed algorithms can be divided into three parts as follows:

- (1) CNN-based intent recognition algorithm is partitioned into several layers, which can be expressed as a workflow task. According to the priority queue of the workflow, we construct a matrix of task execution as follows:

$$Q = \begin{bmatrix} q_1^1 & q_2^1 & q_3^1 \\ q_1^2 & q_2^2 & q_3^2 \\ \dots & \dots & \dots \\ q_1^n & q_2^n & q_3^n \end{bmatrix}, \quad (9)$$

$$\begin{cases} q_1^i + q_2^i + q_3^i = 1, \quad i \in \{1, 2, \dots, n\}, \\ q_1^i, q_2^i, q_3^i \in \{0, 1\} \end{cases}$$

where  $n$  is the total number of tasks.  $q_j^i$  is an execution vector of task  $t_i$ ,  $j = 1, 2, 3$  denote that  $t_i$  is executed locally or offloaded to the edge or the cloud, respectively. Accordingly, the matrix of task execution is regarded as the task scheduling plan

- (2) When the task  $t_i$  is scheduled to be offloaded to the edge according to the execution vector ( $q_2^i=1, q_1^i=0, q_3^i=0$ ), the optimal edge server should be selected by selection optimization algorithm (SOA) based on the location and the moving path of the mobile device
- (3) Based on the models presented in Section 3, a novel task offloading and scheduling strategy (TOSS) is proposed to optimize the whole scheduling process from a global viewpoint. After all offloading decisions are made, workflow scheduling is conducted for all types of resources allocated in the MEC environment

**4.1. Selection Optimization Algorithm.** In this subsection, we present a selection optimization algorithm. Once the task  $t_i$  is scheduled to be offloaded to the edge, the selection optimization algorithm (SOA algorithm) is used to screen out the optimal offloading edge server according to the energy consumption, location, moving path, and velocity of mobile device.

According to the location and velocity of intelligent prosthesis, SOA algorithm is used to screen out the optimal offloading edge server within the maximum communication range of edge server. The inner loop of SOA algorithm is based on the greedy algorithm. Let the number of elements in candidate edge server set be  $D$ ; the computation complexity of SOA algorithm is  $O(D)$ .

**4.2. Task Offloading and Scheduling Strategy.** In this section, we propose a task offloading and scheduling strategy based on particle swarm optimization algorithm (PSO).

**4.2.1. Fitness Value.** The fitness value is designed to evaluate the impact of offloading decision, which can calculate the latency and energy consumption of the mobile device. The smaller fitness value is regarded as the lower energy consumption of the task offloading and scheduling strategy. According to the energy consumption and latency model proposed in Section 3, we construct the fitness function by

$$\begin{cases} T_{\text{sum}} = T_m + T_{\text{es}} + T_c, \\ E_{\text{sum}} = E_m + E_{\text{es}} + E_c, \end{cases} \quad (10)$$

$$\text{fitness} = (f_1 \times E_{\text{sum}}) + \left( f_2 \times 10 \times E_{\text{sum}} \times \frac{T_{\text{sum}}}{T_{\text{respond}}} \right), \quad (11)$$

where  $E_{\text{sum}}$  is the total energy consumption,  $T_{\text{sum}}$  is the latency of the workflow, and  $T_{\text{respond}}$  is the deadline constraint. According to the motor coordination study of

```

Input  $t_i = (IN_i, OUT_i, l_i)$ ,  $t_i \in T$ ; a candidate set of edge servers  $S$ ; mobile device MD;
Output Optimal edge server(  $S_{opti}$ );
1 For each  $S_j \in S$ 
2 {      If( $d(Loc_j, Loc_m) < Dis_j$ )
3           $ESset \leftarrow S_j$ ; //Initialize  $ESset$ 
4 } //End For each  $S_j \in S$ 
5 If ( $ESset = NULL$ )
6 {      Calculate the deferred execution time;
7          Update the position coordinates of the mobile device by Formula (7);
8          Return  $S_{opti}$ ;
9 } //End if
10 For each  $S_j \in ESset$ 
11 {      Update the position coordinate of the mobile device according to the initial position coordinate and the moving path
by Formula (6) and Formula (7)
12          If ( $d(Loc_j, Loc_{end}) > Dis_j$ )
13              { Calculate the migration time  $T_{migr}$ ; //  $T_{migr} = OUT_i / R_{migr}$ 
14              } //End if
15              Calculate  $T_{es}$  and  $E_{es}$  by Formula (8);
16              Update the minimum energy consumption;
17 } //End For each  $S_j \in ESset$ 
18 Update the position coordinates of the mobile device;
19 Return  $S_{opti}$ ;

```

ALGORITHM 1: Selection Optimization Algorithm (SOA).

```

Input Workflow  $T = \{t_1, t_2, \dots, t_n\}$ ; matrix of task execution  $Q$ ; a set of edge servers,  $S$ ; mobile device, MD; cloud server, CS; moving
path;
Output The optimal task scheduling plan
1 For each  $i \in [1, k]$ 
2 {      initial the matrix of task execution  $Q_i$  and search speed  $v_i$  randomly;
3          calculate the energy consumption, the latency and the fitness value according to the matrix of task execution and the moving
path;
4 } //End For each  $i \in [1, k]$ 
5 While ( $i < \text{Maximum iterations}$ ) //Set maximum number of iterations
6 {      update the matrix of task execution according to the search speed  $v_i$ ;
7          For each  $j \in [1, k]$ 
8              { calculate the energy consumption, latency and fitness value according to the matrix of task execution  $Q_j$  and the moving
path;
9              } //End For each  $j \in [1, k]$ 
10          Select the matrix of task execution  $Q_j$  with the lowest fitness value as the optimal task scheduling plan;
11          Update the inertia weight;
12          Update the search speed  $v_j$ ;
13 } //End While
14 Update the optimal task scheduling plan;

```

ALGORITHM 2: Task Offloading and Scheduling Strategy (TOSS).

human beings, the deadline constraint is set as 0.6 s [3]. As shown in Formula (11), the fitness value can be calculated by two parts. One part is the total energy consumption when  $T_{\text{respond}} \geq T_{\text{sum}}$ , ( $f_1 = 1, f_2 = 0$ ); the other part is the total energy consumption when  $T_{\text{respond}} < T_{\text{sum}}$ , ( $f_1 = 0, f_2 = 1$ ). Accordingly, the penalty for unsatisfying constraint condition is regarded as the fitness value ( $f_1 = 0, f_2 = 1$ ). The penalty coefficient is set as 10, which is the same as the previous work [22].

**4.2.2. Algorithm Description.** TOSS algorithm mainly includes three steps: (1) first step: initialization of the task scheduling plan; (2) second step: iterative process; (3) third step: update the best task scheduling plan. The outer loop updates the task scheduling plan, the inertia weight, and the search speed. The inner loop calculates the fitness value of each task scheduling plan. Let the number of initial task scheduling plans, iterations, and tasks are  $I, k$ , and  $T$ , respectively. The computation complexity of TOSS algorithm is  $O(IkT)$ .

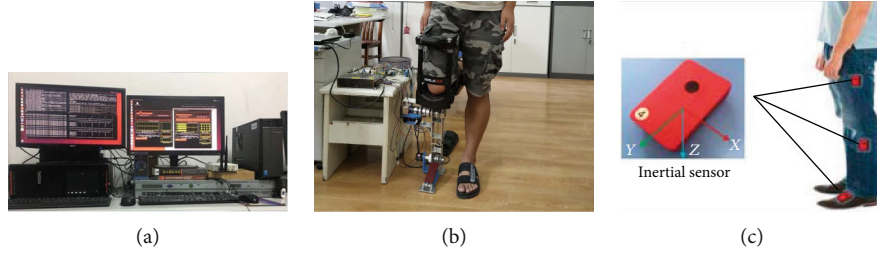


FIGURE 7: MEC environment and intelligent prosthesis used for experiments.

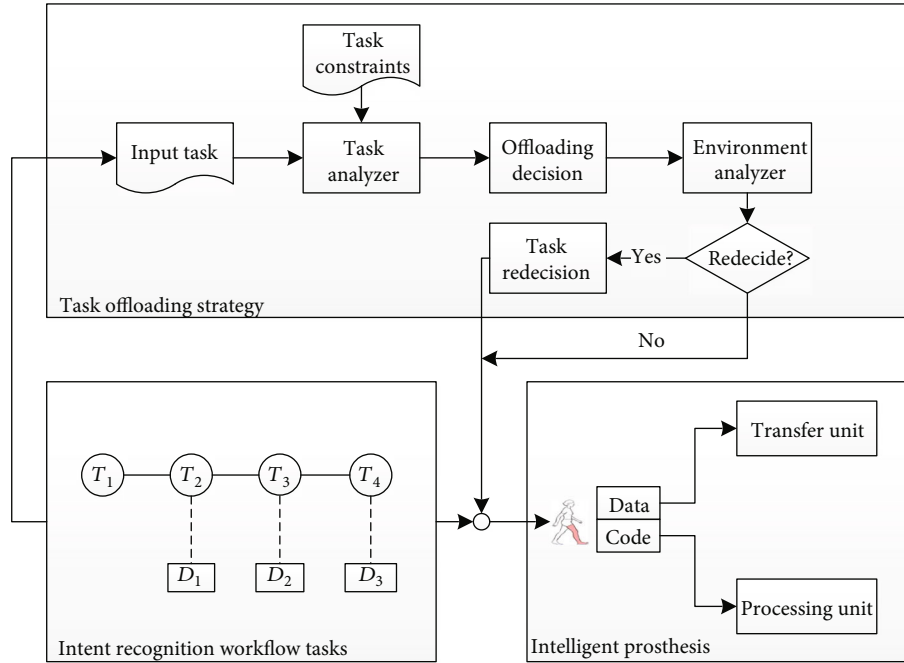


FIGURE 8: MEC-based computation management for intent recognition.

## 5. Experimental Results

In this section, we demonstrate the implementation of the proposed algorithm in MEC environment. As shown in Figure 7, the hardware environment is divided into two parts. The first part is consisted of computation resources. The cloud server is AlibabaCloud c6a (16 Core AMD EPYCTM Rome 7H12, 32 GB memory). The edge server is Dell PowerEdge XE2420 (8 Core Intel Xeon Bronze 3204, 16 GB memory), as shown in Figure 7(a). As can be seen in Figure 7(b), the second part is the intelligent prosthesis and the one we used for the experiments is iWALK 2.0 (Virtex-7 Xilinx FPGA platform). The transmitting, receiving, idle, and executing powers of the mobile processor are 0.1 W, 0.05 W, 0.02 W, and 0.5 W.

As shown in Figure 7(c), three IMUs (Noitom Perception Legacy) are placed at the thigh, shank, and ankle of the healthy leg. Two edge servers have been deployed in a 200 m  $\times$  100 m area, and the moving path passes through this area.

We implement the proposed model by real-world application in intelligent prosthesis scenario which is intent recognition application. As can be seen in Figure 8, when

the amputee is moving to the destination, the intelligent prosthesis begins to record the sensor data and send the intent recognition computation tasks to the edge servers or cloud server according to the task offloading strategy. The procedure is as follows: (1) the task offloading strategy first analyses the input data and tasks according to the characteristics and constraints of tasks. (2) The task strategy generates the offloading decision based on TOSS algorithm. (3) When the environment is changed (such as the moving path is changed), the offloading strategy decides the offloading decision again according to the environment analyser.

In the following experiments, we discuss the effect of the proposed algorithms, mainly focusing on the effect of fitness value, energy consumption, latency, and accuracy.

**5.1. Fitness Value.** In this experiment, we evaluate the performance of the proposed algorithm against CLOUD, EDGE, MOBILE, LoPRTC [23], and Edge4Sys [17] under varying number of frames. CLOUD, EDGE, and MOBILE indicate that the workflows are executed by cloud server, edge server, and mobile processor, respectively. Figure 8 shows the experiment results.

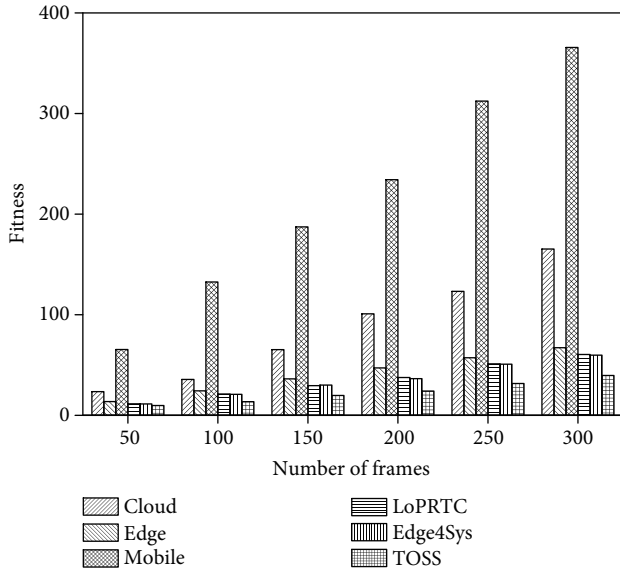


FIGURE 9: Comparison of fitness value.

In Figure 9, as the number of frames increasing, the fitness value of all strategies increases. The fitness value of TOSS is apparently lower than the other strategies. Due to the fitness value which can reflect the effectiveness of task offloading and scheduling strategy, the experiment results show that TOSS can find the offloading and scheduling decision with lower energy consumption under the deadline constraint. In the meantime, it can be seen that the fitness value of MOBILE strategy is always much higher than the other strategies. The experimental results prove that the traditional mobile device-based intent recognition algorithm leads to the growth of the energy consumption.

**5.2. Energy Consumption and Latency.** In this experiment, we compare TOSS with CLOUD, EDGE, MOBILE, LoPRTC, and Edge4Sys in energy consumption and latency. Figures 10 and 11 show the results.

As can be seen in Figure 10, the energy consumption of MOBILE strategy is much higher than the other strategies, which can prove the experiment results shown in Figure 8 in another way. In the meantime, the energy consumption of EDGE strategy is always the lowest. However, the fitness value of EDGE strategy is higher than TOSS, LoPRTC, and Edge4Sys in Figure 9. The main reason is that EDGE strategy may miss the deadline constraint; the penalty for unsatisfying constraint condition leads to the growth of the fitness value. In addition, compared with Edge4Sys, LoPRTC, MOBILE, and CLOUD, the energy consumption returned by TOSS is 9.73%, 10.01%, 71.84%, and 15.98% less than these three strategies. The experimental results show that TOSS can effectively reduce the energy consumption under the deadline constraint.

The latency of five different strategies is shown in Figure 11. It can be seen that the latency of TOSS is always the lowest. When the number of frames is less than 100, MOBILE strategy is the second-lowest. However, when the number of frames exceeds 100, Edge4Sys strategy is the

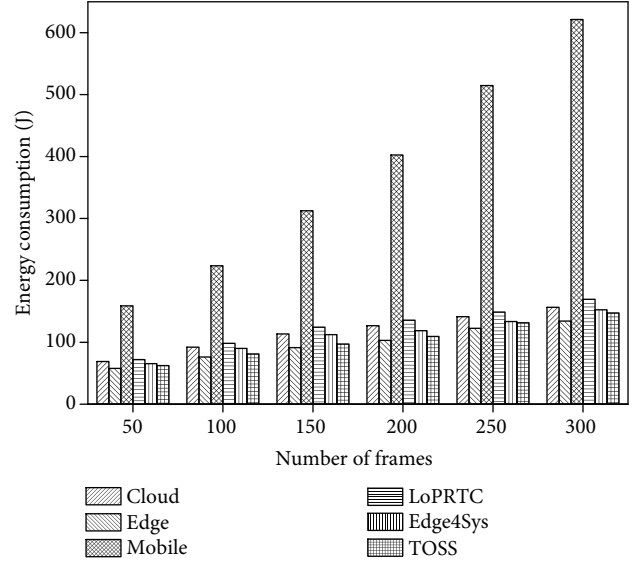


FIGURE 10: Comparison of energy consumption.

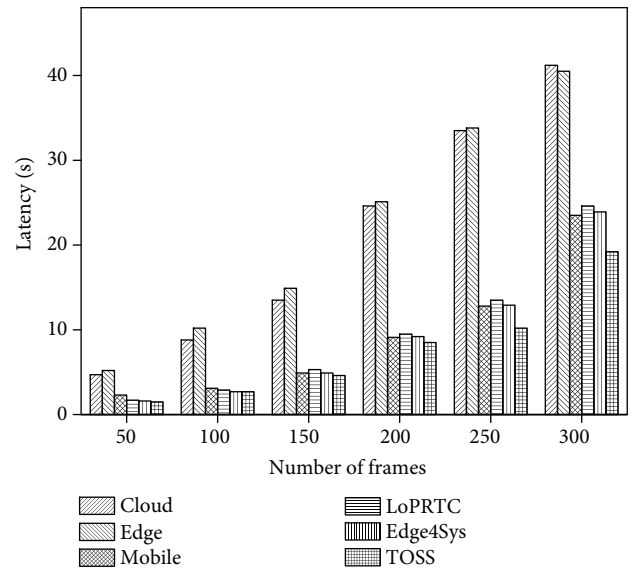


FIGURE 11: Comparison of latency.

second-lowest. It is easily to be noticed that, when the sizes of inputs and outputs are increasing, the latency generated by task transmission is less than that of computation locally.

Besides, although the latency of MOBILE strategy is less than CLOUD strategy and EDGE strategy, the energy consumption is much larger than the other strategies, as can be seen in Figure 10. The main reason is that task execution is the major energy consuming process; the size of data transmission has relatively less effect on the final results. In the meantime, the latency of CLOUD strategy and EDGE strategy is 70.44% and 77.73% higher than TOSS. It implies that data preprocessing can discard the useless information, which can save the communication time and reduce the computation load.

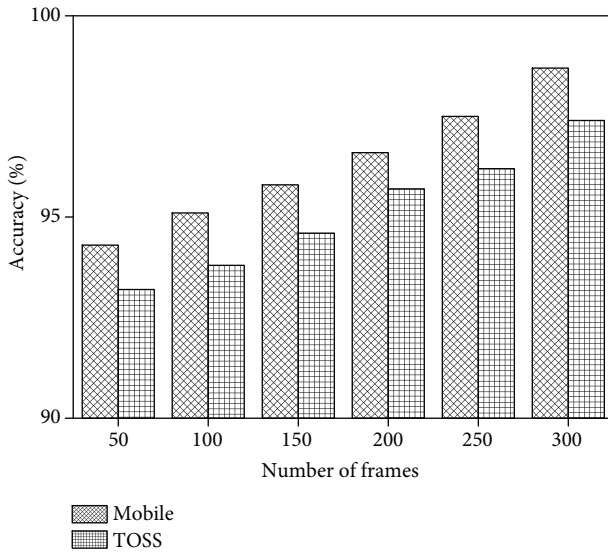


FIGURE 12: Comparison of accuracy.

In summary, the experiment results show that it is still difficult for cloud environment based intent recognition algorithm to achieve the purpose of real-time execution. The edge servers should coordinate with the other computation resources to ensure a good performance. For this reason, the proposed TOSS can always find the optimal task offloading and scheduling decision.

**5.3. Testing Accuracy.** In this experiment, we compare TOSS with MOBILE strategy in testing accuracy. Figure 12 shows the results.

In Figure 12, with the number of frames increased, the testing accuracy of the two strategies both increased as well. The accuracy of MOBILE strategy is a little higher. However, as can be seen in Figures 9 and 10, the energy consumption and latency of MOBILE strategy is much higher than TOSS. Compared with MOBILE strategy, TOSS can considerably reduce the energy consumption (71.84%) and latency (16.16%) at the expense of a relatively small decrease (1.36%) in accuracy, which is very practical in large-scale environment.

## 6. Conclusions and Future Work

In this paper, we consider the computation offloading problem of multiple heterogeneous edge servers in intelligent prosthesis scenario. The detail design of MEC-based task offloading model and mobility-aware task scheduling strategy are proposed to reduce the energy consumption and latency in a real-world MEC environment. The experimental results show that the proposed algorithms shows that the proposed algorithms can considerably reduce the energy consumption (71.84%) and latency (16.16%) at the expense of a relatively small decrease (1.36%) in accuracy.

In the future, we will develop more smart applications, such as path prediction and real-time scheduling. Furthermore, in an MEC system with multiple computation resources, reliability has a significant impact on task offload-

ing and execution, and novel algorithms are needed to offloading the tasks on the trusty resource.

## Data Availability

The experiment data supporting this experiment analysis are from previously reported studies, which have been cited, and are also included within the article.

## Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work is supported by the Key Research and Development Program of Anhui Province, under grant no. 202004a05020010; the Key Program in the Youth Elite Support Plan in Universities of Anhui Province, under grant no. gxyqZD2020043; and Natural Science Foundation of Universities of Anhui Province, under grant no. KJ2020A0694.

## References

- [1] S. U. Ben-yue, N. I. Yu, S. H. E. N. G. Min, and Z. H. A. O. Li-li, "Intent recognition of power lower-limb prosthesis based on improved convolutional neural network," *Control and Decision*, 2020.
- [2] B. H. Hu, E. J. Rouse, and L. J. Hargrove, "Using bilateral lower limb kinematic and myoelectric signals to predict locomotor activities: a pilot study," in *2017 8th International IEEE/EMBS Conference on Neural Engineering (NER)*, pp. 98–101, Shanghai, China, 2017.
- [3] K. Nazarpour, A. R. Sharafat, and S. M. Firoozabadi, "Application of higher order statistics to surface electromyogram signal classification," *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 10, pp. 1762–1769, 2007.
- [4] A. Konstantin, T. Y. Yu, R. L. Carpentier, Y. Aoustin, and D. Farina, "Simulation of motor unit action potential recordings from intramuscular multi-channel scanning electrodes," *IEEE Transactions on Biomedical Engineering*, vol. 67, no. 7, pp. 2005–2014, 2020.
- [5] He Huang, Fan Zhang, L. J. Hargrove, Zhi Dou, D. R. Rogers, and K. B. Englehart, "Continuous locomotion-mode identification for prosthetic legs based on neuromuscular-mechanical fusion," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 10, pp. 2867–2875, 2011.
- [6] M. Liu, F. Zhang, and H. Huang, "An adaptive classification strategy for reliable locomotion mode recognition," *Sensors*, vol. 17, no. 9, article 2020, 2017.
- [7] B. Y. Su, J. Wang, S. Q. Liu, M. Sheng, J. Jiang, and K. Xiang, "A CNN-based method for intent recognition using inertial measurement units and intelligent lower limb prosthesis," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 5, pp. 1032–1042, 2019.
- [8] O. P. Idowu, A. E. Ilesanmi, X. Li, O. W. Samuel, P. Fang, and G. Li, "An integrated deep learning model for motor intention recognition of multi-class EEG signals in upper limb amputees," *Computer Methods and Programs in Biomedicine*, vol. 206, no. 3, article 106121, 2021.

- [9] A. Gautam, M. Panwar, A. Wankhede, S. P. Arjunan, and D. K. Kumar, "Locomo-net: a low-complex deep learning framework for sEMG-based hand movement recognition for prosthetic control," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 8, article 2100812, 2020.
- [10] L. Lin, X. Liao, H. Jin, and P. Li, "Computation offloading toward edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1584–1607, 2019.
- [11] J. Chen and X. Ran, "Deep learning with edge computing: a review," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1655–1674, 2019.
- [12] S. K. Au, P. Dilworth, and H. Herr, "An ankle-foot emulation system for the study of human walking biomechanics," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*, pp. 2939–2945, Orlando, FL, USA, 2006.
- [13] X. Dongfang and Q. Wang, "Noninvasive human-prosthesis interfaces for locomotion intent recognition: a review," *Cyborg and Bionic Systems*, vol. 2021, article 9863761, pp. 1–14, 2021.
- [14] D. Xu and Q. Wang, "On-board training strategy for IMU-based real-time locomotion recognition of transtibial amputees with robotic prostheses," *Frontiers in Neurorobotics*, vol. 14, no. 47, pp. 1–12, 2020.
- [15] E. Li, Z. Zhou, and C. Xu, "Edge intelligence: on-demand deep learning modelco-inference with device-edge synergy," in *Proceedings of the Workshop on Mobile Edge Communications*, pp. 31–36, New York, NY, 2018.
- [16] H. Li, H. Chenghao, J. Jiang, Z. Wang, Y. Wen, and W. Zhu, "JALAD: joint accuracy and latency-aware deep structure decoupling for edge-cloud execution," in *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 671–678, Singapore, 2018.
- [17] H. Gao, X. Yi, X. Liu et al., "Edge4Sys a device-edge collaborative framework for MEC based smart systems," in *2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, Melbourne, VIC, Australia, 2020.
- [18] A. Zomaya, A. Abbas, and S. Khan, "Modeling and simulation of distributed fog environment using fog net sim++(chapter 11)," in *Fog Computing: Theory and Practice*, John Wiley & Sons, 2020.
- [19] A. Malik, T. Qayyum, and A. U. Rahman, "xFogSim: a distributed resource management framework for sustainable IoT services," *IEEE Transactions on Sustainable Computing*, vol. 23, no. 9, pp. 2005–2014, 2020.
- [20] S. S. Shah, M. Ali, A. W. Malik, M. A. Khan, and S. D. Ravana, "VFog: a vehicle-assisted computing framework for delay-sensitive applications in smart cities," *IEEE Access*, vol. 7, pp. 34900–34909, 2019.
- [21] J. Xu, X. Li, X. Liu et al., "Mobility-aware workflow offloading and scheduling strategy for mobile edge computing," in *Algorithms and Architectures for Parallel Processing. ICA3PP 2019*, S. Wen, A. Zomaya, and L. Yang, Eds., vol. 11945 of Lecture Notes in Computer Science, pp. 184–199, Springer, Cham, 2019.
- [22] J. Hu, M. Jiang, Q. Zhang, Q. Li, and J. Qin, "Joint optimization of UAV position, time slot allocation, and computation task partition in multiuser aerial mobile-edge computing systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 7231–7235, 2019.
- [23] T. Zhu, T. Shi, J. Li, Z. Cai, and X. Zhou, "Task scheduling in deadline-aware mobile edge computing systems," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4854–4866, 2019.

## Research Article

# Joint Load Balancing and Offloading Optimization in Multiple Parked Vehicle-Assisted Edge Computing

Xinyue Hu <sup>1</sup>, Xiaoke Tang <sup>2</sup>, Yantao Yu <sup>1</sup>, Sihai Qiu <sup>2</sup> and Shiyong Chen <sup>1</sup>

<sup>1</sup>School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China

<sup>2</sup>Beijing Smart-Chip Microelectronics Technology Co., Ltd., Beijing 100192, China

Correspondence should be addressed to Yantao Yu; yantaoyu@cqu.edu.cn

Received 9 August 2021; Revised 26 October 2021; Accepted 30 October 2021; Published 23 November 2021

Academic Editor: Muhammad Shiraz

Copyright © 2021 Xinyue Hu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The introduction of mobile edge computing (MEC) in vehicular network has been a promising paradigm to improve vehicular services by offloading computation-intensive tasks to the MEC server. To avoid the overload phenomenon in MEC server, the vast idle resources of parked vehicles can be utilized to effectively relieve the computational burden on the server. Furthermore, unbalanced load allocation may cause larger latency and energy consumption. To solve the problem, the reported works preferred to allocate workload between MEC server and single parked vehicle. In this paper, a multiple parked vehicle-assisted edge computing (MPVEC) paradigm is first introduced. A joint load balancing and offloading optimization problem is formulated to minimize the system cost under delay constraint. In order to accomplish the offloading tasks, a multiple offloading node selection algorithm is proposed to select several appropriate PVs to collaborate with the MEC server in computing tasks. Furthermore, a workload allocation strategy based on dynamic game is presented to optimize the system performance with jointly considering the workload balance among computing nodes. Numerical results indicate that the offloading strategy in MPVEC scheme can significantly reduce the system cost and load balancing of the system can be achieved.

## 1. Introduction

As road traffic density continues to increase and traffic data explodes, the limited computing capacity of onboard terminals cannot meet the communication and computing demand of computationally intensive onboard applications [1, 2]. The introduction of mobile edge computing (MEC) has become an effective solution to the problem of resource scarcity in vehicular networks [3, 4]. Usually, MEC can enhance network resources and enable localized data processing by deploying computation and storage resources at the edge of the network close to the users [5, 6]. Compared with remote cloud computing, it can use resource-rich servers at the roadside unit (RSU) to provide users with low-latency, high-bandwidth application services [7].

However, onboard applications such as augmented reality and autonomous driving are with higher demands on data processing and storage capabilities and still require more available resources [8, 9]. When the number of offloading tasks is large, the MEC server with limited resources will be over-

loaded and result in less efficient task execution. Moreover, deploying massive MEC servers to augment vehicular network resources would entail huge economic and time costs, which is clearly not feasible. In order to address the problem, end devices such as vehicles and gateways with limited resources can be used as infrastructures to effectively extend the computational, communication, and storage capabilities of the edge server. For example, the idea of using vehicles as communication and computing infrastructure to collaborate with other edge devices to perform tasks has been proposed in [10], which can meet the demand for considerable communication and computation capabilities. The computing power of mobile vehicles on the road has been used to assist in offloading tasks and speed up the task execution process [11–13]. As mobile vehicles are highly dynamic, it is difficult to guarantee the stability and reliability of task offloading.

Note that the PVs with vast idle resources in the roadside and parking lots can act as static network infrastructures to enhance vehicular network [14, 15]. According to a survey, about 70% of vehicles are parked for more than 20 hours



per day [16]. And most of the vehicles in the vehicular network have been equipped with sensors, wireless devices, and onboard units, which facilitates the vehicles to establish stable and reliable wireless communication and consequently form a vehicular adhoc network (VANET) [17, 18].

In the existing studies of vehicular edge networks, the data interaction between the driving vehicles and the roadside units is mainly considered. Generally, it leads to low utilization of the PV resources. Based on the above problems, a multiple parked vehicle-assisted edge computing framework is proposed in this paper, which has multiple parked vehicles to assist the edge server to perform offloading tasks. And the total system cost is minimized under the constraint of maximum allowable delay while taking load balancing and offloading optimization into account.

The main contributions of this work are summarized as follows.

- (i) The system model of multiple parked vehicle-assisted edge computing is analyzed. And the joint load balancing and offloading optimization problem is formulated to minimize the total system cost under the delay constraint. The offloading strategy is proposed to solve the optimization problem, which includes offloading node selection and workload allocation
- (ii) Considering the parked probability and resource availability of PVs, a multiple offloading nodes selection algorithm is adopted to select several candidate offloading nodes among vehicles and MEC server
- (iii) Considering the sequential nature of offloading decisions and the resource consumption during task execution, an efficient workload allocation strategy based on dynamic game is proposed to optimize system utility while considering load balancing

The rest of this paper is organized as follows. In Section II, related works about task offloading in vehicular networks are firstly introduced. The system model for multiple parked vehicle-assisted edge computing is described in detail in Section III. In Section IV, the workload allocation problem among multiple tasks and multiple computing nodes is modeled as a dynamic game process. In Section V, an efficient offloading strategy is proposed to solve the node selection and workload allocation problems. Simulation results for proposed scheme and the related analysis for different cases are provided in Section VI. Finally, the research work is concluded in Section VII.

## 2. Related Work

In the last few years, the existing work in vehicular networks is mainly utilizing MEC to provide offloading service. These research works can be divided into two main categories: one is only using MEC servers to handle task offloading requests, and the other is using remote clouds or vehicles to assist MEC servers.

*2.1. Vehicular Edge Computing.* Researches in the first category aim to solve the offloading problem in the vehicular work by only using edge servers. The MEC server has more computing power and can provide a large amount of resources for offloading services. The existing work mainly focused on improving the efficiency of task execution and avoiding server overload by optimizing resource allocation. For example, a collaborative computing offload and resource allocation optimization scheme, based on the scalable nature of tasks in driver assistance applications, was presented in [19]. In order to balance resource consumption and user experience with limited computing and spectrum resources, edge computing and social networking were combined to propose a new network system—vehicular social edge computing (VSEC) in [20]. Thus, the quality of service and quality of experience of drivers were improved by optimizing the available network resources. Moreover, a multipath transmission workload balancing optimization scheme was investigated in [21], which uses multipath transport to support communication between vehicles and edge nodes. In [22], the fiber-wireless (FiWi) technology was introduced to enhance vehicular network, and a SDN-based load-balancing task offloading scheme was also proposed to minimize the processing delay.

*2.2. Collaborate Vehicular Edge Computing.* The second category of method for handling task offloading requests is to use other infrastructure such as remote clouds, UAVs, and vehicles to collaborate with MEC servers.

*2.2.1. Remote Clouds Collaborate Vehicular Edge Computing.* The remote clouds are often introduced in edge computing to provide more offloading services. For instance, a two-tier offloading architecture for cloud-assisted MEC to improve system utility and computational latency by using collaborative computational offloading and resource allocation optimization schemes was discussed in [23]. A multi-layer data flow processing system, i.e., EdgeFlow, was presented in [24], to integrally utilize the computing capacity throughout the whole network and optimally the transmission resource allocation to minimize the system latency. Furthermore, a cloud-based tiered vehicle edge computing offloading framework that introduces nearby backup servers to make up for the lack of computing resources of MEC servers was presented in [25]. A game theoretic algorithm was used to design the optimal multilevel offloading scheme to improve the utility of vehicles and servers.

*2.2.2. Mobile Vehicles or UAVs Collaborate Vehicular Edge Computing.* Moreover, many works have proposed solutions for task offloading by using mobile vehicles or UAVs to assist MEC servers. In [26], a UAV-MEC system was investigated based on the idea of utilizing the UAV as a computing node to improve the average user latency. In [27], a cooperative UAV-enabled MEC network structure was presented to collaborate UAV offloading tasks, which the long-term utility was maximized by deep reinforcement learning-based algorithms. A distributed collaborative task offloading architecture by treating mobile vehicles as edge

computing resources was discussed to guarantee low latency and application performance in [28]. A joint energy and latency cost minimization problem was formulated while using vehicles to assist task offloading. And an ECOS scheme with three phases was proposed to effectively solve the optimal problem in [29].

**2.2.3. Parked Vehicles Collaborate Vehicular Edge Computing.** Task offloading via UAVs and mobile vehicles is highly dynamic and lead to discontinuity in communication which is highly unstable [30]. In contrast, vehicles parked on the roadside or in parking lots are relatively static and can provide a more stable and reliable task offloading service. Thus, another recent work introduces parked vehicles to extend edge computing capabilities. For instance, serving PVs as static nodes to extend vehicular network resources and the concept of parked vehicle assistance (PVA) was proposed in [31, 32]. In addition, using PVs to assist edge servers in handling offloading tasks was presented in [33], by organizing PVs into parking clusters and abstracting them as virtual edge servers. Eventually, the task offloading performance was effectively improved by a task scheduling strategy and an associated trajectory prediction model. In [34], a three-stage contract-Stackelberg offloading incentive mechanism was developed to optimize the system utility by making full use of the large amount of free resources in the parking lot. The computing resources were also classified to provide different contracts, and the problem was solved using backward induction. In [35], the system task allocation was optimized according to the collaborative vehicle edge computing (CVEC) framework by designing a contract-based incentive mechanism to schedule PVs to handle offloading tasks. And an optimal contract that maximizes subjective utility under information asymmetry was formulated to optimize user utility.

The related works discussed above in parked vehicle-assisted vehicular network rarely consider the load balance among computing nodes or just allocate the load between MEC server and single parked vehicle. Compared with them, in this paper, a MPVEC framework with multiple parked vehicles collaborating MEC server while executing offloading tasks is presented. The computing framework with distributed characteristics increases computing capacity of task offloading and provides users with more efficient and flexible offloading options. To ensure the reliable and stable task execution, a multiple offloading node selection algorithm based on the parking behavior and resource availability is proposed to select multiple appropriate PVs to accomplish the offloading tasks. Considering that the resource states of MEC servers and PVs are time-varying during task execution, an efficient workload allocation strategy is developed to optimize system performance and keep the load balancing.

### 3. System Model

**3.1. Network Entities.** In this section, the MPVEC system with network entities is mainly composed of requesting vehicles, service provider, MEC server, and several PVs, as

shown in Figure 1. More details of the function of the network entities in the system are described as follows.

**Requesting vehicle:** the requesting vehicle makes task offloading decisions based on the information provided by the service provider. Part of the task is processed by requesting vehicles locally, and the other part is uploaded to the nearby RSU through vehicle to infrastructure (V2I) communication and reasonably distributes the workloads to corresponding edge nodes.

**Service provider:** based on the computational and storage capacity of the MEC server, the service provider can collect global information, including task information as well as the computational capacity and unit energy consumption of the requesting vehicles, MEC server, and PVs. Simultaneously, according to the offloading decision, it can dispatch the MEC server and PVs to execute the corresponding workload on demand.

**MEC server:** the MEC server is richer in computing resources and can provide offloading services for requesting vehicles. The task requests are transmitted wired to the MEC server for processing via the RSU.

**Parked vehicle:** RSUs are wired to each other, and wireless connections are established between PVs and RSU via V2I communication. PVs in the parking lot can use the idle computing resources to perform offloading tasks.

**3.2. System Model.** As shown in Figure 1, it is assumed that a one-way road is within the coverage of RSU, and there are  $N$  requesting vehicles moving on the road. Each vehicle generates a computation task, which can be described as  $D_i = \{d_i, c_i, t_i^{\max}\}$  and  $i \in N = \{1, 2, \dots, N\}$ . Here,  $d_i, c_i, t_i^{\max}$  denotes the data size of task, the number of CPU cycles needed for executing task, and the maximum allowable time delay of the task, respectively.

To ensure uninterrupted communication during task execution, the task offloading process needs to be completed before the vehicle leaves the RSU coverage area. Assuming that the length of the road section covered by the RSU is  $L$ , the requesting vehicle moves on the one-way road at a constant speed of  $v$ , and its position is away from the starting position of the road section by  $l_i$ . Then, the maximum allowable time delay  $t_i^{\max}$  of the task can be represented as  $t_i^{\max} = (L - l_i)/v$ .

The task generated by the requesting vehicle can be executed locally or offloaded to edge computing nodes. The set of  $J = \{1, 2, \dots, j\}$  represents the edge computing nodes. Among them,  $j = 1$  represents the MEC server,  $j > 1$  represents PVs, and the offloading part of the task can be offloaded to multiple edge computing nodes for parallel processing. The parameter of  $x_{ij} \in \{0, 1\}$  represents the node selection variable. If the  $j$ -th edge computing node is selected to execute the  $i$ -th offloading task,  $x_{ij} = 1$  is set; otherwise,  $x_{ij} = 0$ . Let  $k_{ij}$  ( $0 \leq k_{ij} \leq 1$ ) denotes the allocation workload ratio of the  $i$ -th offloading task to the  $j$ -th edge computing node. And  $(1 - \sum_{j=1}^J x_{ij} k_{ij})$  represents the unoffloading ratio of the  $i$ -th task and should be executed locally. As the computing resource of the system is time-varying, and the resource consumption cost of each node is different, it is a key

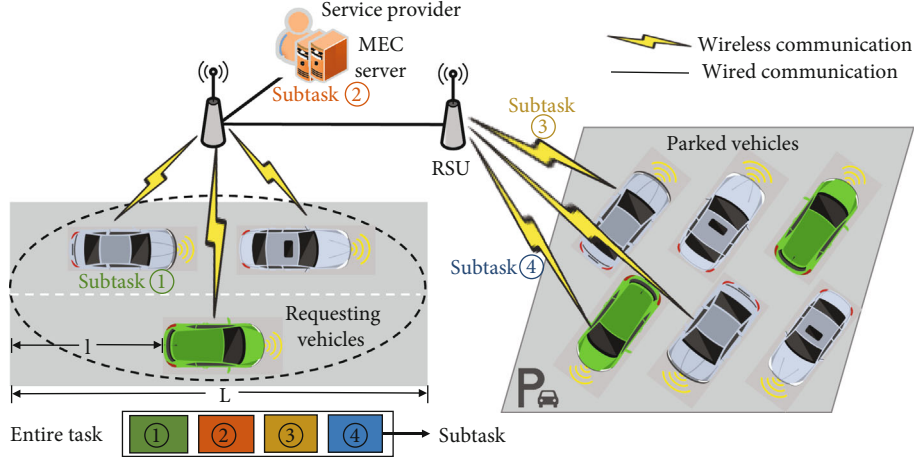


FIGURE 1: Multiple parked vehicle-assisted edge computing for task offloading.

challenge to balance the workload of each node while minimizing the whole system cost. The communication and computation model in the MPVEC framework will be described in the following sections. And the rest main parameters that will be used in this paper are listed in Table 1.

**3.3. Communication Model.** For the convenience of analysis, it is assumed that the network topology and wireless channels remain unchanged during the task execution. When the vehicle moves into the RSU coverage area, it can establish a V2I communication connection with the RSU based on IEEE 802.11p. If the task is partially or completely offloaded, the offloading part of the task is firstly transmitted to the RSU. And the MEC server, which establishes wired connection with RSU, calculates the corresponding offloading tasks. Simultaneously, the remaining offloading part is forwarded to the RSU at the parking lot, and the RSU will issue the task to the PVs for processing. Finally, the task execution result is returned. Since the size of task execution result is small, its transmission delay and energy consumption can be ignored, and only the task distribution process is considered in this paper.

- (1) Requesting vehicle to RSU: when the  $i$ -th requesting vehicle who generate a computation task  $D_i$  is occurred in the coverage of RSU, the uplink transmission rate between the  $i$ -th requesting vehicle and RSU can be expressed as

$$R_i^u = B \log_2 \left( 1 + \frac{P_i^u h_i^u}{N_0} \right), \quad (1)$$

where  $p_i^u$  is the transmission power of the  $i$ -th requesting vehicle, and  $h_i^u$  is the power gain between the  $i$ -th requesting vehicle and RSU.  $B$  and  $N_0$  are the channel bandwidth and the background noise power, respectively.

The transmission time and energy consumption of the uplink are related to the size of the task, which

can be calculated by

$$T_i^u = \sum_{j=1}^J \frac{x_{ij} k_{ij} d_i}{R_i^u}, \quad (2)$$

$$E_i^u = \rho_{\text{trans}}^u \sum_{j=1}^J x_{ij} k_{ij} d_i,$$

where  $\rho_{\text{trans}}^u$  is the uplink cost coefficient and represents the cost to calculate the unit data volume in the uplink.

- (2) RSU to MEC server ( $j=1$ ): since the connection between RSU and MEC server is in a wired manner, the transmission rate is relatively high, and the data transmission time and energy consumption are negligible
- (3) RSU to PV ( $j > 1$ ): there is a RSU near the parking lot, and the RSU is connected with the roadside RSU by wire. The transmission time and energy consumption can be neglected. The RSU will send the received offloading task requests to the PVs for processing via V2I communication, and the downlink transmission rate between the RSU and the  $j$ -th PV is

$$R_{rsu,j}^d = B \log_2 \left( 1 + \frac{P_{rsu}^d h_{rsu,j}^d}{N_0} \right), \quad (3)$$

where  $P_{rsu}^d$  is the transmission power of the RSU, and  $h_{rsu,j}^d$  is the power gain between the RSU and the  $j$ -th PV.

Furthermore, the tasks can be offloaded in parallel transmission; thus, the data transmission time of the downlink is the maximum task transmission time of each offloading part. And the transmission energy consumption is the sum of the transmission energy consumption of each offloading

TABLE 1: Main parameters.

Parameters	Description
$d_i, c_i, t_i^{\max}$	Task data size, task required computing resource, and the maximum allowable time delay of the task.
$x_{ij}, k_{ij}$	The node selection variable and the workload ratio of the $i$ -th offloading task to the $j$ -th edge computing node.
$B, N_0$	Wireless channel bandwidth and white Gaussian noise power.
$p_i^u, p_i^d$	Uplink and downlink transmission power.
$h_i^u, h_{rsu,j}^d$	The power gain between the $i$ -th requesting vehicle and RSU and the power gain between RSU and the $j$ -th PV.
$f_i^{\text{loc}}, f_{ij}^{\max}$	CPU computing power of the $i$ -th requesting vehicle and the max CPU computing power of the $j$ -th edge computing node.
$\rho_{\text{trans}}^u, \rho_{\text{trans}}^d$	Uplink and downlink cost coefficient.
$\rho_{\text{cal}}^{\text{loc}}, \rho_{\text{cal}}^{\text{off}}$	Energy consumption per CPU cycle of requesting vehicle and edge computing nodes.
$r_{\max}$	The maximum allowable computing resources occupancy rate of the edge computing nodes in a certain period of time.
$k_i^*$	The optimal strategy of the $i$ -th requesting vehicle.
$T$	Time span of the time period.
$\delta(t), \chi(t)$	Probability density function and accumulative distribution function of the PV parking durations $t$ .
$P_{ij}, aq_{ij}$	The probability value of the $j$ -th PV remaining parked at the execution time period of the $i$ -th task and accumulative parking durations of the $j$ -th PV.

part, which are defined as

$$T_i^d = \max \left\{ \frac{k_{ij}d_i}{R_{rsu,j}^d} \right\}, \quad (4)$$

$$E_i^d = \rho_{\text{trans}}^d \sum_{j=2}^J x_{ij}k_{ij}d_i.$$

We let  $\rho_{\text{trans}}^d$  as the downlink cost coefficient and represents the cost to calculate the unit data volume in the uplink.

As a result, the total transmission time and energy consumption for data transmission to the edge computing node for the offloading part of  $D_i$  are expressed, respectively, as

$$T_i^{\text{trans}} = T_i^u + T_i^d = \sum_{j=1}^J \frac{x_{ij}k_{ij}d_i}{R_i^u} + \max \left\{ \frac{k_{ij}d_i}{R_{rsu,j}^d} \right\}, \quad (5)$$

$$E_i^{\text{trans}} = E_i^u + E_i^d = \rho_{\text{trans}}^u \sum_{j=1}^J x_{ij}k_{ij}d_i + \rho_{\text{trans}}^d \sum_{j=2}^J x_{ij}k_{ij}d_i.$$

### 3.4. Computation Model

- (1) Compute task locally: the unoffloading part of the task is calculated by the requesting vehicles locally. And the delay and energy consumption are related to the number of CPU cycles required by the task  $D_i$ , which can be calculated by

$$T_{ij} = \frac{k_{ij}c_i}{f_{ij}} + T_{ij}^{\text{wait}}, \quad (7)$$

$$E_{ij} = \rho_{\text{cal}}^{\text{off}} k_{ij}c_{ij}.$$

$$T_i^{\text{loc}} = \frac{(1 - \sum_{j=1}^J x_{ij}k_{ij})c_i}{f_i^{\text{loc}}}, \quad (6)$$

$$E_i^{\text{loc}} = \rho_{\text{cal}}^{\text{loc}} \left( 1 - \sum_{j=1}^J x_{ij}k_{ij} \right) c_i,$$

where  $f_i^{\text{loc}}$  is the computing capability of the  $i$ -th requesting vehicle, and  $\rho_{\text{cal}}^{\text{loc}}$  is the energy consumption required to calculate the unit CPU cycle.

- (2) Compute task by edge computing nodes: when the task  $D_i$  is offloaded partially to the  $j$ -th ( $j \in J$ ) edge computing node, the task processing delay is related to the computing capability of the edge computing node

The computing resources of the  $j$ -th edge node are limited and changed with time during the task  $D_i$  execution, which can be described as  $f_{ij} \in [0, f_{ij}^{\max}]$ , and  $f_{ij}^{\max}$  is the max computing power of the  $j$ -th edge node.

When the computing resources occupancy rate of the edge nodes in a certain period of time is greater than its own threshold of  $r_{\max}$ , the tasks in the node will not be processed in parallel and need to be stored in the waiting queue and executed in sequence according to the delay constraint. Therefore, the task processing delay at the edge node mainly includes two parts: task calculation time and task waiting time, which can be written as

Let  $\rho_{\text{cal}}^{\text{off}}$  represents the energy consumption required to calculate the unit CPU cycle of the edge node, where the  $\rho_{\text{cal}}^{\text{off}}$  of the PV is smaller than that of the MEC server.

The offloading part of the task request  $D_i$  generated by the  $i$ -th requesting vehicle can be processed in parallel by multiple edge computing nodes. Therefore, the task offloading delay is mainly composed of the task transmission delay and the task processing delay. And the largest processing latency among edge computing nodes is used as the task offloading latency

$$T_i^{\text{off}} = T_i^{\text{trans}} + \max \{T_{ij}\}. \quad (8)$$

Task offloading energy consumption is the sum of the transmission energy consumption and processing energy consumption of each offloading part

$$E_i^{\text{off}} = E_i^{\text{trans}} + \sum_{j=1}^J E_{ij} \quad (9)$$

**3.5. Problem Formulation.** For the task  $D_i$  generated by the  $i$ -th requesting vehicle, there is a delay and energy consumption during processing, which mainly contain two aspects: the local processing part and the offloading processing part. Due to the parallel processing of tasks, the total task processing latency is the maximum latency for local processing and task offloading processing, which can be expressed as

$$T_i = \max \{T_i^{\text{loc}}, T_i^{\text{off}}\}. \quad (10)$$

The total energy consumption of the task processing can be written as

$$E_i = E_i^{\text{loc}} + E_i^{\text{off}}. \quad (11)$$

A cost function for task  $D_i$  is defined as the combination of executing time and energy consumption.

$$U_i = \alpha T_i + \beta E_i, \quad (12)$$

where  $\alpha + \beta = 1$ . The goal in this paper is to minimize the whole cost of all distributed tasks with the latency constraint, while joint consider load balancing and offloading decision. The optimizing problem for all tasks can be formulated as

$$\min_{\{x_{ij}, k_{ij}\}} U = \min_{\{x_{ij}, k_{ij}\}} \sum_{i=1}^N U_i, \quad (13)$$

$$\text{s.t. } 0 \leq k_{ij} \leq 1, \forall i \in N, j \in J, \quad (14)$$

$$T_i \leq t_i^{\text{max}}, \forall i \in N, \quad (15)$$

$$x_{ij} \in \{0, 1\}, \forall i \in N, j \in J, \quad (16)$$

$$0 \leq \sum_{j=1}^J x_{ij} k_{ij} \leq 1, \forall i \in N, j \in J, \quad (17)$$

$$\alpha + \beta = 1, \alpha \geq 0, \beta \geq 0. \quad (18)$$

In the above optimization model, constraint (14) denotes the workload ratio  $k_{ij}$  is a continuous variable, which cannot exceed to 1. And each requesting vehicle can offload its task to multiple computing nodes according to the workload ratio of  $k_{ij}$ . Constraint (15) ensures that the task execution delay cannot exceed the maximum allowable task delay  $t_i^{\text{max}}$ . Constraint (16) indicates the offloading node selection variable, and if the  $i$ -th requesting vehicle selects the  $j$ -th edge computing node to offload part of the task, then  $x_{ij} = 1$ ; otherwise,  $x_{ij} = 0$ . Constraint (17) presents the total offloading task of the  $i$ -th requesting vehicle cannot exceed to 1 and makes the problem a mixed integer nonlinear optimization problem. In constraint (18),  $\alpha$  and  $\beta$  are the weights of time delay and energy consumption in the total cost, respectively, which can be dynamically adjusted according to the task type to meet the computing requirements of different tasks.

## 4. Multitask Multinode Dynamic Game

In this section, the workload allocation for multiple tasks in multiple computing nodes is modeled as a dynamic game process. Considering that offloading decisions are sequential, the requesting vehicle who makes the former decision will have an impact on the requesting vehicle who makes the subsequent decision. Thus, to ensure the sequential rationality of the game process, each requesting vehicle is required to make the optimal decision, so that the overall strategy of the system is optimal.

The service provider in the proposed framework can provide requesting vehicles with global information (including the available computing capability of edge nodes and task queuing sequence). To optimize the total system cost of task execution, sequential decisions for different requesting vehicles are made to offload part or all tasks to edge computing nodes. At the same time, both sides of the game complete distributed autonomous decision making in the game process, which can obtain the optimal utility and effectively relieve the computational pressure of the MEC server.

The dynamic game process with multiple tasks and multiple computing nodes can be defined by  $G(N, K, U)$ , while the three elements of the game can be described as

- (1)  $N = \{1, 2, \dots, i, \dots, n\}$  represents the requesting vehicle players in the game that generates the tasks and makes task offloading decisions
- (2)  $K_n = \{k_1, k_2, \dots, k_i, \dots, k_n\}$  means the task offloading decision of the requesting vehicle players, where  $k_i = \{k_{i1}, k_{i2}, \dots, k_{ij}\}$ . And  $k_{ij}$  represents the workload proportion of task  $D_i$  performed by  $j$ -th the edge node
- (3) The cost function  $U_i$  represents the cost required for requesting vehicle players to perform tasks, including task execution time and energy consumption

Therefore, the offloading decision for requesting vehicle players other than the  $i$ -th requesting vehicle player can be described as  $k_{-i} = \{k_1, \dots, k_{i-1}, k_{i+1}, \dots, k_n\}$ , and the  $i$ -th player needs to choose a strategy to minimize the task execution time and energy consumption, which can be expressed as

$$\min_{k_i \in (0,1)} U_i(k_i, k_{-i}), \forall i \in N. \quad (19)$$

Next, the existence of the Nash equilibrium point in the dynamic game process is discussed.

*Definition 1.* There exists a strategy set  $K_n^* = \{k_1^*, k_2^*, \dots, k_n^*, \dots, k_n^*\}$  in the dynamic game  $G(N, K, U)$  and if

$$U_i(k_i^*, k_{-i}^*) \leq U_i(k_i, k_{-i}), \forall k_i \in K_n, \quad (20)$$

then strategy set  $K_n^*$  is the Nash equilibrium of game  $G$ . At the Nash equilibrium point, it is impossible for any player to change the strategy to obtain greater utility; that is, each requesting vehicle has made the optimal offloading decision to minimize the task execution cost.

Meanwhile, for the  $i$ -th requesting vehicle, in order to minimize its own cost, the optimal strategy  $k_i^*$  needs to be obtained by solving the following problem.

$$\mu(k_i) = \arg \min_{\{k_i\}} U_i = \alpha T_i + \beta E_i. \quad (21)$$

The optimal strategy of  $k_i^*$  can be obtained by solving the following formula:

$$\frac{\partial^2 U_i(k_i)}{\partial^2 k_i} = 0. \quad (22)$$

It can be easily concluded that the formula for solving the optimal strategy is convex, and there is an optimal solution. Hence, there is an optimal strategy in the dynamic game process between multiple tasks and multiple computing nodes.

## 5. Efficient Workload Allocation Strategy

In this section, an efficient offloading strategy is proposed to minimize the total cost of task execution, which joint consider load balancing and offloading optimization. In the task scheduling problem formulated in this paper, the value of the offloading selection variable is 0 or 1, while the task offloading ratio can be any value between 0 and 1. Therefore, the optimization problem is a mixed integer nonlinear optimization problem. We divide it into two subproblems to solve, namely, offloading node selection and workload allocation.

*5.1. Offloading Node Selection.* Different from the reported works that mainly use single computing node to collaborate MEC server in task processing, in this paper, the task  $D_i$  generated by the  $i$ -th requesting vehicle is considered to decom-

pose into multiple subtasks, and then it is offloaded to multiple edge computing nodes for joint execution. And a multiple offloading nodes selection algorithm is designed to select several appropriate computing nodes for parallel processing tasks, which is described in Algorithm 1.

It is assumed that the maximum computing power of the  $i$ -th requesting vehicle and the  $j$ -th edge computing node are  $f_{ij}^{\max}$  and  $f_i^{\text{loc}}$ , respectively. The maximum computing power of each node is fixed, but the computing resources of each node change dynamically during task execution process. Part of the computing resources are occupied during task execution, and released after the task execution is completed. When the  $i$ -th requesting vehicle selects offloading nodes, the MEC server must be used as one of the offloading nodes to prevent overloading at the PVs, considering that it can provide strong computing power. As a result, the appropriate offloading nodes are mainly selected among the PVs within the coverage area of RSU.

The appropriate offloading nodes in the PVs are selected to minimize the system cost by evaluating the execution cost of subtasks. As shown in Algorithm 1, the computation task is first divided into several subtasks with equal size, and the workload ratio of the single offloading task  $\omega$  is set to 0.1. Then, the local and each edge node processing cost increment required for this subtask  $\Delta u_i^{\text{loc}}$  and  $\Delta u_{ij}$  is calculated and compared according to equation (12). The additional waiting time  $T_{ij}^{\text{wait}}$  caused by resource consumption is also considered. The PV that satisfies the parking probability constraint  $P_{ij} \geq P_{th}$  will be selected while its cost increment is lower than the local ( $\Delta u_{ij} < \Delta u_{loc}$ ). Simultaneously, if the  $j$ -th edge node is selected to executing part of the task  $D_i, x_j$  is set to 1, and the workload ratio  $k_{ij}$  and computing power  $f_{ij}$  will be updated during task execution.

During the task scheduling process, the  $i$ -th requesting vehicle evaluates the current resource availability status of the  $j$ -th PV, which can be described by the probability value  $P_{ij}$  of the  $j$ -th PV remaining parked state at the execution time period of the  $i$ -th task [16]. And the  $P_{ij}$  can be calculated by

$$P_{ij} = \int_{aq_{ij}+T}^{t_{\max}} \frac{\delta(t)}{1 - \chi(aq_{ij})} dt = \frac{1 - \chi(aq_{ij} + T)}{1 - \chi(aq_{ij})}, \quad (23)$$

where  $t \in [0, t_{\max}]$  indicates the parking durations,  $\delta(t)$  denotes the probability density function of the  $j$ -th PV parking duration, and  $\chi(t)$  is the cumulative distribution function of  $\delta(t)$ .  $T$  is time span of the time period. The  $q_{ij}$  denotes the time interval detecting the parking behavior of the  $j$ -th PV, and the parameter of  $a$  is constant. The accumulative parking durations until now is recorded as  $aq_{ij}$ . Thus, the probability that the PV will continue to stay parked for at least  $T$  time slots can be predicted.

When the PV stays for a specified period of time with a higher probability, it can provide more stable and reliable resources for task execution. If the task is assigned to the

**Input:** Task  $D_i = \{d_i, c_i, t_i^{\max}\}$ ; the offloading task workload  $\omega$ ; the computing power  $f_i^{lo}, f_{ij}$ ; the parking probability  $P_{ij}$ .  
**Output:** The node selection variable  $x_{ij}$  ( $j=1$  denotes MEC server,  $j>1$  denotes PV).

- 1: **Initialization:**  $\Delta u = 0, \Delta t = 0, \Delta e = 0$  and  $x_{i1} = 1$ .
- 2: **calculate** the execution time  $T_i^{loc}$  and  $T_{ij}$ , the energy consumption  $E_i^{loc}$  and  $E_{ij}$ .
- 3: **set**  $\Delta u_i^{loc} = T_i^{loc} + E_i^{loc}$
- 4: **if**  $k_{ij} \geq r_{\max} f_{ij}^{\max}$  **then**
- 5:   **Calculate**  $T_{ij}^{wait}$
- 6:   **set**  $\Delta t_{ij} = T_{ij} + T_{ij}^{wait}$
- 7: **else**
- 8:   **set**  $\Delta t_{ij} = T_{ij}$
- 9: **end if**
- 10: **set**  $\Delta e_{ij} = E_{ij}$
- 11: **then calculate**  $\Delta u_{ij} = \Delta t_{ij} + \Delta e_{ij}$
- 12: **if**  $P_{ij} \geq P_{th}$  and  $\Delta u_{ij} < \Delta u_{i,loc}$  **then**
- 13:   **set**  $x_{ij} = 1$
- 14:   **update**  $k_{ij} = k_{ij} + \omega, f_{ij} = f_{ij} - \omega$
- 15: **else**
- 16:   **set**  $x_{ij} = 0$
- 17: **end if**
- 18: **return**  $x_{ij}$

ALGORITHM 1: Multiple offloading node selection algorithm.

PV, the extra task retransmission overhead caused by the departure of the PV can be effectively avoided. Thus, the probability that the PV keeps parked state during the task execution period is used as an important indicator to measure the availability of PV resources.

In Algorithm 1, the PVs satisfying  $P_{ij} \geq P_{th}$  can be hopefully selected as offloading nodes to execute the workload, where  $P_{th}$  is a predefined threshold value in the PV selection process. Furthermore, when selecting a suitable PV as an offloading node, it is necessary to consider the computing power of the corresponding PV itself and the energy consumption per unit. The PVs that satisfy both the parking probability constraint of  $P_{ij} \geq P_{th}$ , and the less task executing cost of  $\Delta u_{ij} < \Delta u_{i,loc}$  will be selected as the candidate offloading nodes. According to Algorithm 1, stable and reliable offloading nodes can be obtained to meet the task workload allocation requirements.

**5.2. Workload Allocation.** The workload allocation between multiple tasks and multiple nodes is modeled as a dynamic game process. Since the requesting vehicles can obtain global information according to the service provider, the game process can be regarded as a complete information game. The requesting vehicles need to make sequential decisions based on the priorities defined by task delay constraints. Considering the resource consumption in the system, a workload allocation algorithm based on dynamic game is proposed, and the backward induction is used to assist requesting vehicles in formulating their strategies. When all requesting vehicles make the optimal decision, the Nash Equilibrium is reached and the game ends. The specific steps are described in Algorithm 2 as follows.

Assuming that the task set generated by the requesting vehicles has been prioritized according to the delay

constraint, that can be given as  $t_1^{\max} < t_2^{\max} < \dots < t_i^{\max}, i \in N$ . According to the proposed algorithm, the requesting vehicles allocate the workload among the local and the selected offloading nodes and makes offloading decisions in turn.

Among them, the  $(i+1)$ -th requesting vehicle who makes the later decision develops an offloading strategy  $k_{i+1}$  based on the former decision  $k_i$  of the  $i$ -th requesting vehicle, and then it feeds the developed strategy  $k_{i+1}$  to the  $i$ -th requesting vehicle. If the former  $i$ -th requesting vehicle has a lower-cost strategy  $k_i$  in this case, the strategy  $k_i$  is updated and the later strategy  $k_{i+1}$  changes accordingly. Iteration keeps until the strategies and costs are both no longer changed, and then the optimal solution  $k_i^*$  and  $k_{i+1}^*$  are obtained. This step is repeated until all requesting vehicles obtain the optimal strategy  $K_n^* = \{k_1^*, k_2^*, \dots, k_i^*, \dots, k_n^*\}$ , which results in the minimum total system cost and joint consider the load balancing.

## 6. Numerical Results

In this section, the performance of the proposed MPVEC scheme through numerical research is evaluated. By formulating an efficient offloading strategy, the requesting vehicles allocate the task requests to the local and multiple edge computing nodes for joint execution.

**6.1. Parameter Setting.** We consider a unidirectional road with a section of length  $L = 600$  m in the coverage of RSU, and the RSU is equipped with a MEC server which can provide offloading services. On the road section, there are [10-50] requesting vehicles driving at a constant speed of  $v = 40$  km/h, and each vehicle generates a delay-sensitive task request. And there are [5-15] vehicles parking in the parking lot nearby, which can provide idle resources. The data size of

**Input:** The task  $D_i = \{d_i, c_i, t_i^{\max}\}$ ,  $i \in N$ ; the node selection variable  $x_{ij}$ ; the offloading task workload  $\omega$ .

**Output:** The final workload strategy  $K_n^* = \{k_1^*, k_2^*, \dots, k_i^*, \dots, k_n^*\}$ ,  $k_i^* = \{k_{i1}, k_{i2}, \dots, k_{ij}\}$ ,  $k_{ij} \in [0, 1]$ , let  $j = 0$  denotes local,  $j > 0$  denotes edge node.

**Initialization:**  $k_{ij} = 0$ ,  $u_i = 0$ .

**Step 1.** Workload allocation between nodes.

- 1: **for** each task  $D_i$
- 2:   **while**  $\omega_i < c_i$  **do**
- 3:     **calculate** the incremental cost of processing unit-sized tasks locally  $\Delta u_{i0} = T_i^{loc} + E_i^{loc}$
- 4:     **if**  $x_{ij} \neq 0$  **then**
- 5:       **calculate**  $\Delta u_{ij} = T_{ij} + T_{ij}^{wait} + E_{ij}$
- 6:     **end if**
- 7:     **compare**  $\Delta u_{i0}, \Delta u_{i1}, \dots, \Delta u_{ij}$
- 8:     **allocate** subtask to node  $j$  with the smallest cost increment **then**  $k_{ij} = k_{ij} + \omega$ .  $u_i = u_i + \Delta u_{ij}$
- 9:     **set**  $\omega_i = \omega_i + \omega$
- 10:    **end**
- 11: **set**  $k_i = \{k_{i0}, k_{i1}, k_{i2}, \dots, k_{ij}\}$

**Step 2.** Iterative to NE based on backward induction.

- 12: **let**  $i = i + 1$  and repeat step 1
- 13: **calculate**  $k_{i+1} = \{k_{i+1,0}, k_{i+1,1}, k_{i+1,2}, \dots, k_{i+1,j}\}$ ,  $u_{i+1}$
- 14: **send**  $k_{i+1}$  to user  $i$  and repeat step 1
- 15: **calculate**  $k_i, u_i$
- 16: **if**  $u_i < u_{i+1}$  **then**
- 17:    **update**  $k_i = k_i'$
- 18:    **send**  $k_i$  to user  $i + 1$  and repeat step 1
- 19:     $n = n + 1$
- 20:    **calculate**  $k_{i+1}, u_{i+1}$
- 21:    **repeat** step 2 until  $u_{i+1}^{(n)} = u_{i+1}^{(n-1)}$  and  $u_i^{(n)} = u_i^{(n-1)}$
- 22:    **set**  $k_i^* = k_i^{(n)}$  and  $k_{i+1}^* = k_{i+1}^{(n)}$
- 23:    **end if**
- 24: **return**  $k_i^*, k_{i+1}^*, u_i, u_{i+1}$

ALGORITHM 2: Workload allocation algorithm based on dynamic game.

the task is  $d_i = [100, 1000]$  KB, the number of CPU cycles required for computing  $c_i$  is  $[500, 1500]$  megacycles, and the maximum allowable task delay  $t_i^{\max} = (L - l_i)/v$  is related to the location of the moving vehicle. We set the location  $l_i = [0, 300]$  m, where the requesting vehicle is located to ensure that the vehicle can complete the task offloading process before leaving the RSU coverage area.

In addition, the probability density function of parking durations of these PVs  $\delta(t)$  is formulated by [16], and the parking probability  $P_{ij}$  can be calculated according to equation (23). In the simulation, the requesting vehicles prefer to choose the PV as the candidate offloading node if its parking probability is larger than 0.85. More simulation parameters are shown in Table 2.

**6.2. Performance Comparison.** In this section, the proposed offloading strategy in MPVEC is simulated, and the effectiveness and feasibility of the proposed scheme is evaluated under the same or different parameters and compared with the following schemes.

- (1) All task requests generated by the requesting vehicles are computed locally (local computing, LC)

- (2) There is only one MEC server in the system to provide offloading services, and there is no PV to assist in the computation. The workload is allocated between local and MEC server (no parked vehicles, NP)
- (3) The system has one MEC server and multiple PVs to provide offloading services, but tasks can only be offloaded to single node for processing (single node computing, SNC)

In Figure 2, the system cost of different offloading scenarios with the same parameters is compared. We set the number of tasks generated by the requesting vehicle is  $N = 10$ , the data size of the requesting task is equal to 500 KB, and the number of required CPU cycles is equal to 1000 megacycles. And there is one MEC server and 10 PVs in the scenario to provide offloading services. As is shown in Figure 2, the requesting vehicle generates the largest system cost when choosing LC scheme, due to that the requesting vehicle itself has weak computing power and generates large computing latency. Compared with the LC scheme, the tasks can be offloaded to the MEC server and PVs, which have much larger resources than the requesting vehicles. Hence, the system cost of the other three schemes cut down as a result.



TABLE 2: Simulation parameters.

Parameters	Description	Value
$d_i$	Task data size (KB)	[100, 1000]
$c_i$	Task required computing resource (megacycles)	[500, 1500]
$B$	Wireless channel bandwidth (MHz)	10
$p_i^u$	Uplink transmission power (W)	1
$p_i^d$	Downlink transmission power (W)	5
$h_i^u, h_{rsu,j}^d$	Power gains	1
$N_0$	White Gaussian noise power (dBm)	-100
$f_i^{loc}$	Max computing power of requesting vehicle (GHz)	[0.5, 1]
$f_{ij}^{max}$	Max computing power of edge node (GHz)	8, [2, 3]
$\rho_{trans}^u$	Uplink cost coefficient (J/KB)	$1 \times 10^{-4}$
$\rho_{trans}^d$	Downlink cost coefficient (J/KB)	$1 \times 10^{-4}$
$\rho_{cal}^{loc}$	Energy consumption per CPU cycle of requesting vehicle (J/megacycles)	$1.2 \times 10^{-3}$
$\rho_{cal}^{off}$	Energy consumption per CPU cycle of edge node (J/megacycles)	$2 \times 10^{-3}$ [1, 2] $\times 10^{-3}$
$\alpha, \beta$	The weights of time delay and energy consumption in the total cost	0.5, 0.5
$P_{th}$	Predefined threshold value.	0.85

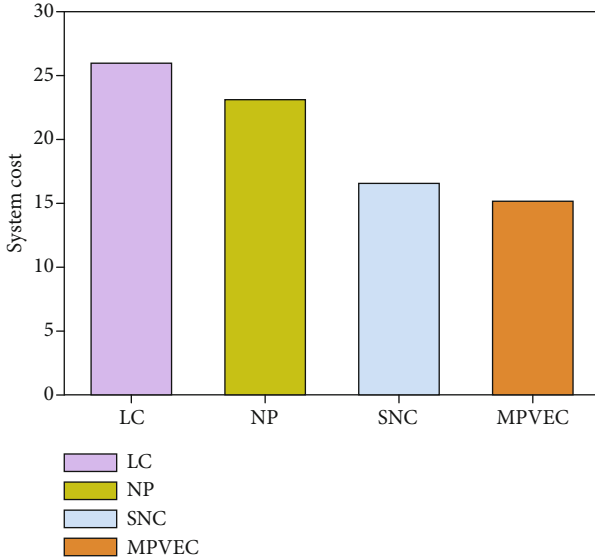


FIGURE 2: System cost under different scenarios.

Simultaneously, the MPVEEC scheme shows the excellent performance under the same parameters, which can significantly reduce the system cost. As no PVs can provide offloading services in NP, by comparing NP with MPVEEC, it clearly proves that the introduction of PVs is beneficial to the system performance. Moreover, compared with the SNC scheme, the MPVEEC decomposes the tasks to multiple nodes for joint execution, the multinode distributed processing can provide more node selectivity for users, and the system cost is efficiently reduced. Numerical results show that the system cost of the proposed MPVEEC is 41.5%, 34.6%, and 7.7% lower than of LC, NP, and SNC, respectively.

In Figure 3, the impact of the different number of tasks generated by the requesting vehicle on the system cost is illustrated. With the increasing number of tasks generated by the requesting vehicles, the system cost of all schemes presents an upward trend. Due to the limited resources of the computing nodes, the number of tasks in the waiting queue increases after the resource consumption reaches the threshold value of the edge nodes themselves. It will generate additional execution time costs and lead to the increasing system cost. In addition, it can be concluded from Figure 3 that the MPVEEC shows better performance than other schemes under the same conditions. It is because that PVs can provide more computing resources with lower cost for offloading tasks execution. And the offloading strategy in MPVEEC can effectively optimize the task execution efficiency by reasonably allocating load among computing nodes. Results indicate that the system cost of LC, NP, and SNC is 17.1%, 5.1%, and 2.6%, respectively higher than MPVEEC when the number of tasks reached to 50.

In Figure 4, the impact of the number of CPU cycles required for the task of requesting vehicle generation on the system cost is illustrated. With the increase in number of CPU cycles, the computational latency and energy consumption of each node increases accordingly. Hence, the system cost has maintained an upward trend of all schemes. Moreover, it can be visualized from Figure 4 that the optimization performance of MPVEEC scheme is more obvious than other schemes. When the task requires a larger amount of computing power, the task computing requirements can still be met at a lower cost in MPVEEC. The numerical results indicate that the system cost increase with increasing CPU cycles is 60.1%, 32.6%, and 6.1% higher for LC, NP, and SNC than MPVEEC, respectively. As a result, it can be

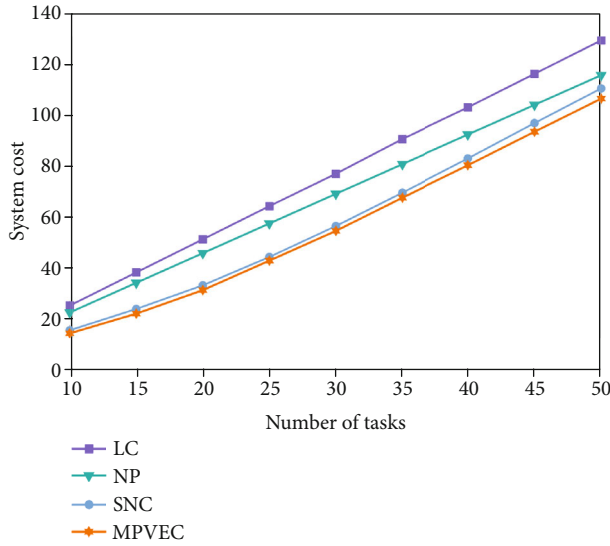


FIGURE 3: System cost under the change of task numbers.

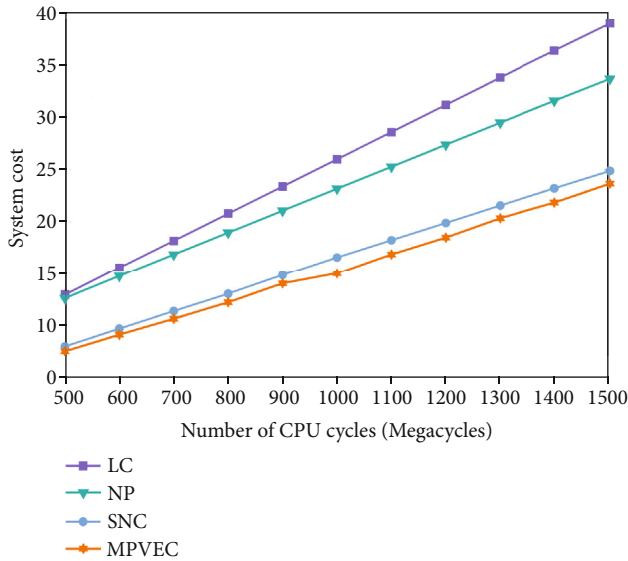


FIGURE 4: System cost under the change of CPU cycle numbers.

effectively proved that utilizing the large amount of idle resources of multiple PVs and allocating workload reasonably can greatly expand the edge computing capacity while providing offloading services to users at low cost.

In Figure 5, the impact of the different number of PVs on the system cost and the workload borne by the MEC server is illustrated. It can be seen the workload ratio of the MEC server and system cost decrease significantly at the beginning increase of the PV numbers. It is because that more available resources are provided for the system to accomplish offloading tasks, and the workload ratio of MEC server is reduced. It proves that the MPVEEC scheme is feasible to use PVs to assist edge computing and the computational pressure on the server is considerably relieved. And the reasons for the decrease of system cost can be explained through two aspects: on the one hand, the PVs can execute

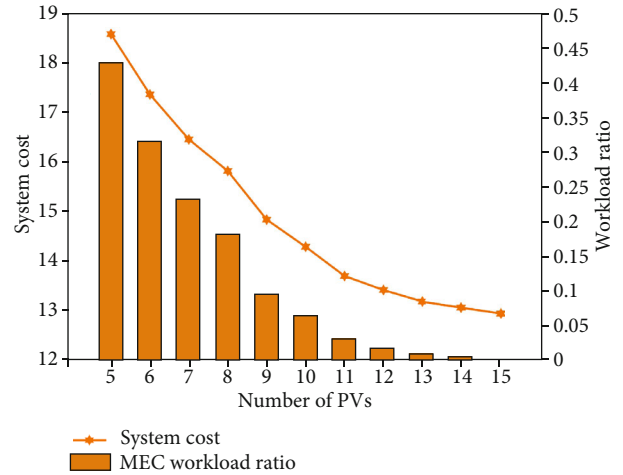


FIGURE 5: System cost and MEC workload ratio under the change of PV numbers.

part of the offloading tasks at a lower energy consumption per unit than the MEC server, and the system computing energy consumption is reduced as the number of the PVs becomes larger. On the other hand, as the number of PVs assisted in offloading increased, the tasks can be offloaded to more computing nodes for parallel processing, and the tasks waiting time in MEC server and the tasks computing time can be both reduced, which results in the reduction of system cost. Therefore, the multiple PV-assisted MEC can improve the system performance, and the workload of MEC server can be effectively relieved.

In Figure 6, the impact of different numbers of PVs on the load balancing of the system is illustrated. We set the number of PVs in the parking lot to 5 and 10, respectively. Comparing the load ratio of the selected computing nodes, it can be clearly seen that when the number of PVs is 5, except for individual nodes taking more workload, the workload ratio of the remaining computing nodes has a small difference, and load balancing of some nodes (except node 1) can be achieved. And when the number of PVs increases to 10, all the selected computing nodes can achieve load balancing. It can be easily concluded that through the proposed efficient offloading strategy in MPVEEC, the workload of each node is reasonably distributed, which can effectively reduce the overload phenomenon and realize the system load balancing.

**6.3. Complexity Analysis.** The computational complexity of the proposed offloading strategy in MPVEEC is  $O(NM)$ , where  $N$  and  $M$  are the number of requesting vehicles and the total number of computing nodes (include requesting vehicle itself and MEC server and PVs), respectively. In [28], the collaborative task offloading strategy based on a computation task and resource sharing mechanism between vehicles and edge infrastructures was reported. Its computational complexity is  $O(NM)$ , where  $N$  and  $M$  are the total number of edge infrastructures and the number of vehicles in the task offloading subcloudlet, respectively. A cloud-based mobile edge computing (MEC) offloading framework

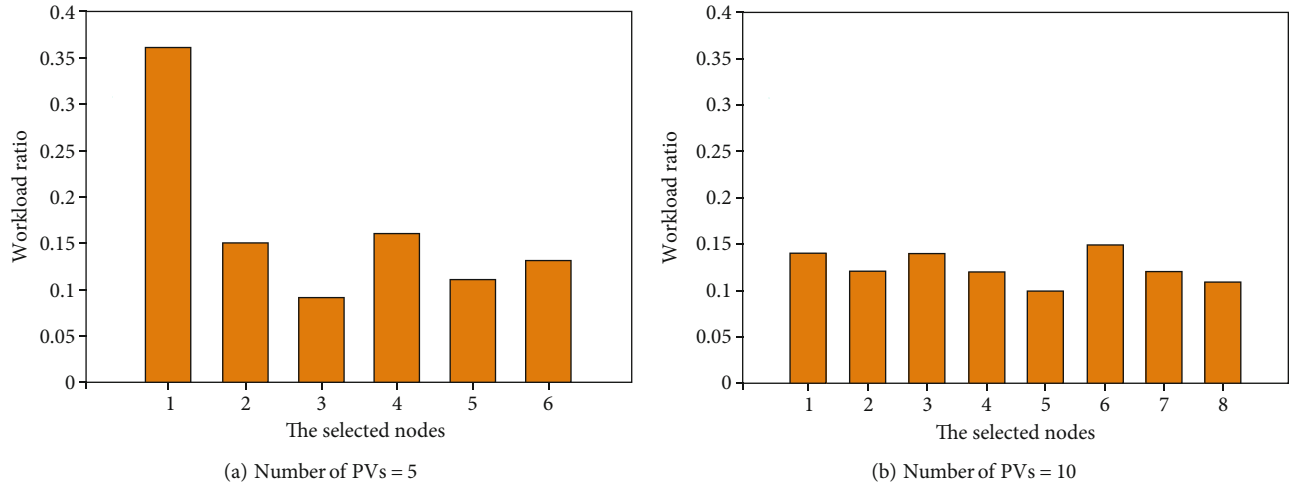


FIGURE 6: Workload ratio of selected nodes with different number of PVs.

in vehicular networks was proposed, and an efficient computation offloading strategy based on contract theoretic approach was introduced to maximize the benefit of the MEC service provider and improve the utilities of the vehicles in [36]. Its computational complexity was given as  $O(NM)$ , where  $N$  is the number of computation tasks types, and  $M$  is the number of MEC servers. The computation complexity of the proposed offloading strategy is similar to that of the algorithms mentioned above.

## 7. Conclusion

In this paper, an offloading strategy in MPVEC is investigated to optimize the system performance with jointly considering the workload balance among computing nodes. First, a multiple offloading node selection algorithm is proposed to select appropriate PVs to take part in computing tasks. Furthermore, a workload allocation strategy based on the idea of dynamic game is presented to optimize system performance and consider the load balancing at the same time. Numerical results have demonstrated that the proposed offloading strategy in MPVEC can effectively reduce the system cost under delay constraint while achieving the load balancing of the system. In this work, only the computing resources of PVs are considered to optimize the system performance. In the future work, the communication resources allocation in multiple PV-assisted MEC will be researched. This study can be reviewed as a reference for task offloading in the vehicular network.

## Data Availability

All the data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the 2020 State Grid Corporation of China Science and Technology Program under Grant 5700-202041398A-0-0-00.

## References

- [1] Y. Ku, D. Y. Lin, C. F. Lee et al., "5G radio access network design with the fog paradigm: confluence of communications and computing," *IEEE Communications Magazine*, vol. 55, no. 4, pp. 46–52, 2017.
- [2] Y. Shih, W. Chung, A. Pang, T. Chiu, and H. Wei, "Enabling low-latency applications in fog-radio access networks," *IEEE Network*, vol. 31, no. 1, pp. 52–58, 2017.
- [3] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: a survey," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 450–465, 2018.
- [4] T. Ouyang, Z. Zhou, and X. Chen, "Follow me at the edge: mobility aware dynamic service placement for mobile edge computing," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 10, pp. 2333–2345, 2018.
- [5] B. Mohammed, M. Hamdan, J. S. Bassi et al., "Edge computing intelligence using robust feature selection for network traffic classification in internet-of-things," *IEEE Access*, vol. 8, pp. 224059–224070, 2020.
- [6] Y. R. B. al-Mayouf, N. F. Abdullah, O. A. Mahdi et al., "Real-time intersection-based segment aware routing algorithm for urban vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 7, pp. 2125–2141, 2018.
- [7] J. Xu, L. Chen, and P. Zhou, "Joint service caching and task offloading for Mobile edge computing in dense networks," in *IEEE INFOCOM 2018- IEEE Conference on Computer Communications*, pp. 207–215, Honolulu, HI, USA, 2018.
- [8] H. Shah-Mansouri and V. W. S. Wong, "Hierarchical fog-cloud computing for IoT systems: a computation offloading game," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3246–3257, 2018.
- [9] Y. Hung and C. Wang, "Fog micro service market: promoting fog computing using free market mechanism," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, Barcelona, Spain, 2018.

- [10] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: a viewpoint of vehicles as the infrastructures," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3860–3873, 2016.
- [11] Z. Wang, D. Zhao, M. Ni, L. Li, and C. Li, "Collaborative Mobile computation offloading to vehicle-based cloudlets," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 1, pp. 1–781, 2020.
- [12] J. Du, F. R. Yu, X. Chu, J. Feng, and G. Lu, "Computation offloading and resource allocation in vehicular networks based on dual-side cost minimization," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1079–1092, 2019.
- [13] J. Sun, Q. Gu, T. Zheng, P. Dong, and Y. Qin, "Joint communication and computing resource allocation in vehicular edge computing," *International Journal of Distributed Sensor Networks*, vol. 15, no. 3, 2019.
- [14] K. Nguyen, S. Drew, C. Huang, and J. Zhou, "EdgePV: collaborative edge computing framework for task offloading," in *ICC 2021 - IEEE International Conference on Communications*, Montreal, QC, Canada, 2021.
- [15] W. Qi, Q. Li, Q. Song, L. Guo, and A. Jamalipour, "Extensive edge intelligence for future vehicular networks in 6G," *IEEE Wireless Communications*, vol. 28, no. 4, pp. 128–135, 2021.
- [16] X. Huang, R. Yu, J. Liu, and L. Shu, "Parked vehicle edge computing: exploiting opportunistic resources for distributed Mobile applications," *IEEE Access*, vol. 6, pp. 66649–66663, 2018.
- [17] F. H. Rabman, A. Y. M. Iqbal, S. H. S. Newaz, A. T. Wan, and M. S. Ahsan, "Street parked vehicles based vehicular fog computing: Tcp throughput evaluation and future research direction," in *2019 21st International Conference on Advanced Communication Technology (ICACT)*, pp. 26–31, Pyeong-Chang, Korea (South), 2019.
- [18] Y. R. B. al-Mayouf, M. Ismail, N. F. Abdullah et al., "Efficient and stable routing algorithm based on user mobility and node density in urban vehicular network," *PLoS One*, vol. 11, no. 11, pp. 1–24, 2016.
- [19] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint offloading and resource allocation in vehicular edge computing and networks," in *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–7, Abu Dhabi, United Arab Emirates, 2018.
- [20] F. Lin, X. Lü, I. You, and X. Zhou, "A novel utility based resource management scheme in vehicular social edge computing," *IEEE Access*, vol. 6, pp. 66673–66684, 2018.
- [21] Z. Haitao, D. Yi, Z. Mengkang, W. Qin, S. Xinyue, and Z. Hongbo, "Multipath transmission workload balancing optimization scheme based on mobile edge computing in vehicular heterogeneous network," *IEEE Access*, vol. 7, pp. 116047–116055, 2019.
- [22] J. Zhang, H. Guo, J. Liu, and Y. Zhang, "Task offloading in vehicular edge computing networks: a load-balancing solution," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2092–2104, 2020.
- [23] J. Zhao, Q. Li, Y. Gong, and K. Zhang, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7944–7956, 2019.
- [24] P. Wang, C. Yao, Z. Zheng, G. Sun, and L. Song, "Joint task assignment, transmission, and computing resource allocation in multilayer mobile edge computing systems," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2872–2884, 2019.
- [25] K. Zhang, Y. Mao, S. Leng, S. Maharjan, and Y. Zhang, "Optimal delay constrained offloading for vehicular edge computing networks," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, Paris, France, 2017.
- [26] L. Zhang and N. Ansari, "Latency-aware IoT service provisioning in UAV-aided mobile-edge computing networks," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10573–10580, 2020.
- [27] Y. Liu, S. Xie, and Y. Zhang, "Cooperative offloading and resource management for UAV-enabled Mobile edge computing in power IoT system," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 12229–12239, 2020.
- [28] G. Qiao, S. Leng, K. Zhang, and Y. He, "Collaborative task offloading in vehicular edge multi-access networks," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 48–54, 2018.
- [29] R. Yadav, W. Zhang, O. Kaiwartya, H. Song, and S. Yu, "Energy-latency tradeoff for dynamic computation offloading in vehicular fog computing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14198–14211, 2020.
- [30] R. Yadav, W. Zhang, I. A. Elgendy et al., "Smart healthcare: RL-based task offloading scheme for edge-enable sensor networks," *IEEE Sensors Journal*, vol. 21, no. 22, pp. 24910–24918, 2021.
- [31] N. Liu, M. Liu, W. Lou, G. Chen, and J. Cao, "PVA in VANETs: stopped cars are not silent," in *2011 Proceedings IEEE INFOCOM*, pp. 431–435, Shanghai, China, 2011.
- [32] F. Malandrino, C. Casetti, C. Chiasserini, C. Sommer, and F. Dressler, "The role of parked cars in content downloading for vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 9, pp. 4606–4617, 2014.
- [33] C. Ma, J. Zhu, M. Liu, H. Zhao, N. Liu, and X. Zou, "Parking edge computing: parked-vehicle-assisted task offloading for urban VANETs," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 9344–9358, 2021.
- [34] Y. Li, B. Yang, Z. Chen, C. Chen, and X. Guan, "A Contract-Stackelberg Offloading Incentive Mechanism for Vehicular Parked-Edge Computing Networks," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, pp. 1–5, Kuala Lumpur, Malaysia, 2019.
- [35] X. Huang, R. Yu, D. Ye, L. Shu, and S. Xie, "Efficient workload allocation and user-centric utility maximization for task scheduling in collaborative vehicular edge computing," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 4, pp. 3773–3787, 2021.
- [36] K. Zhang, Y. Mao, S. Leng, A. Vinel, and Y. Zhang, "Delay constrained offloading for mobile edge computing in cloud-enabled vehicular networks," in *2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM)*, pp. 288–294, Halmstad, Sweden, 2016.

## Research Article

# NOMA and OMA-Based Massive MIMO and Clustering Algorithms for Beyond 5G IoT Networks

Taj Rahman <sup>1</sup>, Feroz Khan <sup>1</sup>, Inayat Khan <sup>2</sup>, Niamat Ullah <sup>2</sup>, Maha M. Althobaiti <sup>3</sup>,  
and Fawaz Alassery <sup>4</sup>

<sup>1</sup>Department of Computer Science, Qurtuba University of Science and Technology Peshawar, 25000, Pakistan

<sup>2</sup>Department of Computer Science, University of Buner, Buner, 19290, Pakistan

<sup>3</sup>Department of Computer Science, College of Computers and Information Technology, Taif University, Taif, Saudi Arabia

<sup>4</sup>Department of Computer Engineering, College of Computer and Information Technology, Taif University, PO Box. 11099, Taif 21994, Saudi Arabia

Correspondence should be addressed to Maha M. Althobaiti; maha\_m@tu.edu.sa

Received 30 August 2021; Accepted 29 October 2021; Published 20 November 2021

Academic Editor: Muhammad Shiraz

Copyright © 2021 Taj Rahman et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Internet of Things (IoT) has brought about various global changes, as all devices will be connected. This article examines the latest 5G solutions for enabling a massive cellular network. It further explored the gaps in previously published articles, demonstrating that to deal with the new challenges. The mobile network must use massive multiple input and output (MIMO), nonorthogonal multiple access (NOMA), orthogonal multiple access (OMA), signal interference cancellation (SIC), channel state information (CSI), and clustering. Furthermore, this article has two objectives such as (1) to introduce the cluster base NOMA to reduce the computational complexity by applying SIC on a cluster, which ultimately results in faster communication and (2) to achieve massive connectivity by proposing massive MIMO with NOMA and OMA. The proposed NOMA clustering technique working principle pairs the close user with the far user; thus, it will reduce computational complexity, which was one such big dilemma in the existing articles. This will specifically help those users that are far away from the base station by maintaining the connectivity. Despite NOMA's extraordinary benefits, one cannot deny the significance of the OMA; hence, the other objective of the proposed work is to introduce OMA with MIMO in small areas where the user is low in number, it is already in use, and quite cheap. The next important aspect of the proposed work is SIC, which helps remove interference and leads to enhancement in network performance. The simulation result has clearly stated that NOMA has gained a higher rate than OMA: current NOMA users' power requirement (weak signal user 0.06, strong signal user 0.07), spectral efficiency ratio for P-NOMA and C-NOMA (21%, 5%), signal-to-noise ratio OMA, P-NOMA, C-NOMA (28, 40, 55%), and user rate pairs NOMA, OMA (7, 3), C-NOMA, and massive MIMO NOMA SINR (4.0, 2.5).

## 1. Introduction

The Internet was considered as a network for connecting devices, such as desktop computers, laptops, routers, sensor nodes, smartphones, and home appliances [1]. The interaction between these devices via the Internet is described as the Internet of Things (IoT). The internet users are growing rapidly as approximately 1000-fold data traffic has been increased by 2020 [2]. Consequently, spectral efficiency could become a key challenge to control such explosive data

traffic [3]. Enhanced technologies have been the major need for satisfaction of these requirements [4]. Millimeter-wave communications, ultradense network, massive multiple-input and output (MIMO), and nonorthogonal multiple access (NOMA) have been proposed to address the 5th Generation challenges. At present, NOMA schemes have gotten more attention as compared to other multiple access techniques, which is further divided into 2 phases, that is, power domain multiplexing [5, 6] and code domain multiplexing, including multiple access with low-density spreading (LDS)

[7–9], sparse code multiple access (SCMA) [10], and multi-user shared access (MUSA) [11]. Some other multiple access schemes, such as pattern division multiple access (PDMA) and bit division multiplexing (BDM) [10], are also proposed. In [12], the author has put forward a low-complexity sub-optimal grouping user method. However, the mentioned method works by exploiting the channel gain difference among users in the NOMA cluster and gang them either single cluster or multiple clusters to improve system throughput. In [13], the author has clearly mentioned that the orthogonal multiple access (OMA) cannot thoroughly vanish; NOMA is a futuristic term that could facilitate the users in terms of rendering massive connectivity and capacity improvement. Nonetheless, this never means that NOMA could thoroughly replace OMA scheme in the coming 5G networks. OMA could be better for the specimen for a small network where the near and far effect is insignificant. On the other hand, NOMA would be better if the network is big. Thus, the futuristic 5G will have a combo of NOMA and OMA to fully fill the demands of various applications and services.

More importantly, even though NOMA can render attractive merits, some hurdles should be sorted, such as advanced transmitter design and the trade-off between performance and receiver complexity [14–16]. This research focuses on both NOMA and OMA multiple access techniques that utilize power domain/code domain and time/frequency domain, respectively. The purpose of the proposed combination of OMA and NOMA is, firstly, one cannot neglect OMA since it is already implemented in 4G and working properly. Secondly, the study has shown that NOMA is superior to OMA as NOMA renders 1 msec throughput compare to OMA, but it is costly. Thirdly, when the users are near the base station, then the spectral efficiency is better than NOMA. Therefore, it is directed to present a combo to utilize their advantages.

The main contribution of this article is elaborated in the next sentences. Till now, none of the authors from [13, 17–19] have proposed a combination of NOMA and OMA. Thus, the objective of this paper is to propose a combination of NOMA and OMA in order to achieve high performance. Further, the proposed work also introduces cluster base NOMA to reduce the complexity. Several articles have discussed about user 1 and user 2 due to the increase in users. Decoding each user's signal using signal interference cancellation (SIC) requires additional implementation complexity. Therefore, it is recommended to use clustering, helping diminish the additional complexity [14, 20]. Next, with the aid of SIC, one can avoid interference and achieve a low complexity objective. After that, channel state information (CSI) works as a backbone to SIC as it senses the weak users signal and strong users and allocates the power accordingly. Without perfect CSI, SIC decoding cannot be decided by base station (BS) directly. Thus, an explicit SIC decoding order must be acquired 1st.

The rest of the paper is organized as follows: Section 2 describes the related work. Section 3 presents a research methodology. Section 4 illustrates the simulations results. Section 5 presents the discussion, and finally, Section 6 concludes.

## 2. Related Work

*2.1. 5G Enabled IoT.* IoT has been considered an imperative for the coming services and application environment which is indeed of massive capacity, high volume of nodes, dense traffic with adaptable and even wider bandwidth from narrowband to broadband, very low latency, and energy-efficient design [21, 22]. For this reason, 5G plays a major role in enabling IoT due to disruptive improvements in the radio and antenna systems, spectrum, and network architecture [23]. 5G is known as 5th generation wireless technology, an unutilized network with a high data rate, trustworthy, and low latency than the previous generations. 5th generation has followed the footprints of 4G; the fifth-generation encoding type is OFDM [24]. 5G networks can work as low frequencies and high as “millimeter wave,” and that frequency can communicate a large amount of information/data, however, few blocks at a moment of time. 5G networks are further possibilities to be networks of minicells such as the size of a house router than to be a big tower; it is far from extending the network scope. The objective is to have extraordinary speed on hand and massive scope at low latency than 4G. The latency rate of 4G has been recorded near 50 milliseconds; however, 5G cut all the way down to almost one millisecond [25, 26], that is especially treasured for driverless vehicles and automatic programs. The motive of 5G is to attain transmission pace to 20-30 Gbps, which is 50 times faster than 4G networks [27]. And its speed has been being examined uninterrupted up to 1.5 Gbps while traveling 100 km/h and max up to 7.5 Gbps [28]. 5th generation network is determined to provide up to one million of connections per square kilometer. It also implies the entire wireless international interconnection with very high data rates [28, 29].

*2.2. Generation towards 5G.* After introducing the 5th generation wireless system in the previous section, this phase summarizes the comparison among mobile network generations. In [30], the authors have explained that 1G (Bell Labs) was introduced in the nineteen seventies and is based on analogue technology. The first generation (1G) communication medium used the frequency division multiplexing technique (FDMA), where the analogue signals were considered. The major fault in this analogue technology had a large size, poor voice, and battery. After this, the researchers [31] had invented the (2nd) generation in the late nineteen eighties having amazing features like the global system for mobile communication (GSM), and it was circuit switch, connection primly based technology, where the end system was dedicated for the whole call duration. As a result, it causes poor efficiency in the utilization of bandwidth and resources. This technology was also known as digital technology. Some of the negative points are digital technology that is lower data rate and inability. Another technology specifically developed for the marketing purpose and not officially described was 2.5 G [32]. With the increasing demands of users and technological development, the third generation (3G) was introduced by [25] that was based on extraordinary features such as dealing with the complex data, providing

high data rate, supporting video, audio, message, and improving the overall mean of communication. This phenomenal technology used the code division multiplexing technique (CDMA). It guaranteed the globe with an increment in bandwidth up to 2Mbps [33]. Finally, with the amazing feature, the 4th generation (4G) took place in 2011 by [34]. The requirement for the 4th generation is specified by International Military Tribunal (IMT-A) in 2009. 4th generation fulfill all the need of the users by providing the data rate up to 1Gbps, HD Mobile TV, enhanced audio, and video calls, etc. 4G is so far the biggest achievement in the cellular sector because of having tight security mechanism and assuring the personal user communication from the security point of view such as gaming services, internet usage, and streamed media. 4G is thoroughly based on coded orthogonal frequency division multiplexing (COFDM) and MIMO. The distinction between OMA and NOMA is depicted in Table 1.

**2.3. Multiple Access Techniques.** The multiple access technique is categorized into two parts. The first part is called orthogonal, and the second part is called nonorthogonal [34]. Both orthogonal and nonorthogonal use different access techniques. The orthogonal part uses frequency division multiple access (FDMA), time division multiple access (TDMA), and orthogonal FDMA (OFDMA) techniques, whereas the second part, nonorthogonal, uses code domain and power domain multiplexing. The main job of NOMA is serving multiple users at the same/time-frequency resources by assigning them various power levels. As per [35, 36], the orthogonal is better for the packet domain, having channel aware time and frequency schedule. As mentioned in the above comparison as shown in Table 1 of NOMA and OMA, one of the downsides of the NOMA is higher power requirement by a far user from the base station (BS) and (higher interference ratio) due to massive connectivity that ultimately leads to receiver complexity. NOMA's working principle is to serve multiple users at the same time/frequency by assigning different power levels; for the specimen, those users which are far away from the base station require more power to decode its information and maintain connectivity as compared to near user having strong connectivity. Thus, this causes complexity in the receiver and more power required. This can be avoided by using clustering and SIC with NOMA, which is discussed in detail in the methodology part in Sections 3.1, 3.2, 3.3, and 3.4 and detail view.

On the contrary, in OMA, every user can utilize orthogonal resources within a specific time slot, frequency band, or code to avoid multiple access interference, which definitely results in low power requirement and receiver complexity. Throughput of OMA is smaller due to rendering connectivity to the limited user and assigning resources for a specific time. As a result, many users have to wait until the first user is served, whereas NOMA serves multiple users at the same time by assigning them different power levels. To get clearer picture, the reader is suggested to go through the methodology part where each parameter has been discussed in detail.

OMA with NOMA as shown in Table 1 describes energy consumption, receiver complexity, user pairs, number of users in the cluster, and system throughput [37].

**2.4. Advantages of NOMA for IoT.** NOMA [38] has been found one of the most effective technologies in the telecommunication sector that will come up with certain benefits: To name a few of them such as high spectral efficiency, massive connectivity, low latency, quality of service, MIMO, NOMA with beam forming and MIMO, NOMA with radio and RA, and NOMA with clustering [39, 40].

NOMA is a vital enabling technology for 5G wireless networks because it allows them to meet heterogeneous criteria such as low latency, high dependability, huge connection, increased fairness, and high throughput [41]. NOMA is based on the idea of serving numerous users in the same resource block, such as a time slot, subcarrier, or spreading code. The NOMA principle is a broad framework, with numerous newly suggested 5G multiple access systems serving as examples [42]. The authors presented an overview of the latest NOMA and its various applications.

Extraordinary expectations for data speeds and capacity must be addressed beyond 5G networks [43]. The NOMA approach results in increased diversity gains, and huge interconnectedness could be a possibility to overcome these difficulties. One disadvantage of NOMA is the extra receiver complexity required to eliminate interuser interference (IUI) via SIC. In this way, the authors demonstrated how a cooperative relaying scheme could increase the NOMA system's total diversity gain and data rates. The cooperative NOMA system's user fairness and performance while implementing the irregular convolutional code are examined using extrinsic information transfer (EXIT) charts (IRCC). The suggested system's convergence analysis is evaluated utilizing the EXIT chart and IRCC [44, 45].

### 3. Research Methodology

The network is deployed based on the signal strength between the user and the base station. The nodes that occupy weaker signals describe that the nodes are far away from the BS, which requires additional power from the BS. Therefore, NOMA is considered here to connect nodes with the BS briefly described in Section 3.1. On the other hand, the nodes have a strong signal that requires less power for data communication. Thus, OMA is considered briefly described in Section 3.2. These considerations of NOMA and OMA ultimately provide massive connectivity without interruption and the least energy dissipation. Algorithms 1 and 2 are used for resource allocation and pairing users to reduce computational complexity, improve performance, and gain massive connectivity. Moreover, Section 3.2 describes OMA with MIMO that help render massive connectivity and reduce the chances of dropping connection. As a result, one can achieve higher spectral efficiency at a minimum cost. Furthermore, 3.3 and 3.6 illustrate NOMA with SIC works, helping avoid a collision often caused when two or more packets arrive simultaneously. 3.4 and 3.5 depict. Finally, Sections 3.7 and 3.8 highlight the latest

TABLE 1: Distinction between OMA and NOMA.

Specifications	OMA	NOMA
Full form	Orthogonal multiple access	Nonorthogonal multiple access
Energy consumption	Less	More
Receiver complexity	Low	High
Number of user pairs	More	Less
Number of users/clusters	Higher	Lower
System throughput (assumption: user fairness is guaranteed)	Smaller	Larger

```

# NOMA = Non-orthogonal Multiple Access
# SIC = Signal Interference Cancellation
# OMA=Orthogonal Multiple Access
Step 1: total 12 nodes randomly deploy
Step 2: User sends a request to BS for the allocation of resources
Step 3: Request processes to CSI
Step 4: CSI (Channel state information) calculates the distance
Step 4.1: if (distance >100 m)
    {
        Users get NOMA;
    }
Step: After allocation of NOMA (all the strong users will be paired up with weak users) and SIC will be applied to avoid inter user interference.
Step6: else
    {
        Users get OMA;
    }
Step7: end if.

```

ALGORITHM 1: Step by step working of user clustering.

parameters such as CSI and Massive MIMO, which helps in massive connectivity, allocating resources, and providing a free spectrum to carry out the transmission.

*3.1. Use of NOMA with Massive MIMO.* NOMA concept is based on power domain multiplexing assigns different power levels to the users based on higher and lower signals of the user, and users are distinguished based on their power levels while the prior technologies used to rely on code, frequency, and time-division multiplexing. The major problem with the OMA is low spectral efficiency, which normally causes when allocating resources like subcarrier channel to a user with poor CSI. However, in NOMA, users with poor channel state information CSI can also have access to all the resources like subcarrier by using the help of a strong CSI user, which ultimately results in high spectral efficiency. NOMA also uses superposition coding schemes at the transmitter side, such as the Success Interference Scheme (SIC), where the receiver can separate the users in downlink and uplink.

NOMA with massive MIMO has been proposed as one of the finest radio access technologies for the 5th generation mobile network. Massive MIMO is basically the upgraded version of MIMO, helping with NOMA in providing high spectral efficiency and throughput. MIMO has 2 to 4 antennas, whereas massive MIMO has more than 100 antennas that provide connectivity to massive users with high band-

width. Deployment of NOMA in a mobile network demands high computational power to implement real-time power allocation and successive interference cancellation algorithms. The deployment time of 5G is expected to be 2020, and it means that the computational capacity for both handset devices and access points is anticipated to be high enough to run NOMA algorithms.

The following algorithm shows resources allocation using NOMA and OMA.

*3.2. Detail View.* This side of the article gives a detailed view of Algorithms 1 and 2, DFD, and Sections 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7, and 3.8. Let us take a random basic 12 user's model [12] using SIC where all the users are clustered to reduce computational complexity. The pairing is based on the closeness of the users. The clustering mechanism is by pairing U1 with U2, U3 is with U4, and U5 is with U6, and so on. Signal interference cancellation is placed on the base station to avoid interuser interference and ensure guaranteed connectivity. As a result, larger throughput can be achieved. Further, the resources and power would be allotted to a particular user by calculating the maximum distance between the base station and using channel state information CSI. Moreover, it is vividly seen that users are multiplexed in the BS transmitter by power domain. The users are sorted in a cluster so that a user with poorer channel conditions will decode its information first than a cell edge



```

Step1: Total deployed 12
Step2: CSI calculate the distance of each user
Step3: if (distance >600 m)
    {
        User 12 attached with user 1;
    }
Step4: else if (distance >500 m)
    {
        User 11 attached with user 2;
    }
Step5: else if (distance >400 m)
    {
        User 10 attached with user 3;
    }
Step6: else if (distance >300 m)
    {
        User 9 attached with user 4;
    }
Step7: else if (cluster distance >200 m)
    {
        User 8 attached with user 5;
    }
Step8: else
    {
        User 7 attached with user 6;
    }
    End if

```

ALGORITHM 2: User clustering algorithm.

user with strong connectivity. Let us take an example of U1 and U2, U2 information will be decoded by the receiver using SIC with little loss after eliminating U1 interference, and then U1 information will be decoded by the receiver directly due to its high-power strength, which is quite easy to be captured. Furthermore, the steps of allocation resources and power are discussed following. First, the user requests the base station for the allocation of resources. The channel state information CSI will calculate the distance between the BS and the user [27]. After calculating the distance successfully, the power allocation and resources will be based on the far and close distance. For instance, U1 sent the request to the base station. As soon as the channel state information receives the request, it will calculate the distance. For example, the distance is 600 meters; so, the power and resources will be assigned accordingly.

In Algorithm 2, 12 users are randomly deployed [12], and the distance between the users and base station is 600 meters. Channel state information is used to calculate the distance between the base station and the user to pair it accordingly. For instance, user 12 is far from the BS; so, it will be paired with user 1 as user 1 is close to BS, and it has high power, which means that user 1 can easily decode the information of user 2 without any loss; hence, the high connectivity goal would be achieved. Moreover, if the distance is greater than 10, then user 11 will be paired with 2.

3.3. *Use of OMA with MIMO.* Here, the user deployment is based on the signal strength between the base station and

user. All users with strong signals and fewer chances of dropping connections will get resources from OMA. The reason for assigning OMA is to achieve better spectral efficiency with minimum cost. The working principle of OFDMA is quite smooth and straightforward. Here, each user is allocated separate channel and orthogonal resources in time, frequency, or code domain; hence, no interference exists due to orthogonal allocation of the resources, which will lead to spectral efficiency improvement. Next, by employing massive MIMO, we can have massive connectivity features and a strong connectivity rate.

3.4. *NOMA with SIC.* The working principle of SIC is as follows. The receiver mostly uses this method in a wireless data transmission which permits two or more than two packets decoding that arrive simultaneously (collision can be commonly found in a regular system due to the arrival of more packets at the same time). SIC is used to decode the stronger signal first at the receiver side, subtract that from the combined signal, and then decode the difference as the weaker signal.

The major characteristic of NOMA is serving multiple users at the same time/frequency/code, however, with different power levels, which yields a significant spectral efficiency. For the specimen, below, we have considered user 1 and user 2. Both have different power levels; so, the BS serves them by allocating them the same resource but differentiating them by assigning them different power levels. First, SIC detects the user 1 signals with low power and decodes. Then, user 2 will directly decode its signals as it is close to the base station and has high power compared to user 1. This process continues until the last user is left in the queue.

3.5. *User Clustering Scheme with Low-Complexity Suboptimal.* A user clustering scheme with low-complexity suboptimal has been proposed for the downlink NOMA in this part. This scheme develops the channel gain distinctions in users and aims to enhance the throughput of the considered cell.

To pair the high channel gain, users will always benefit from the low channel gain user, which will cause the enhancement of the throughput. The main purpose of doing this is to high channel grow user can attain a higher rate even with the low power levels while making a large fraction of power available for the weak user, ensuring the guaranteed connectivity. The key feature of this clustering scheme is downlink NOMA which is achieved by the highest pair channel gain user and the least channel gain user into the similar NOMA cluster, while the second highest channel retains user and the second lowest channel retains user into another NOMA cluster, and so on. Second, to utilize the services of OMA, it is recommended to use OMA for an area where a user is less in number, and the near-far effect does not matter. Figure 1 expresses the clustering process of NOMA and OMA using data flow diagram.

3.6. *Cluster Process Data Flow Diagram User Pairing Scheme.* Within the underneath information stream graph Figure 2, first, user requests to the base station for allocation of

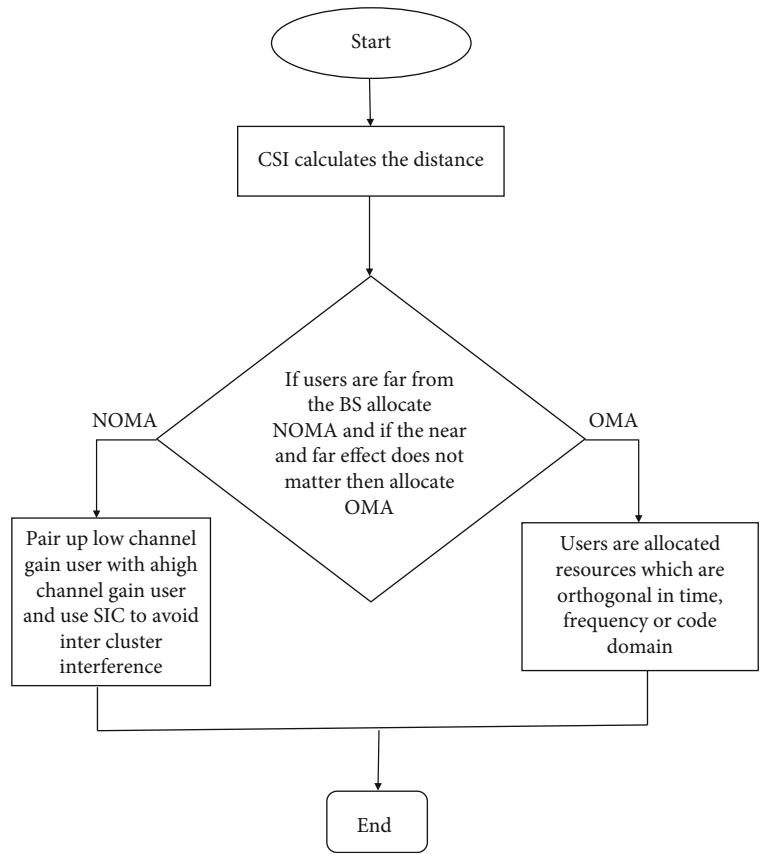


FIGURE 1: Clustering process of NOMA and OMA by using data flow diagram.

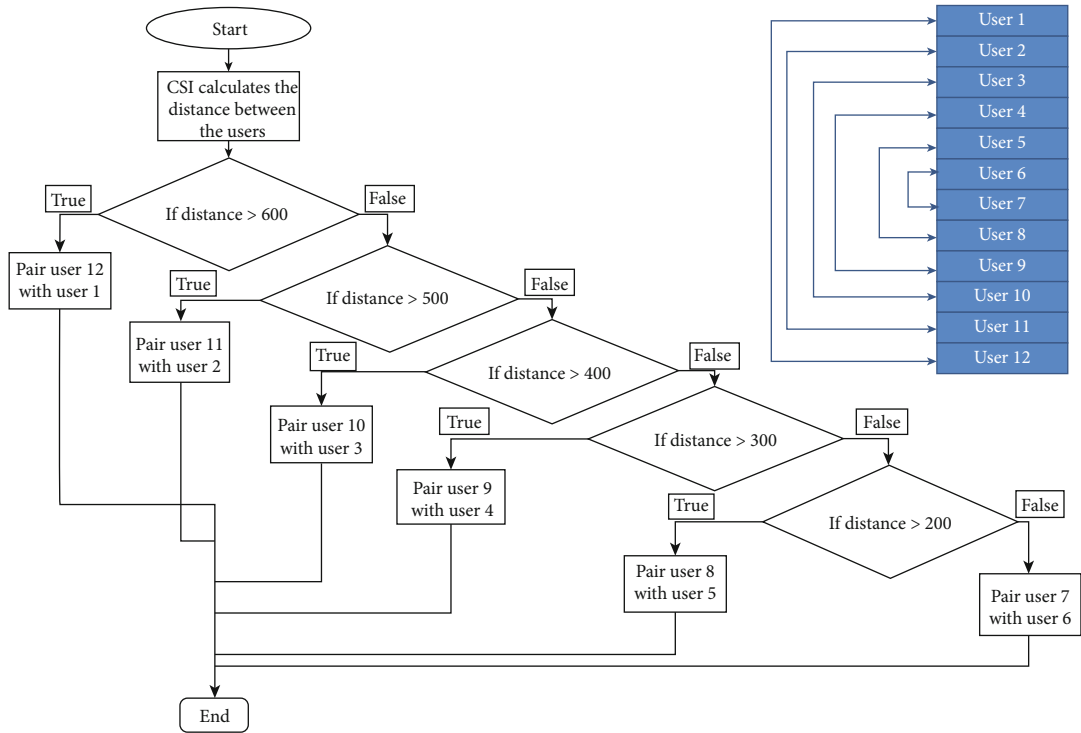


FIGURE 2: Cluster process flow chart.

resources. Once the BS receives the user request, the base station calculates the distance using CSI. If the user is far from the base station and has weak channel gain, BS will assign NOMA. The NOMA pairs the user having poor channel condition with a user having good channel condition. This way, both weak and robust users can utilize the channel simultaneously; however, if the near and far effect does not matter and the number of users is less, then BS will allocate OMA. The beauty of the OMA scheme is that radio resources can be allocated to multiple orthogonal users in frequency, time, or code domain. As a result, no interference occurs between users due to the orthogonally resources allocation.

**3.7. Successive Interference Cancellation.** Interference management is being considered the main cause of improvement in network capacity substantially. The SIC's main role is to enable users with the strongest signals to be sensed 1, hence, the least interference-contaminated signal. After this, signals are reencoded by a strongest user. As soon as reencoding is done, then these signals are subtracted from the composite signals. Now, this process is being followed by the 2nd strongest users' signals, which become strongest. When all these are done and the last user signal is sensed, the decoding of information by the weak user will not suffer from any kind of interference.

**3.8. Channel State Information (CSI).** CSI provides a communication link in the wireless communication world is used to propagate the signals from transmitter to receiver and represent the combined effects, such as power decay with distance, scattering, and fading. This whole process is known as channel estimation. High data rate and reliable communication in multiantenna systems can be gained by aiding CSI in adapting the transmission to current channel conditions.

CSI at the receiver and CSI at the transmitter are called channel state information receiver (CSIR) and channel state information transmitter (CSIT), respectively. The estimation of CSI is mandatory and often quantizes and feedback to the transmitter (though the reverse-link estimation is possible in the TDD system).

**3.9. Massive Multiple-Input and Multiple-Output (MIMO).** Massive MIMO includes multiple futuristic technologies, which provides the user with many antennas for smooth and interference-free communication, unlike prior technology, and the antennas were confined in numbers like 2 or 4, which causes delay and interference. Massive MIMO enables the concept of NOMA in providing interference-free and fast communications using these antennas. Therefore, the upcoming 5G will rely on NOMA along with the MIMO.

## 4. Simulation Results

This section is about simulation results of the current NOMA and proposed NOMA. The simulation experiment is conducted using MATLAB, one of the most prominent tools used for simulation results. In the simulation environ-

ment, users are randomly deployed in 200 to 600m areas. Data rates are set as 300 kbps and 2.4 GHz band. This section compares the current NOMA with proposed NOMA in power allocation, spectral efficiency, sum rate, SNR (OMA, P-NOMA and C-NOMA), and SINR. The parameters used for simulation and results are shown in Table 2.

**4.1. Current NOMA.** The current NOMA result is depicted using Figure 3. To compare the current NOMA with the aforementioned, we propose NOMA. At first, we randomly took total 7 users, which are shown using "x-axis," and then on "y-axis," we use parameter power allocation to see the performance difference between C-NOMA and P-NOMA. By doing this, we analyze that in current NOMA, the power is almost equally assigned to both strong and weak users, causing performance degradation. In other words, the big dilemma in the current system is treating weak and strong user equally in terms of power allocation due to the imperfect channel state information CSI that not only effect on performance but also cause complexity. As per the below experiment, the minimum power used by strong and weak users is 0.01-0.02, and the maximum power by both users is 0.06-0.07.

**4.2. Proposed NOMA.** Figure 4 illustrates the power allocation mechanism of the proposed NOMA. On the "x-axis" number of users and "y-axis" power allocation, as per the simulation result below, one can easily understand the proposed NOMA advantage over the current NOMA. The working principle of the proposed NOMA is by allocating more power to the user having weak signal and possibly dropping connection. In Figure 4, the blue line denotes the strong user channel power requirement, and the red line denotes the weak user power requirement. The power allocation to each user is carried out based on the signal strength and distance of the user from the base station BS. If the user signals are weak, it will require more power to decode its information using SIC without losing the connection, which ultimately leads to helping in achieving high connectivity. The minimum power required by the strong user is 0.02, and the maximum is 0.07, whereas in a weak user, the minimum power is 0.03, and the maximum is 0.1.

**4.3. P-NOMA vs. C-NOMA.** Figure 5 compares current NOMA with proposed NOMA. Total 10 users have been clustered in P-NOMA to analyze the spectral efficiency. The result has clearly shown that as the number of clusters increases in P-NOMA, the spectral efficiency also increases, whereas in C-NOMA, spectral efficiency decreases with the increment of users. Moreover, the figure explains that the P-NOMA spectral efficiency is double C-NOMA, the most significant advantage. In contrast to C-NOMA, users without clusters have 5% spectral efficiency, whereas P-NOMA, a cluster user has a 21% spectral efficiency rate. This confirms that P-NOMA outperforms C-NOMA.

**4.4. C-NOMA, P-NOMA, and OMA.** Figure 6 analyzes C-NOMA, P-NOMA, and OMA. Signal-to-noise ratio parameter used to measure the interference ratio. By looking at the below figure, one can see the noise ratio of C-NOMA, which

TABLE 2: Parameters used for the simulation results.

Parameter	Value
Intersite distance	200 m, 600 m
Carrier frequency	2.4 GHz
Bandwidth per sub channel	180GHZ
Subchannel ( $N$ )	5, 10
Noise power	173 dBm
Number of transmitter antenna	64
Number of receiver antenna	64
Number of users per cluster	2
Algorithm	Low complexity suboptimal
Total transmission power	44 dBm (25w) for ISD = 200 m 49 dBm (80w) for ISD = 600 m
BS antenna height	10 m for ISD = 200, 32 m for ISD = 600 m
User equipment UE height	1.5 m
Minimum distance between UE and cell	35 m
Data rate	300 kbps

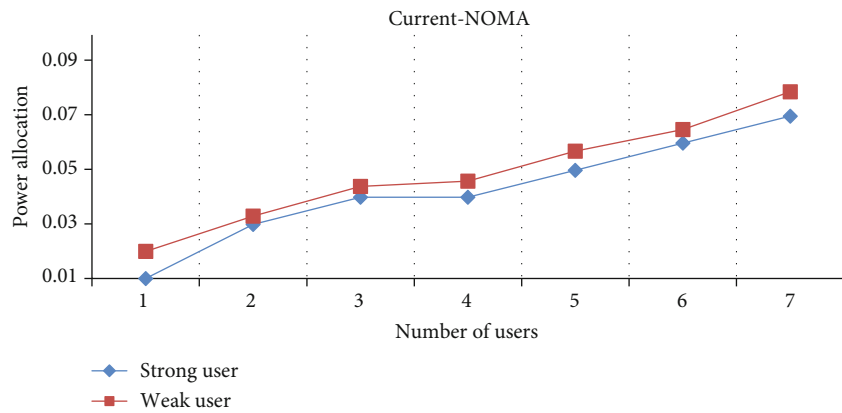


FIGURE 3: Current NOMA.

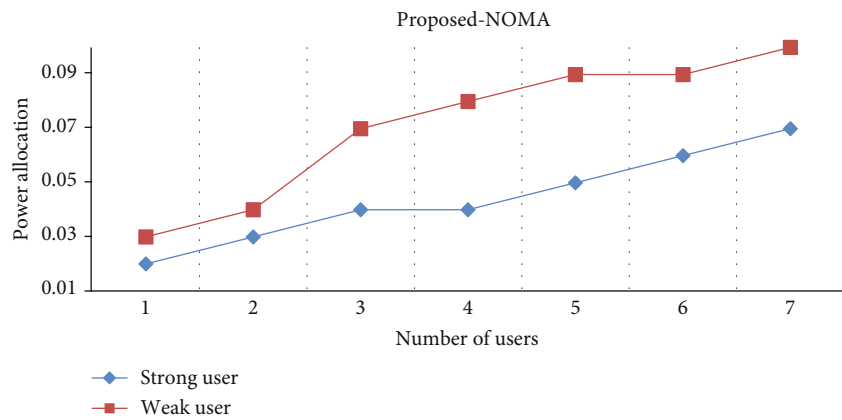


FIGURE 4: Proposed NOMA.

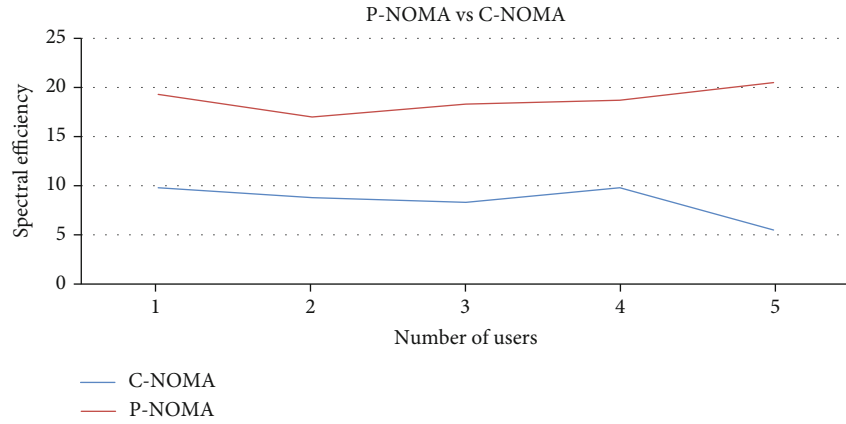


FIGURE 5: P-NOMA vs. C-NOMA spectral efficiency.

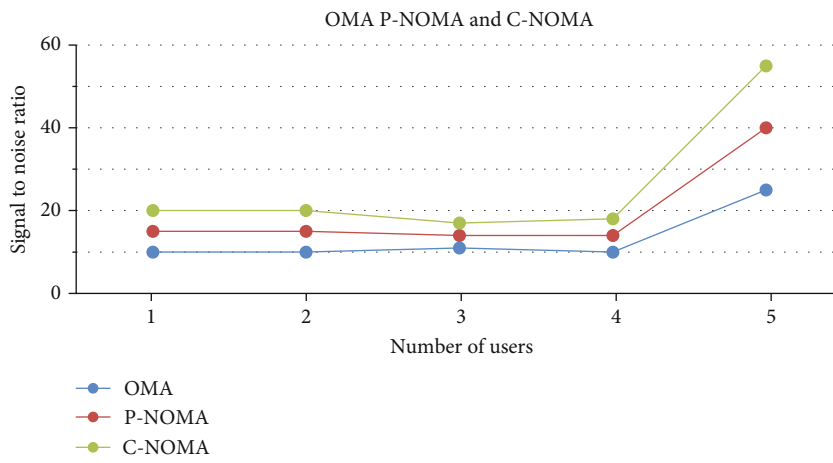


FIGURE 6: OMA P-NOMA and C-NOMA signal-to-noise ratio.

is higher than the other two, means that the OMA has a clear benefit over NOMA. Therefore, we cannot deny the significance of OMA. Besides, the evident results clearly state the ratio of noise for all the three 10, 15, and 20 percent, respectively; nonetheless, as the user increase, the ratio of noise also increases, and at the end, the value is reported 28, 40, and 55%, which means OMA services cannot be neglected.

**4.5. Rate Pairs.** Two users have been taken into the network to analyze the boundary of the attainable rate region for these users. Here, we are considering an asymmetric down-link channel so that the users are at equal distance to the BS:  $SNR_1 = SNR_2 = 10$  dB. Figure 7 depicts the boundary of available rate regions R1 and R2 for the NOMA and OFDMA. As demonstrated in the figure, NOMA obtains higher rate pairs than the OFDMA because of low fairness. Hence, it is certain that by looking to start where both start with almost 0 percent and end with 3 and 7% percent. NOMA can work better than OMA when users are clustered, and the following result has proved it.

**4.6. Massive MIMO NOMA User Cluster SINR vs. Current NOMA SINR.** The comparison of signal interference noise ratio between massive MIMO NOMA cluster and current

NOMA is being shown in Figure 8. Total power is set 30 dBm in simulation, and users are set as 12 for MIMO NOMA cluster (2 users per cluster) and 6 for current NOMA. During decoding information, the current NOMA signal-to-noise ratio has been recorded at the highest minimum 2.5 and a maximum of nearly 4.

The current NOMA users face a high interference ratio that causes computational complexity as signal interference cancellation SIC will be applied on each user to cancel the noise ratio. The more the user transmits the data, the more time SIC would take to cancel/reduce the noise ratio. On the other hand, the massive MIMO NOMA signal interference ratio is a minimum 2 while the maximum 2.5 clearly states the superiority of massive MIMO NOMA. Here, the SIC is applied on the whole cluster instead of an individual user to cancel the noise, which ultimately helps in low computational complexity. The SINR rate achieved by MIMO NOMA cluster is shown in Figure 9.

## 5. Discussion

First, by allocating power to the weak and strong user in propose NOMA, we have found that a weak user requires more power than a strong user to maintain connectivity since it

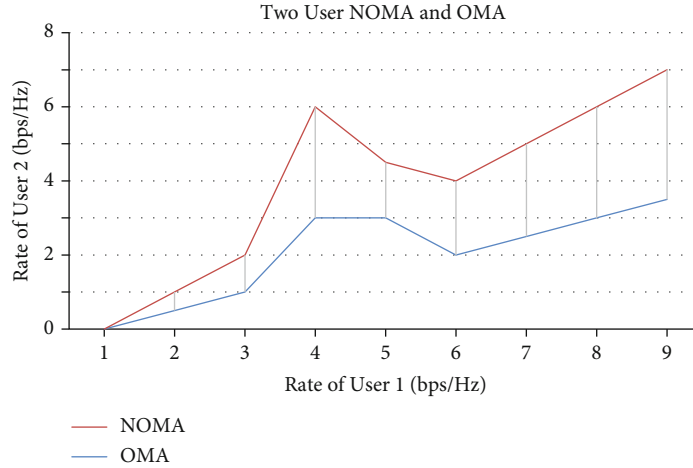


FIGURE 7: NOMA and OFDMA pair rate for downlink NOMA, i.e., SNR1 = SNR2 = 10 dB.

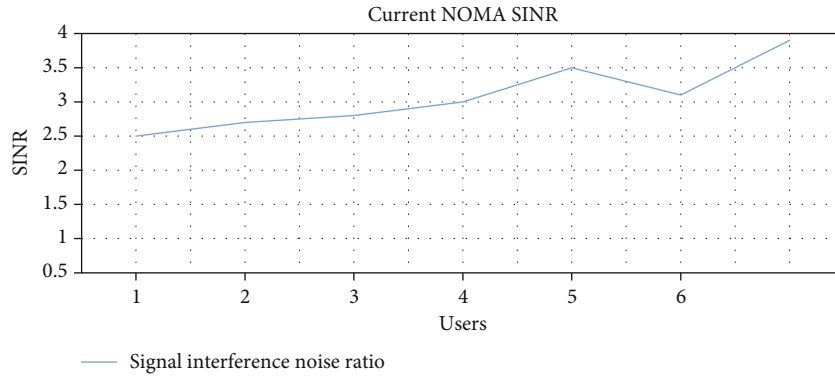


FIGURE 8: SINR rate achieved by current NOMA.

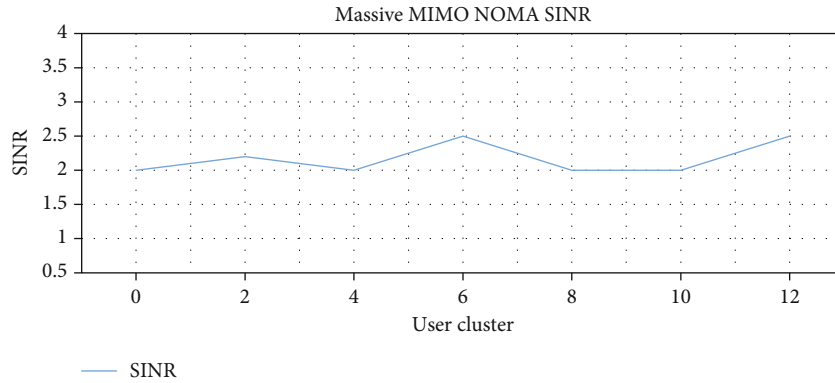


FIGURE 9: SINR rate achieved by MIMO NOMA cluster.

has greater chances of losing connection. However, in the current NOMA, users are allocated equal power, which leads to low-performance issues. Therefore, the proposed NOMA system is suggested for the future 5G challenges as it helps in gaining high performance compared to the current NOMA. Second, by comparing P-NOMA with C-NOMA, we found that spectral efficiency could be gained with P-NOMA because the power would be allocated to the whole cluster

rather than individual user, which automatically led to spectral efficiency gain. On the other hand, in C-NOMA, each user requires power to decode its information separately, leading to low spectral efficiency. Third, by taking the signal-to-noise ratio in P-NOMA, C-NOMA, and OMA, we have clearly seen that the noise ratio is quite high in P-NOMA and C-NOMA compared to OMA service of OMA cannot be ignored. Fourth, two users are taken in a network

to check the pairing rate by considering downlink using NOMA and OMA. Finally, current NOMA and massive MIMO NOMA with SINR ratio are being tested. The massive MIMO NOMA, noise ratio was far less than the current NOMA signal inference ratio. By considering the following results of P-NOMA users' power requirement (strong signal users power consumption 0.07, weak user power consumption 0.1), current NOMA users' power requirement (weak signal user 0.06, strong signal user 0.07), spectral efficiency ratio for P-NOMA and C-NOMA (21%, 5%), signal-to-noise ratio OMA, P-NOMA, and C-NOMA (28, 40, 55%), user rate pairs NOMA, OMA (7, 3), and C-NOMA, and massive MIMO NOMA SINR (4.0, 2.5), the simulation result has clearly stated that NOMA has gained a higher rate than OMA. Thus, it is clear by taking all the simulation results into account that the PROPOSE-NOMA will render certain advantages such as high connectivity, better spectral efficiency, and less interference ratio. 5G is incomplete. Without taking these new parameters, one cannot attain the objective of higher spectral efficiency and reduction in computational complexity, which has been proved by the above results that these parameters must be considered.

## 6. Conclusion

In this article, some of the prior problems have been addressed with the solution, like signal-to-noise ratio is one of the most significant factors in 5G. The key objective behind the introduction of SIC is to avoid interference between users, ultimately leading to enhancement in spectral efficiency. Therefore, the proposed NOMA with SIC and OMA with MIMO will ensure the connectivity of more than one user simultaneously without interference and help provide massive connectivity. Aside from that, the advent of clustering will benefit 5G by grouping users into clusters. As a result, it will reduce computational complexity, hence avoiding computational complexity and improving spectral efficiency. These parameters must be considered (such as NOMA with MIMO, NOMA with OMA, SIC, and clustering).

## Data Availability

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

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

We deeply acknowledge Taif University for supporting this research through Taif University Researchers Supporting Project Number (TURSP-2020/328), Taif University, Taif, Saudi Arabia.

## References

- [1] M. Masoud, Y. Jaradat, A. Manasrah, and I. Jannoud, "Sensors of smart devices in the internet of everything (IoE) era: big opportunities and massive doubts," *Journal of Sensors*, vol. 2019, 26 pages, 2019.
- [2] V.-D. Nguyen, T. Q. Duong, and Q.-T. Vien, "Editorial: emerging techniques and applications for 5G networks and beyond," *Mobile Networks and Applications*, vol. 25, no. 5, pp. 1984–1986, 2020.
- [3] I. Khan, M. A. Khan, S. Khusro, and M. Naeem, "Vehicular lifelogging: issues, challenges, and research opportunities," *Journal of Information Communication Technologies and Robotics Applications*, vol. 8, pp. 30–37, 2017.
- [4] F. Boccardi, R. W. Heath, A. Lozano, T. L. Marzetta, and P. Popovski, "Five disruptive technology directions for 5G," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 74–80, 2014.
- [5] S. Han, I. Chih-Lin, Z. Xu, and Q. Sun, "Energy efficiency and spectrum efficiency co-design: from NOMA to network NOMA," *IEEE COMSOC MMTC E-Letter*, vol. 9, 2014.
- [6] Y. Saito, Y. Kishiyama, A. Benjebbour, T. Nakamura, A. Li, and K. Higuchi, "Non-orthogonal multiple access (NOMA) for cellular future radio access," in *2013 IEEE 77th vehicular technology conference (VTC Spring)*, pp. 1–5, Dresden, Germany, 2013.
- [7] M. Al-Imari, P. Xiao, M. A. Imran, and R. Tafazolli, "Uplink non-orthogonal multiple access for 5G wireless networks," in *2014 11th international symposium on wireless communications systems (ISWCS)*, pp. 781–785, Barcelona, Spain, 2014.
- [8] R. Hoshyar, F. P. Wathan, and R. Tafazolli, "Novel low-density signature for synchronous CDMA systems over AWGN channel," *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1616–1626, 2008.
- [9] I. Khan, S. S. Rizvi, S. Khusro, S. Ali, and T.-S. Chung, "Analyzing drivers' distractions due to smartphone usage: evidence from AutoLog dataset," *Mobile Information Systems*, vol. 2021, 14 pages, 2021.
- [10] H. Nikopour and H. Baligh, "Sparse code multiple access," in *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 332–336, London, UK, 2013.
- [11] M. Shirvanimoghaddam, M. Dohler, and S. J. Johnson, "Massive non-orthogonal multiple access for cellular IoT: potentials and limitations," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 55–61, 2017.
- [12] M. S. Ali, H. Tabassum, and E. Hossain, "Dynamic user clustering and power allocation for uplink and downlink non-orthogonal multiple access (NOMA) systems," *IEEE access*, vol. 4, pp. 6325–6343, 2016.
- [13] L. Dai, B. Wang, Y. Yuan, S. Han, I. Chih-Lin, and Z. Wang, "Non-orthogonal multiple access for 5G: solutions, challenges, opportunities, and future research trends," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 74–81, 2015.
- [14] S. R. Islam, N. Avazov, O. A. Dobre, and K.-S. Kwak, "Power-domain non-orthogonal multiple access (NOMA) in 5G systems: potentials and challenges," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 721–742, 2017.
- [15] W. U. Khan, F. Jameel, X. Li, M. Bilal, and T. A. Tsiftsis, "Joint spectrum and energy optimization of NOMA-enabled small-cell networks with QoS guarantee," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 8, pp. 8337–8342, 2021.

- [16] I. Khan and S. Khusro, "Towards the Design of Context-Aware Adaptive User Interfaces to minimize drivers' distractions," *Mobile Information Systems*, vol. 2020, 23 pages, 2020.
- [17] S. Borkar and H. Pande, "Application of 5G next generation network to internet of things," in *2016 International Conference on Internet of Things and Applications (IOTA)*, pp. 443–447, Pune, India, 2016.
- [18] Y. Cheng, K. H. Li, Y. Liu, K. C. Teh, and G. K. Karagiannidis, "Non-orthogonal multiple access (NOMA) with multiple intelligent reflecting surfaces," *IEEE Transactions on Wireless Communications*, vol. 20, no. 11, p. 1, 2021.
- [19] P. Sharma, A. Kumar, and M. Bansal, "Performance analysis for user selection-based downlink non-orthogonal multiple access system over generalized fading channels," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 11, article e4347, 2021.
- [20] I. Khan, S. Khusro, N. Ullah, and S. Ali, "AutoLog: toward the design of a vehicular lifelogging framework for capturing, storing, and visualizing lifebits," *IEEE Access*, vol. 8, pp. 136546–136559, 2020.
- [21] M. Asif, W. U. Khan, H. Afzal et al., "Reduced-complexity LDPC decoding for next-generation IoT networks," *Wireless Communications and Mobile Computing*, vol. 2021, 10 pages, 2021.
- [22] T. Rahman, Z. Zhou, and H. Ning, "Energy efficient and accurate tracking and detection of continuous objects in wireless sensor networks," in *2018 IEEE International Conference on Smart Internet of Things (SmartIoT)*, pp. 210–215, Xi'an, China, 2018.
- [23] S. Yu, J. Liu, J. Wang, and I. Ullah, "Adaptive double-threshold cooperative spectrum sensing algorithm based on history energy detection," *Wireless Communications and Mobile Computing*, vol. 2020, 12 pages, 2020.
- [24] A. A. Zaidi, R. Baldemair, V. Molés-Cases, N. He, K. Werner, and A. Cedergren, "OFDM numerology design for 5G new radio to support IoT, eMBB, and MBSFN," *IEEE Communications Standards Magazine*, vol. 2, no. 2, pp. 78–83, 2018.
- [25] A. A. Salih, S. Zeebaree, A. S. Abdullaheem, R. R. Zebari, M. Sadeeq, and O. M. Ahmed, "Evolution of mobile wireless communication to 5G revolution," *Technology Reports of Kansai University*, vol. 62, pp. 2139–2151, 2020.
- [26] R. Khan, Q. Yang, I. Ullah et al., "3D convolutional neural networks based automatic modulation classification in the presence of channel noise," *IET Communications*, 2021.
- [27] B. D. Payal and P. Kumar, "Research based study on evolution of cellular generations (5G)," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 3, pp. 7522–7525, 2014.
- [28] I. Ullah, S. Qian, Z. Deng, and J.-H. Lee, "Extended Kalman filter-based localization algorithm by edge computing in wireless sensor networks," *Digital Communications and Networks*, vol. 7, no. 2, pp. 187–195, 2021.
- [29] I. Khan, S. Khusro, S. Ali, and A. U. Din, "Daily life activities on smartphones and their effect on battery life for better personal information management: smartphones and their effect on battery life for better personal information management," *Proceedings of the Pakistan Academy of Sciences: A. Physical and Computational Sciences*, vol. 53, 2016.
- [30] A. Agarwal and K. Agarwal, "The next generation mobile wireless cellular networks—4G and beyond," *American Journal of Electrical and Electronic Engineering*, vol. 2, no. 3, pp. 92–97, 2014.
- [31] S. Kumar, S. Pandey, N. Thakur, and G. Singh, *Channel Modeling of 5th Generation Communication Technology*, 2016.
- [32] M. A. Al-Absi, A. A. Al-Absi, M. Sain, and H. J. Lee, "A state of the art: future possibility of 5G with IoT and other challenges," *Smart Healthcare Analytics in IoT Enabled Environment*, P. Pattnaik, S. Mohanty, and S. Mohanty, Eds., pp. 35–65, 2020.
- [33] S. Patel, V. Shah, and M. Kansara, "Comparative study of 2G, 3G and 4G," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 3, pp. 1962–1964, 2018.
- [34] A. Araujo and I. Urizar, "4G technology: the role of telecom carriers," in *Dynamics of Big Internet Industry Groups and Future Trends*, pp. 201–241, Springer, 2016.
- [35] M. Z. Hassan, M. J. Hossain, J. Cheng, and V. C. Leung, "Joint throughput-power optimization of fog-RAN using rate-splitting multiple access and reinforcement-learning based user clustering," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 8, pp. 8019–8036, 2021.
- [36] L. Anxin, L. Yang, C. Xiaohang, and J. Huiling, "Non-orthogonal multiple access (NOMA) for future downlink radio access of 5G," *China Communications*, vol. 12, Supplement, pp. 28–37, 2015.
- [37] J. Xiang, Z. Zhou, L. Shu, T. Rahman, and Q. Wang, "A mechanism filling sensing holes for detecting the boundary of continuous objects in hybrid sparse wireless sensor networks," *IEEE Access*, vol. 5, pp. 7922–7935, 2017.
- [38] Y. Liu, Z. Qin, M. El-kashlan, Z. Ding, A. Nallanathan, and L. Hanzo, "Non-orthogonal multiple access for 5G and beyond," 2018, <https://arxiv.org/abs/1808.00277>.
- [39] K. Higuchi and A. Benjebbour, "Non-orthogonal multiple access (NOMA) with successive interference cancellation for future radio access," *IEICE Transactions on Communications*, vol. E98.B, no. 3, pp. 403–414, 2015.
- [40] W. U. Khan, N. Imtiaz, and I. Ullah, "Joint optimization of NOMA-enabled backscatter communications for beyond 5G IoT networks," *Internet Technology Letters*, vol. 4, no. 2, article e265, 2021.
- [41] Z. Ding, X. Lei, G. K. Karagiannidis, R. Schober, J. Yuan, and V. K. Bhargava, "A survey on non-orthogonal multiple access for 5G networks: research challenges and future trends," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 10, pp. 2181–2195, 2017.
- [42] A. Ahmed, Z. Elsaraf, F. A. Khan, and Q. Z. Ahmed, "Cooperative non-orthogonal multiple access for beyond 5G networks," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 990–999, 2021.
- [43] V. Basnayake, D. N. K. Jayakody, V. Sharma, N. Sharma, P. Muthuchidambaranathan, and H. Mabed, "A new green prospective of non-orthogonal multiple access (noma) for 5g," *Information*, vol. 11, no. 2, p. 89, 2020.
- [44] A. Akbar, S. Jangsher, and F. A. Bhatti, "NOMA and 5G emerging technologies: a survey on issues and solution techniques," *Computer Networks*, vol. 190, article 107950, 2021.
- [45] I. Khan, S. Ali, and S. Khusro, "Smartphone-based lifelogging: an investigation of data volume generation strength of smartphone sensors," in *International Conference on Simulation Tools and Techniques*, pp. 63–73, Chendu, China, 2019.



## Research Article

# Optimization Strategy of Task Offloading with Wireless and Computing Resource Management in Mobile Edge Computing

Xintao Wu <sup>1</sup>, Jie Gan <sup>2</sup>, Shiyong Chen <sup>1</sup>, Xu Zhao <sup>2</sup> and Yucheng Wu <sup>1</sup>

<sup>1</sup>School of Microelectronics and Communication Engineering, Chongqing University, Chongqing, China

<sup>2</sup>Beijing Smart-Chip Microelectronics Technology Co., Ltd., China

Correspondence should be addressed to Shiyong Chen; [chensy@cqu.edu.cn](mailto:chensy@cqu.edu.cn)

Received 12 August 2021; Accepted 9 October 2021; Published 11 November 2021

Academic Editor: Dr. Saba Bashir

Copyright © 2021 Xintao Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Mobile edge computing (MEC) provides user equipment (UE) with computing capability through wireless networks to improve the quality of experience (QoE). The scenario with multiple base stations and multiple mobile users is modeled and analyzed. The optimization strategy of task offloading with wireless and computing resource management (TOWCRM) in mobile edge computing is considered. A resource allocation algorithm based on an improved graph coloring method is used to allocate wireless resource blocks (RBs). The optimal solution of computing resource is obtained by using KKT conditions. To improve the system utility, a semi-distributed TOWCRM strategy is proposed to obtain the task offloading decision. Theoretical simulations under different system parameters are executed, and the proposed semi-distributed TOWCRM strategy can be completed with finite iterations. Simulation results have verified the effectiveness of the proposed algorithm.

## 1. Introduction

With the continuous development of the Internet of things and ubiquitous computing, mobile devices are increasingly running resource-intensive applications, such as interactive games and augmented reality [1, 2]. However, the limited resources of mobile devices cannot fully meet the requirements of these applications for powerful computing power and high speed. In recent years, many solutions have been proposed to solve the problem. In particular, mobile edge computing (MEC) provides a new way for UEs to complete computing tasks. MEC allows user equipment (UE) to offload computing tasks to network edge nodes through the wireless cellular network and performs the offloading tasks. This not only satisfies the expansion demand of users' computing capabilities but also compensates for the long delay of cloud computing [3]. It is a good method by using small base stations (SBSs) to meet the data rate demand of applications [4, 5]. As one of the key components of 5G, SBSs can enhance the coverage of local hot spots and increase system capacity. Dense network deployment can improve spectrum utilization and reduce end-to-end delay [6, 7].

However, task offloading not only generates additional overhead but also may cause intercell interference as it shares the same wireless frequencies among small cells, which will significantly influence the performance of the network [8]. Therefore, a reasonable offloading decision and interference management become the key to achieve efficient computation offloading [9]. A lot of works have been devoted to the research of computation offloading. Most of them have only focused on the process of offloading computing tasks from UE to MEC [10–18]. Only the optimal offloading decision is considered in [10, 11]. Researchers only focused on optimizing the communication resources [12, 13] or the computing resources [14, 15]. In some works, the combination of optimizing offloading decisions and resource allocation is used to minimize the latency or enhance the system performance [16–18]. Recently, research works by combining task offloading and interference management are proposed to improve the system utility [9, 19–21]. However, the scene of one user per base station is studied in [19, 20]. The work about wireless resource allocation does not take the minimum transmission rate requirement of each user into account [9].

The mobile devices can gather air quality data to analyze the environmental pollution or collect the image data to realize personal identity authentication from monitoring equipment. The MEC server determines whether the task is processed locally or offloaded to the server according to the computing capacity of the mobile device, the size of data, the delay, and the energy consumption requirements. The main contributions in this article are as follows:

- (i) The communication model and the computing model in a multibase station and multiuser MEC scenario are described. The delay and energy consumption in local or remote computing are analyzed
- (ii) The user utility is modeled as the weighted sum of the delay ratio and energy consumption ratio. And the system utility is defined as the sum of all user utilities. The optimization of the system utility is formulated by combining task offloading, wireless resource allocation, and computing resource allocation
- (iii) The optimal goal is decomposed into three subproblems including wireless resource block allocation (RBA), computing resource allocation (CRA), and task offloading decision. The RBA is solved by using a resource allocation algorithm based on an improved graph coloring method. The optimal solution of CRA is obtained by using KKT conditions. In task offloading, a semi-distributed task offloading with wireless and computing resource management (TOWCRM) strategy is proposed to optimize the system utility under the constraints of computing resources

The rest of this article is organized as follows: the related works are discussed in Section 2. The system model with multiple cells and multiple users in the MEC scenario is described in Section 3. The optimization of the system utility is formulated in Section 4. In Section 5, wireless resource optimization and computing resource allocation are discussed. A semi-distributed TOWCRM algorithm is proposed to optimize offloading tasks. The simulation results are given and discussed in Section 6. The conclusion of this work is described in Section 7.

## 2. Related Works

Edge computing could be affected by external environment (such as wireless channel, interferences among mobile users, communication link quality, and the status of the communication channel) during offloading [22]. Therefore, it is very important to establish a suitable environment of offloading policy for computation offloading. In [10, 11], these studies only paid attention to task offloading without optimizing communication and computing resources. It was assumed that the capacity of cloud computing is unlimited, and some studies only focused on the optimization of communication resources in [12, 13]. For instance, to maximize the network

management profit, an optimal solution algorithm based on the idea of branch-and-price was put forward to address joint resource management for device-to-device (D2D) communication [12]. Based on combining resource allocation and task assignment, a low-complexity iteration algorithm was proposed to minimize the task execution latency of all users subject to task and resource constraints in [13]. In contrast, only computing resource was optimized during task offloading [14, 15]. A new market-based framework was proposed to efficiently allocate computing resources of heterogeneous capacity-limited edge nodes (EN) for multiple competing services at the network edge in [14]. In [15], a smart contract that exploited the state-of-the-art machine learning algorithm was used in a private blockchain network to allocate the edge computing resources. In [16–18], joint communication and computing resource optimization were considered during the task offloading. To minimize the average latency of users to complete tasks, a strongly nonconvex problem with coupled variables was described as jointly considering the offloading decision, computation, and broadband resource allocation [16]. In [17], the problem of joint service caching, computation offloading, transmission, and computing resource allocation in a scenario of multiple users with multiple tasks was formulated to minimize the overall computation and delay costs. Moreover, the scenario where each user had a computation cost constraint was studied. A semi-distributed heuristic offloading decision algorithm (HODA) was proposed to maximize the system utility, which jointly optimized the offloading decision, communication, and computing resources [18].

In addition, there have been also some works that consider the joint optimization of task offloading and interference management at the same time [19–21]. Task offloading was studied in a MEC scenario with a single user per cell in [19, 20]. For example, offloading decision was made by considering the effect of intercell interference on system performance, where physical resource block (PRB) and computing resource allocation were treated as a joint optimization problem. The MEC server made the offloading decision to maximize the overhead, and the PRB was allocated by using a graph coloring algorithm [19]. In [21], the problem of joint task offloading and resource allocation was studied to maximize the offloading utility, which was modeled by the weighted sum of task completion time and device energy consumption. The resource allocation (RA) problem using convex and quasiconvex optimization was addressed, and a novel heuristic algorithm was proposed to solve the task offloading. It could achieve a suboptimal solution in polynomial time. However, there was no consideration to minimize interferences among mobile users.

## 3. System Model

This section describes the system model used in our work. Firstly, the network model is introduced in detail. Then, the corresponding communication model and calculation model are derived based on the proposed network model. For simplicity, the key notations used in the article are summarized in Table 1.

TABLE 1: Summary of key notations.

Notation	Description
$\mathcal{S}$	Set of SBSs
$\mathcal{U}_s$	Set of UEs in the coverage area of $s$
$\mathcal{X}$	The task offloading decision
$\mathcal{Y}$	The RB association strategy
$\mathcal{F}$	Computing resource allocation policy
$\mathcal{N}$	Set of RBs
$B$	The bandwidth of every RB
$x_{u_s^m}$	The offloading variable
$y_{u_s^m}^n$	RB assigned variable
$I_{u_s^m}^n$	The interference intensity
$P_{u_s^m}$	The transmission power of $u_s^m$
$K_{u_s^m}$	The number of RBs assigned to $u_s^m$
$H_{u_s^m, s}$	The channel gain between $u_s^m$ and $s$
$R_{u_s^m}^r$	Uplink data rate from $u_s^m$ to $s$
$CT_{u_s^m}$	Computational task of $u_s^m$
$D_{u_s^m}$	Input data of computation task $CT_{u_s^m}$
$C_{u_s^m}$	Workloads of computation task $CT_{u_s^m}$
$f_{u_s^m}^{\text{loc}}$	Local computing capability of $u_s^m$
$T_{u_s^m}^{\text{loc}}$	Local execution time of task $CT_{u_s^m}$
$T_{u_s^m, \text{off}}^r$	Transmission time of task $CT_{u_s^m}$ to the MEC server
$T_{u_s^m, \text{exe}}^r$	Execution time of task $CT_{u_s^m}$ at the MEC server
$E_{u_s^m}^{\text{loc}}$	Energy consumption of $u_s^m$ when executing its task locally
$E_{u_s^m, \text{off}}^r$	Energy consumption of $u_s^m$ when offloading its task $CT_{u_s^m}$
$f_{u_s^m}^r$	Computing resources that the MEC server allocates to $u_s^m$
$f$	Computing resources of the MEC server
$\beta_{u_s^m}^t$	Preference of $u_s^m$ on task completion time
$\beta_{u_s^m}^e$	Preference of $u_s^m$ on task energy consumption
$W_{u_s^m}$	User utility of $u_s^m$
$\mathcal{O}_s$	The set of offloading UEs under each SBS
$R_{u_s^m}^{\text{min}}$	The minimum rate requirement of $u_s^m$
$R_{1 \times \mathcal{O}_s}$	User benefit matrix
$R_{R1 \times N}$	Channel benefit matrix

**3.1. Network Model.** As shown in Figure 1, a two-layer cellular heterogeneous network composed of a macro cell base station (MBS) and  $S$  small cell base stations is considered [19, 20]. The MEC server is deployed on the side of the MBS and can perform multiple computing tasks at the same time.  $S$  SBSs are connected to the MEC server through optical fiber links like the MBS. Let  $\mathcal{S} = \{1, 2, \dots, s, \dots, S\}$  be the

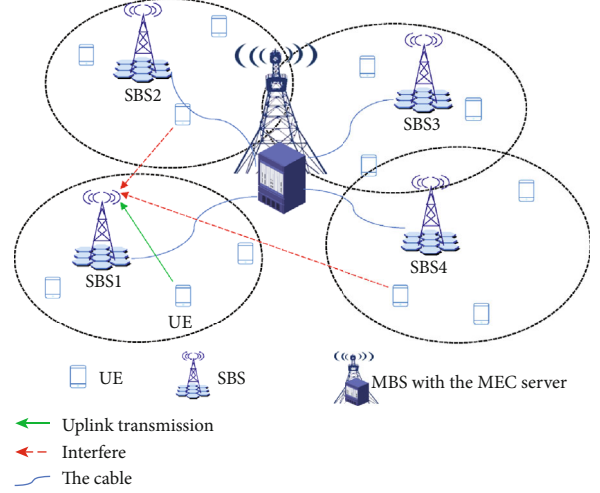


FIGURE 1: Cellular heterogeneous network model in mobile edge computing.

set of SBSs, and there are  $M$  UEs associated with each SBS in its coverage. We denote the set of UEs in the coverage area of  $s$  as  $\mathcal{U}_s = \{u_s^1, u_s^2, \dots, u_s^m, \dots, u_s^M\}$ , where  $u_s^m$  represents a UE belonging to  $s$ . In addition, for simplicity, the mobility of users or the handover among cells was not considered as it was assumed in [23–25]. Similar to many previous works in cloud computing and mobile networks [26–28], it is a semistatic scenario, which means that the position and transmission channel conditions remain unchanged during offloading a task.

**3.2. Communication Model.** It is assumed that each UE has a time-sensitive task that requires a lot of computing resources to complete. Each UE can perform by offloading the computing task to the MEC server through its associated SBS or execute the computing task locally. Therefore, we denote the offloading decision as  $x_{u_s^m} \in \{0, 1\}$ .  $x_{u_s^m} = 0$  means that  $u_s^m$  performs its task locally.  $x_{u_s^m} = 1$  means that the user of  $u_s^m$  chooses to offload the task to the MEC server via a wireless link. The task offloading decision can be expressed as  $\mathcal{X} = [x_{u_s^m}]$ , which is a matrix of  $S \times M$ .

Uplink spectrum multiplexing is used in this model. The spectrum resources of the entire system are divided into  $N$  orthogonal RBs, and the RB set is defined as  $\mathcal{N} = \{1, 2, \dots, n, \dots, N\}$ . The RB associated table is defined as  $Y_s = \{y_{u_s^m}^n\}$ , which is a  $M \times N$  matrix, where  $M$  is the total number of UEs in the  $s$ -th cell and  $N$  is the total number of RBs.  $y_{u_s^m}^n = 1$  means that the  $n$ -th RB is assigned to  $u_s^m$ ; otherwise,  $y_{u_s^m}^n = 0$ . And the RB allocation strategy is defined as  $\mathcal{Y} = \{Y_s\}$ ,  $s \in \mathcal{S}$ .

During uplink transmission, each UE and each SBS have a single antenna for sending and receiving messages. When  $u_s^m$  offloads its task to the MEC server for calculation, interference will occur if there are UEs in other SBSs sharing the same RB(s) with the current  $u_s^m$ . As RBs are assigned orthogonally to users in each cell, there is no interference

in intracell. The interference transmission power from  $u_t^m$  sharing the  $n$ -th RB to the  $s$ -th cell can be described as

$$I_{u_s^m}^n = \sum_{t=1, t \neq s}^S \sum_{m=1}^M x_{u_t^m} y_{u_s^m}^n \frac{P_{u_t^m}}{K_{u_t^m}} H_{u_t^m, s}, \quad (1)$$

where  $P_{u_t^m}$  represents the transmission power of  $u_t^m$ ,  $K_{u_t^m}$  stands for the number of RBs assigned to  $u_t^m$ , and  $H_{u_t^m, s}$  denotes the channel gain between  $u_t^m$  and  $s$ .

Given the decision matrix  $\mathcal{X}$  and the RB associated strategy  $\mathcal{Y}$ , the uploading rate achieved by  $u_s^m$  connected to  $s$  can be obtained by Shannon's formula as [19]

$$R_{u_s^m}^r(\mathcal{X}, \mathcal{Y}) = x_{u_s^m} \sum_{n=1}^N y_{u_s^m}^n B \log_2 \left( 1 + \frac{P_{u_s^m} H_{u_s^m, s}}{K_{u_s^m} (I_{u_s^m}^n + \sigma^2)} \right), \quad (2)$$

where  $\sigma^2$  is the variance of background noise,  $B$  is the bandwidth of each RB,  $P_{u_s^m}$  represents the transmission power of  $u_s^m$ ,  $K_{u_s^m}$  stands for the number of RB allocated to  $u_s^m$ , and  $H_{u_s^m, s}$  denotes the channel gain between  $u_s^m$  and  $s$ .

**3.3. Calculation Model.** The computing task of  $u_s^m$  is described as  $\text{CT}_{u_s^m} = \langle D_{u_s^m}, C_{u_s^m} \rangle$ , in which  $D_{u_s^m}$  (in kB) represents the size of transmission data and  $C_{u_s^m}$  (in megacycles) specifies the workload, i.e., the number of CPU cycles required to complete the computing task. The values of  $D_{u_s^m}$  and  $C_{u_s^m}$  can be obtained by carefully analyzing the offloading task [29, 30]. The delay and power consumption of local and remote computation will be discussed, respectively.

- (1) **Local computing:** let  $f_{u_s^m}^{\text{loc}} > 0$  represent the local computing capacity of  $u_s^m$  in terms of the number of CPU cycles/s. The computation time  $T_{u_s^m}^{\text{loc}}$  for the local execution of the task  $\text{CT}_{u_s^m}$  can be expressed as

$$T_{u_s^m}^{\text{loc}} = \frac{C_{u_s^m}}{f_{u_s^m}^{\text{loc}}}, \quad (3)$$

and the energy consumption  $E_{u_s^m}^{\text{loc}}$  is denoted as

$$E_{u_s^m}^{\text{loc}} = k \left( f_{u_s^m}^{\text{loc}} \right)^2 C_{u_s^m}, \quad (4)$$

where  $k \left( f_{u_s^m}^{\text{loc}} \right)^2$  is the energy consumption per calculation cycle and  $k$  depends on the energy coefficient on the chip architecture. According to the actual measurement,  $k = 10^{-27}$  is usually adopted [21].

- (2) **Remote computing:**  $u_s^m$  is connected to the corresponding  $s$  through a wireless network, and its task is offloaded to the MEC server for calculation. The

computing resources provided by the MEC server are quantified by the computing capacity  $f$  (CPU cycles/s), which can be shared among the related UEs. The uplink transmission delay of  $u_s^m$  can be expressed as follows:

$$T_{u_s^m, \text{off}}^r = \frac{D_{u_s^m}}{R_{u_s^m}^r(\mathcal{X}, \mathcal{Y})}. \quad (5)$$

When a computing task  $\text{CT}_{u_s^m}$  is offloaded to the MEC server, the MEC server allocates specific computing resources to process the task, which is represented by  $f_{u_s^m}^r$  (CPU cycles/s). And the computing resource allocation profile is defined as  $\mathcal{F} = \{f_{u_s^m}^r\}$ . During the execution of the task, it is assumed that the calculation speed assigned by the MEC server to each UE is fixed. The time of the MEC server executing the task is described as

$$T_{u_s^m, \text{exe}}^r = \frac{C_{u_s^m}}{f_{u_s^m}^r}. \quad (6)$$

In addition, a feasible computing allocation strategy must satisfy the constraints of computing resources, which can be expressed as

$$\sum_{s \in \mathcal{S}} \sum_{u_s^m \in \mathcal{U}_s} x_{u_s^m} f_{u_s^m}^r \leq f. \quad (7)$$

The total delay of  $u_s^m$  for finishing the task is given by the following equation:

$$T_{u_s^m}^r = T_{u_s^m, \text{exe}}^r + T_{u_s^m, \text{off}}^r = \frac{C_{u_s^m}}{f_{u_s^m}^r} + \frac{D_{u_s^m}}{R_{u_s^m}^r(\mathcal{X}, \mathcal{Y})}. \quad (8)$$

Through the above analysis, the energy consumption of  $u_s^m$  during the transmission data can be calculated as

$$E_{u_s^m, \text{off}}^r = P_{u_s^m} \times T_{u_s^m, \text{off}}^r = \frac{P_{u_s^m} D_{u_s^m}}{R_{u_s^m}^r(\mathcal{X}, \mathcal{Y})}, \quad (9)$$

where  $P_{u_s^m}$  represents the transmitting power of  $u_s^m$ .

We mainly consider the energy consumption and delay of UEs, and the computing energy consumption of the MEC server is omitted. As the amount of data returned to the mobile users is small, the power consumption and latency of UE receiving the returned data are omitted.

## 4. Problem Formulation

In this section, the problem of task offloading, wireless RBs, and computing resource allocation is formulated under the definition of user and system utility.

In a mobile cloud computing system, UEs' preference is mainly manifested in task completion time of  $\beta_{u_s^m}^t$  and energy consumption of  $\beta_{u_s^m}^e$ .  $\beta_{u_s^m}^t, \beta_{u_s^m}^e \in [0, 1]$ , and  $\beta_{u_s^m}^t + \beta_{u_s^m}^e$

= 1. The quality of experience (QoE) can be described by comparing the delay and the power consumption of remote computing with that of local execution [18, 21]. The user utility of  $W_{u_s^m}$  for  $u_s^m$  can be defined as

$$W_{u_s^m} = \left( \beta_{u_s^m}^t \frac{T_{u_s^m}^{\text{loc}} - T_{u_s^m}^r}{T_{u_s^m}^{\text{loc}}} + \beta_{u_s^m}^e \frac{E_{u_s^m}^{\text{loc}} - E_{u_s^m}^r}{E_{u_s^m}^{\text{loc}}} \right) x_{u_s^m}. \quad (10)$$

$\beta_{u_s^m}^t$  and  $\beta_{u_s^m}^e$  can be determined according to the life of the remaining battery and the mission completion time requirements. From the above expression, it is clear that its user utility  $W_{u_s^m}$  is equal to 0 when the task of  $u_s^m$  is executed locally ( $x_{u_s^m} = 0$ ). When the task of  $u_s^m$  is executed on the MEC server ( $x_{u_s^m} = 1$ ), its user utility  $W_{u_s^m}$  is larger than 0.

Given the offloading policy of  $\mathcal{X}$ , the RB allocation strategy of  $\mathcal{Y}$ , and the calculating resource allocation policy of  $\mathcal{F}$ , the system utility can be defined as the sum of all user utilities and is expressed as follows:

$$W(\mathcal{X}, \mathcal{Y}, \mathcal{F}) = \sum_{s \in \mathcal{S}} \sum_{u_s^m \in \mathcal{U}_s} W_{u_s^m}. \quad (11)$$

To maximize the system utility by jointly optimizing task offloading, wireless RBs, and computing resource allocation in mobile edge computing, the optimal goal can be formulated as

$$\begin{aligned} & \max_{\mathcal{X}, \mathcal{Y}, \mathcal{F}} W(\mathcal{X}, \mathcal{Y}, \mathcal{F}) \\ & \text{s.t. C1: } x_{u_s^m} \in \{0, 1\} \forall u_s^m \in \mathcal{U}_s, s \in \mathcal{S} \\ & \text{C2: } y_{u_s^m}^n \in \{0, 1\} \forall u_s^m \in \mathcal{U}_s, n \in \mathcal{N}, s \in \mathcal{S}, \\ & \text{C3: } \sum_{s \in \mathcal{S}} \sum_{u_s^m \in \mathcal{U}_s} x_{u_s^m} f_{u_s^m}^r \leq f. \end{aligned} \quad (12)$$

The constraints in the above formula can be interpreted as follows: constraint C1 in (12) implies that the task can be executed locally or offloaded to the MEC server for execution. Constraint C2 in (12) indicates whether the  $n$ -th RB is assigned to  $u_s^m$ . Constraint C3 in (12) ensures that the sum of computing resources allocated to all offloading UEs does not exceed the computing capacity of the MEC server.

Due to the existence of integer variables, the above equation is a mixed integer nonlinear program (MINLP) problem [31]. The equation of (12) can be rewritten as follows:

$$\max_{\mathcal{X}, \mathcal{Y}, \mathcal{F}} W(\mathcal{X}, \mathcal{Y}, \mathcal{F}) = \max_{\mathcal{X}} \left( \max_{\mathcal{Y}, \mathcal{F}} W(\mathcal{X}, \mathcal{Y}, \mathcal{F}) \right). \quad (13)$$

From (13), it can be seen that offloading decision, RB allocation, and computing resource allocation are decoupled from each other [32].

The original problem can be translated into offloading decision and resource allocations. In the next section, we will present solutions to both the resource allocations and task offloading decision.

## 5. Resource Optimization and Task Offloading Strategy

In this section, considering the time delay and energy consumption demand of UEs, a resource allocation algorithm based on improved graph coloring is used to allocate RBs. The solution of computing resources is obtained by using KKT conditions, and a semi-distributed TOWCRM algorithm is adopted to optimize the offloading decision.

The set of offloading UEs for the  $s$ -th SBS is defined as  $\mathcal{O}_s$ .

If a feasible task offloading decision is given, the objective function of (12) can be translated as follows:

$$\begin{aligned} \max_{\mathcal{X}, \mathcal{F}} W(\mathcal{X}, \mathcal{Y}, \mathcal{F}) &= \max_{\mathcal{Y}, \mathcal{F}} \left( \sum_{s \in \mathcal{S}} \sum_{u_s^m \in \mathcal{U}_s} (\beta_{u_s^m}^t \beta_{u_s^m}^e) - V(\mathcal{X}, \mathcal{Y}, \mathcal{F}) \right), \\ \text{s.t. C1: } & y_{u_s^m}^n \in \{0, 1\} \forall u_s^m \in \mathcal{U}_s, n \in \mathcal{N}, s \in \mathcal{S}, \\ & \text{C2: } \sum_{s \in \mathcal{S}} \sum_{u_s^m \in \mathcal{U}_s} x_{u_s^m} f_{u_s^m}^r \leq f, \end{aligned} \quad (14)$$

where

$$V(\mathcal{X}, \mathcal{Y}, \mathcal{F}) = \sum_{s \in \mathcal{S}} \sum_{u_s^m \in \mathcal{O}_s} \left( \frac{\beta_{u_s^m}^t T_{u_s^m}^r}{T_{u_s^m}^{\text{loc}}} + \frac{\beta_{u_s^m}^e E_{u_s^m}^r}{E_{u_s^m}^{\text{loc}}} \right). \quad (15)$$

From (14), it is easy to see that  $\sum_{s \in \mathcal{S}} \sum_{u_s^m \in \mathcal{U}_s} (\beta_{u_s^m}^t + \beta_{u_s^m}^e)$  is an exact value for a specific offloading decision of  $\mathcal{X}$ . The  $V(\mathcal{X}, \mathcal{Y}, \mathcal{F})$  can be regarded as the total offloading cost of all UEs who need to be offloaded. Therefore, the equation of (14) can be equivalent to minimize the total offloading overheads.

$$\begin{aligned} \min_{\mathcal{Y}, \mathcal{F}} V(\mathcal{X}, \mathcal{Y}, \mathcal{F}) &= \min_{\mathcal{Y}, \mathcal{F}} \left( \sum_{s \in \mathcal{S}} \sum_{u_s^m \in \mathcal{O}_s} \frac{\phi_{u_s^m} + \psi_{u_s^m}}{R_{u_s^m}(\mathcal{X}, \mathcal{Y})} + \sum_{s \in \mathcal{S}} \sum_{u_s^m \in \mathcal{O}_s} \frac{\eta_{u_s^m}}{f_{u_s^m}^r} \right) \\ \text{s.t. C1: } & y_{u_s^m}^n \in \{0, 1\} \forall u_s^m \in \mathcal{U}_s, n \in \mathcal{N}, s \in \mathcal{S}, \\ & \text{C2: } \sum_{s \in \mathcal{S}} \sum_{u_s^m \in \mathcal{O}_s} f_{u_s^m}^r \leq f, \end{aligned} \quad (16)$$

where  $\phi_{u_s^m} = \beta_{u_s^m}^t D_{u_s^m} / T_{u_s^m}^{\text{loc}}$ ,  $\psi_{u_s^m} = \beta_{u_s^m}^e D_{u_s^m} P_{u_s^m} / E_{u_s^m}^{\text{loc}}$ , and  $\eta_{u_s^m} = \beta_{u_s^m}^t f_{u_s^m}^{\text{loc}}$ .

It can be seen from (16) that RB allocation and computing resource allocation are decoupled from each other in the target and constraint. We can decouple problem (16) into two independent problems, namely, resource block allocation (RBA) and computing resource allocation (CRA), and their respective solutions are presented in the following sections.

**5.1. Resource Block Allocation (RBA).** Taking the first term in (16) as the objective function, the RB assignment problem of  $\Gamma(\mathcal{X}, \mathcal{Y})$  can be written as

$$\min_{\mathcal{Y}} \Gamma(\mathcal{X}, \mathcal{Y}) = \min_{\mathcal{Y}} \sum_{s \in \mathcal{S}} \sum_{u_s^m \in \mathcal{O}_s} \frac{\phi_{u_s^m} + \psi_{u_s^m}}{R_{u_s^m}^r(\mathcal{X}, \mathcal{Y})} \quad (17)$$

$$\text{s.t. } \gamma_{u_s^m}^n \in \{0, 1\} \forall u_s^m \in \mathcal{U}_s, n \in \mathcal{N}, s \in \mathcal{S}.$$

Note that in the RB allocation phase, it is assumed that all UEs are transmitted with a fixed transmission power of  $P_{u_s^m}$ . The transmitted power of each UE is equally distributed over each RB assigned to it. From (17), the minimal value of  $\Gamma(\mathcal{X}, \mathcal{Y})$  is obtained if the transmission rate of each offloading UE is maximized.

In order to better illustrate the transmission quality, the minimum transmission rate (when all UEs of the system are offloaded, computing resources are equally distributed to all UEs) is expressed as

$$R_{u_s^m}^{\min} = \frac{D_{u_s^m} \times f}{T_{u_s^m}^{\text{loc}} \times f - C_{u_s^m} \times S \times M}. \quad (18)$$

For the above RBA problem, it can be equivalent to the matching problem of the number of offloading UEs and the number of RBs. Therefore, a resource allocation algorithm based on an improved graph coloring method [19] is proposed to solve the above problem. The algorithm flow is simply described as follows:

- (1) Initialization (step 1): in this step, the MEC server sets the RB allocation strategy  $\mathcal{Y}$  to zeros and constructs  $S$  user benefit matrices as  $R_{1 \times \mathcal{O}_s} = \{r_{u_s^m}\}$ , where every user rate is  $r_{u_s^m} = B \log_2(1 + (P_{u_s^m} H_{u_s^m, s} / \sigma^2))$ ,  $u_s^m \in \mathcal{O}_s, s \in \mathcal{S}$ . At the same time, each UE also constructs its own channel benefit matrix  $R_{1 \times N} = \{\gamma_{u_s^m}^n B \log_2(1 + (P_{u_s^m} H_{u_s^m, s} / K_{u_s^m} (I_{u_s^m}^n + \sigma^2)))\}$ ,  $u_s^m \in \mathcal{O}_s, s \in \mathcal{S}$
- (2) Orthogonal allocation (step 2): if the number of the offloading UEs is less than or equal to the number of RBs, RBs are allocated according to a uniform zero frequency reuse (UZFR) method [9]. Otherwise, the elements in  $S$  user benefit matrices are sorted by  $r_{u_s^m}$  in descending order, and RBs are assigned to the first  $N$  UEs according to the UZFR method
- (3) Allocate the RB with the greatest channel benefit (step3): on the basis of step 2, the MEC server selects a UE with the best user benefit from the remaining UEs. The selected UE starts to update its own channel benefit matrix according to the RB allocation strategy at this time and then selects the RB with the greatest channel benefit as the transmission RB. The MEC server deletes the selected UE from the remaining UEs. Step 3 will be repeated until all offloading UEs are assigned to RB
- (4) Check whether all offloading UEs meet the minimum rate (step 4): according to (18), all offloading UEs will be checked whether they meet the minimum transmission rate. If satisfied, the algorithm

terminates. If not, these UEs need to continue to allocate RBs. The set of these UEs is denoted as  $I'$

- (5) RB reallocation (step 5): the MEC server selects a UE with the best user benefit from UEs that needed to continue to allocate RBs. The selected UE starts to update its own channel benefit matrix according to the RB allocation strategy at this time and then selects the RB with the greatest channel benefit as the transmission RB. The MEC server deletes the selected UE from UEs that needed to continue to allocate RBs. Repeat step 5 until all UEs that do not meet the minimum rate are once again assigned to RB
- (6) Iterative loop (step 6): step 4 to step 5 will be repeated until all UEs meet the minimum rate or  $I' = \emptyset$ . Then, the algorithm terminates and the RB allocation strategy  $\mathcal{Y}^*$  is obtained under the offloading decision

The optimal objective function of (17) can be calculated as

$$\min \Gamma(\mathcal{X}, \mathcal{Y}) = \min \sum_{s \in \mathcal{S}} \sum_{u_s^m \in \mathcal{O}_s} \frac{\phi_{u_s^m} + \psi_{u_s^m}}{R_{u_s^m}^r(\mathcal{X}, \mathcal{Y}^*)}. \quad (19)$$

5.2. *Computing Resource Allocation (CRA)*. From (16), computing resource allocation (CRA) is to optimize the second term of formula (16) and is expressed as follows:

$$\begin{aligned} & \min_{\mathcal{F}} \phi(\mathcal{X}, \mathcal{F}) \\ & \text{s.t. C1: } \sum_{s \in \mathcal{S}} \sum_{u_s^m \in \mathcal{O}_s} f_{u_s^m}^r \leq f, \\ & \text{C2: } f_{u_s^m}^r > 0, \end{aligned} \quad (20)$$

where

$$\Phi(\mathcal{X}, \mathcal{F}) = \sum_{s \in \mathcal{S}} \sum_{u_s^m \in \mathcal{O}_s} \frac{\eta_{u_s^m}}{f_{u_s^m}^r}. \quad (21)$$

From the above equation, it is a convex optimization problem. And constraint C2 in (20) is slack based on Karush-Kuhn-Tucker conditions, and it can be solved by using the KKT conditions.

The equivalent Lagrange function of this problem can be expressed as

$$L(f_{u_s^m}^r, \beta) = \sum_{s \in \mathcal{S}} \sum_{u_s^m \in \mathcal{O}_s} \frac{\eta_{u_s^m}}{f_{u_s^m}^r} + \beta \left( \sum_{s \in \mathcal{S}} \sum_{u_s^m \in \mathcal{O}_s} f_{u_s^m}^r - f \right). \quad (22)$$

Let  $\beta > 0$  be the Lagrange operator; the derivatives of the Lagrange function of  $L(f_{u_s^m}^r, \beta)$  can be described as

$$\frac{\partial L(f_{u_s^m}^r, \beta)}{\partial f_{u_s^m}^r} = -\frac{\eta_{u_s^m}}{(f_{u_s^m}^r)^2} + \beta. \quad (23)$$

By setting the above equation equal to 0, the solution of optimal computing resource allocation for problem (20) can be obtained.

$$(f_{u_s^m}^r)^* = \sqrt{\frac{\eta_{u_s^m}}{\beta}}, \quad (24)$$

where

$$\sum_{s \in \mathcal{S}} \sum_{u_s^m \in \vartheta_s} (f_{u_s^m}^r)^* = f. \quad (25)$$

By substituting (24) into (25) and setting  $f_{u_s^m}^r = 0$ , if  $u_s^m$  does not belong to  $\vartheta_s$ ,  $s \in \mathcal{S}$ , the solution of  $\beta$  can be obtained as

$$\beta^* = \left( \frac{1}{f} \sum_{s \in \mathcal{S}} \sum_{u_s^m \in \vartheta_s} \sqrt{\eta_{u_s^m}} \right)^2. \quad (26)$$

Substituting (26) into (24), the optimal solution can be obtained as follows:

$$(f_{u_s^m}^r)^* = \frac{f \sqrt{\eta_{u_s^m}}}{\sum_{s \in \mathcal{S}} \sum_{u_s^m \in \vartheta_s} \sqrt{\eta_{u_s^m}}}, u_s^m \in \vartheta_s, s \in \mathcal{S}. \quad (27)$$

The optimal objective function of (20) can be expressed as

$$\Phi(\mathcal{X}, \mathcal{F}^*) = \frac{\left( \sum_{s \in \mathcal{S}} \sum_{u_s^m \in \vartheta_s} \sqrt{\eta_{u_s^m}} \right)}{f}. \quad (28)$$

**5.3. Task Offloading Decision.** In the previous section, for a given task offloading decision  $\mathcal{X}$ , the solutions of RBA and CRA are obtained. According to (13), (16), (19), and (28), the system utility can be expressed as follows:

$$W^*(\mathcal{X}) = \sum_{s \in \mathcal{S}} \sum_{u_s^m \in \mathcal{U}_s} \left( \beta_{u_s^m}^t + \beta_{u_s^m}^e \right) - \Gamma(\mathcal{X}, \mathcal{Y}^*) - \Phi(\mathcal{X}, \mathcal{F}^*). \quad (29)$$

Given the RB allocation strategy of  $\mathcal{Y}^*$  and computing allocation strategy of  $\mathcal{F}^*$ , the objective function of (13) can be written as

$$\begin{aligned} & \max_{\mathcal{X}} W^*(\mathcal{X}) \\ & \text{s.t. } x_{u_s^m} \in \{0, 1\} \forall u_s^m \in \mathcal{U}_s, s \in \mathcal{S}. \end{aligned} \quad (30)$$

From the above equation, it is not a convex function due to the fact that  $\mathcal{X}$  is a binary variable. For the purpose of

solving this nonconvex problem, a semi-distributed TOWCRM algorithm consisting of two stages that can find a local optimum to problem (30) is adopted, as shown in Algorithm 1. In the first stage, each mobile user independently optimizes its user utility after optimizing wireless and computing resource allocation and determines whether to send an offloading request, including the information on mobile user parameters and the features of computation task. In the second stage, the MEC server determines whether the offloading user joins the offloading set by comparing the system utility, which includes the offloading user or not. Finally, the selected mobile users offload their computation tasks.

In stage 1, each UE calculates its own user utility  $W_{u_s^m}$ , according to  $\beta_{u_s^m}^t ((T_{u_s^m}^{\text{loc}} - T_{u_s^m}^r) / T_{u_s^m}^{\text{loc}}) + \beta_{u_s^m}^e ((E_{u_s^m}^{\text{loc}} - E_{u_s^m, \text{off}}^r) / E_{u_s^m}^{\text{loc}})$ . Moreover, each UE checks whether its user utility is larger than zero. If it satisfies, an offloading request is sent. Otherwise, an empty message is sent, which indicates that local computation is adopted.

In stage 2, the MEC server waits until it has collected all the requests and accepts the top  $N$  UEs of user utility in the offloading request. The initial offloading policy  $\mathcal{X}$  can be got. The corresponding RB allocation strategy of  $\mathcal{Y}^*$  and the corresponding computing resource allocation of  $\mathcal{F}^*$  are obtained, respectively. According to (29), the system utility of  $W(\mathcal{X}, \mathcal{Y}^*, \mathcal{F}^*)$  can be obtained. And let  $K$  be the set of UEs that the server accepts requests but does not accept offloading. The MEC server selects the UE with maximum user utility in  $K$  to add offloading policy  $\mathcal{X}$ , and the RB allocation strategy  $\mathcal{Y}$  and the computing resource allocation  $\mathcal{F}$  will be updated. According to (19), the system utility  $W(\mathcal{X}, \mathcal{Y}, \mathcal{F})$  can be obtained. If  $W(\mathcal{X}, \mathcal{Y}, \mathcal{F}) > W(\mathcal{X}, \mathcal{Y}^*, \mathcal{F}^*)$ , the MEC server removes this UE from the offloading policy. Otherwise, the system utility, RB allocation strategy, and computing resource allocation are updated. Finally, this UE is removed from the set  $K$ , and steps 21 to 33 will be repeated until the set  $K$  is equal to  $\emptyset$ . The MEC server forms the RB allocation strategy and computing resource allocation strategy and starts to send offloading decision to UEs. Receiving this message, UEs start to offload their tasks accordingly.

## 6. Simulation Results and Analysis

In a centralized MEC network, it consists of one MBS with the MEC server and four SBSs are deployed in  $100 * 100 \text{ m}^2$ . The MBS is located in the center of the area, and the four SBSs are placed in the four directions of the area. Each SBS has a coverage area of 30 m. The radio communication parameters follow the Third Generation Partnership Project specification [33]. It is assumed that the data size of  $D_{u_s^m}$  is 420 kB and the workload of  $C_{u_s^m}$  is 1000 megacycles. The MATLAB® package is used to carry out the simulations, and the system parameters are summarized in Table 2.

In addition, UEs are randomly distributed in the coverage of each SBS. If not particularly indicated, the number of RBs is 10. The system utility performance of the proposed

```

Stage 1: at UEs side
1: for each base station  $s \in \mathcal{S}$  do
2:   for each device  $u_s^m \in \mathcal{U}_s$  do
3:      $\mathcal{Y}^* \leftarrow$  improved graph coloring algorithm
4:      $\mathcal{F}^* \leftarrow$  equation (28)
5:      $W_{u_s^m} \leftarrow \beta_{u_s^m}^t (T_{u_s^m}^{loc} - T_{u_s^m}^r / T_{u_s^m}^{loc}) + \beta_{u_s^m}^e (E_{u_s^m}^{loc} - E_{u_s^m,off}^r / E_{u_s^m}^{loc})$ 
6:   end for
7: end for
8: if  $W_{u_s^m} > 0$  then
9:   send an offloading request
10: else
11:   send NULL
12: end if
Stage 2: at the MEC side
13: Wait until all requests are accepted
14: for  $(i = 0; i < N; i++)$  do
15:    $x_{u_s^m} \leftarrow \arg \max (W_{u_s^m}), u_s^m \in \mathcal{U}_s, s \in \mathcal{S}$ .
16:   set  $\mathcal{X} \leftarrow \mathcal{X} \cup \{x_{u_s^m}\}$ 
17: end for
18:  $\mathcal{Y}^* \leftarrow$  step 3;  $\mathcal{F}^* \leftarrow$  step 4
19:  $W(\mathcal{X}, \mathcal{Y}^*, \mathcal{F}^*) \leftarrow$  equation (29)
20: let  $K$  be the set of UEs that the server accepts requests but does not accept offloading
21: while  $|K| > 0$  do
22:    $x_k \leftarrow \arg \max (W_k), k \in K$ 
23:    $\mathcal{X} \leftarrow \mathcal{X} \cup \{x_k\}$ 
24:    $\mathcal{Y} \leftarrow$  step 3;  $\mathcal{F} \leftarrow$  step 4;  $W(\mathcal{X}, \mathcal{Y}, \mathcal{F}) \leftarrow$  step 19
25:   if  $W(\mathcal{X}, \mathcal{Y}, \mathcal{F}) < W(\mathcal{X}, \mathcal{Y}^*, \mathcal{F}^*)$  then
26:      $\mathcal{X} \leftarrow \mathcal{X} / \{x_k\}$ 
27:   else
28:      $W(\mathcal{X}, \mathcal{Y}^*, \mathcal{F}^*) = W(\mathcal{X}, \mathcal{Y}, \mathcal{F})$ 
29:      $\mathcal{Y}^* = \mathcal{Y}$ 
30:      $\mathcal{F}^* = \mathcal{F}$ 
31:   end if
32:    $K \leftarrow K / \{k\}$ 
33: end while
34: The MEC server forms RB allocation strategy  $\mathcal{Y}$  and computing resources  $\mathcal{F}$  and starts to send offloading decision  $\mathcal{X}$  to UEs

```

ALGORITHM 1: Semi-distributed TOWCRM Algorithm.

TABLE 2: Basic parameters of system simulation.

Parameters	Values
RB bandwidth $B$	1 MHz
Number of resource blocks RB	10
UE transmitted power $P_{u_s^m}$	20 dBm
UEs preference $\beta_{u_s^m}^t = \beta_{u_s^m}^e$	0.5
Input data size $D_{u_s^m}$	420 kB
Total number of CPU cycles $C_{u_s^m}$	1000 megacycles
UE computing capacity $f_{u_s^m}^{loc}$	0.7 GHz
MEC computing capacity $f$	100 GHz
The background noise $\sigma^2$	-100 dBm
Pathloss from UE to SBS	$140.7 + 36.7 \log_{10}(r)$

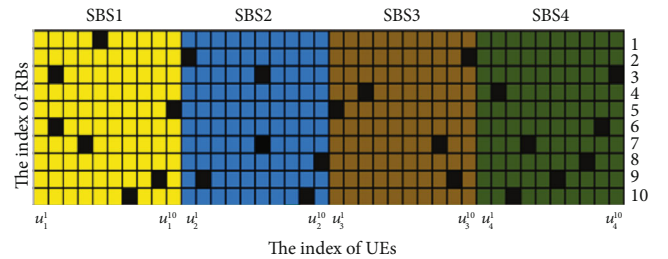
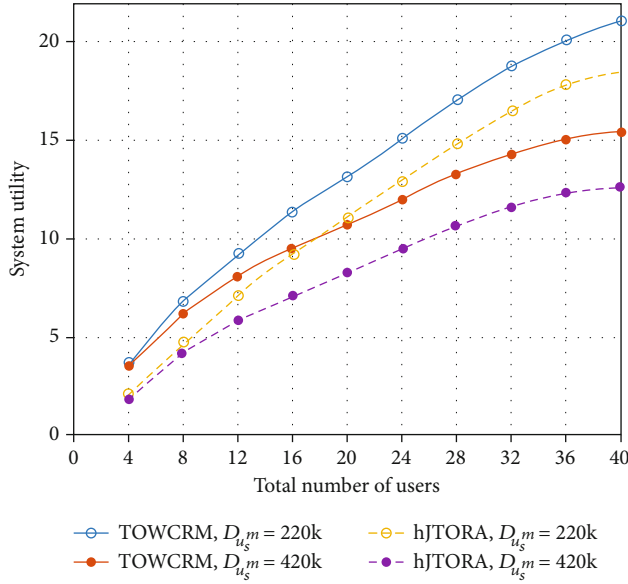


FIGURE 2: RB allocation based on improved graph coloring.

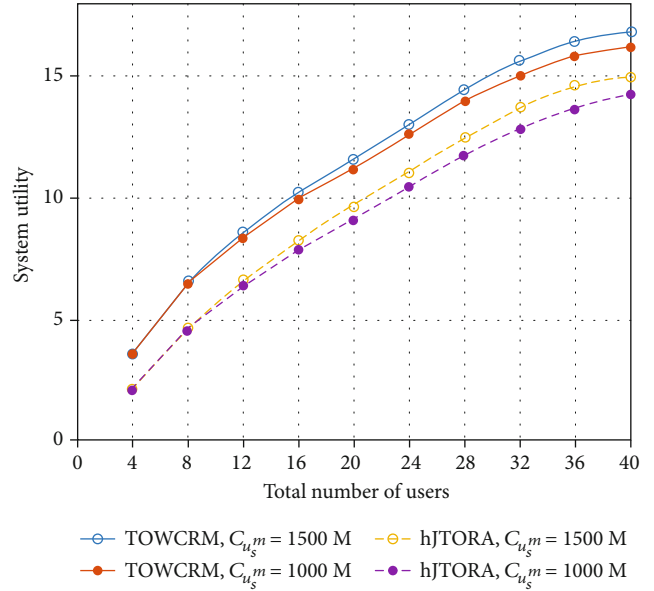
TOWCRM strategy is compared with that of the heuristic joint task offloading scheduling and resource allocation strategy (hJTORA) [21].

In order to visually show the resource allocation algorithm based on improved graph coloring, Figure 2 shows the RB allocation of UEs. There is a total of one MBS, four SBSs, and 10 RBs in the whole system, and there are 10 UEs associated with each SBS, among which there are 23





(a) The system utility with different input data sizes ( $D_{u_s^m}$ )



(b) The system utility with different workloads ( $C_{u_s^m}$ )

FIGURE 3: The system utility against different task input data sizes ( $D_{u_s^m}$ ) or workloads ( $C_{u_s^m}$ ).

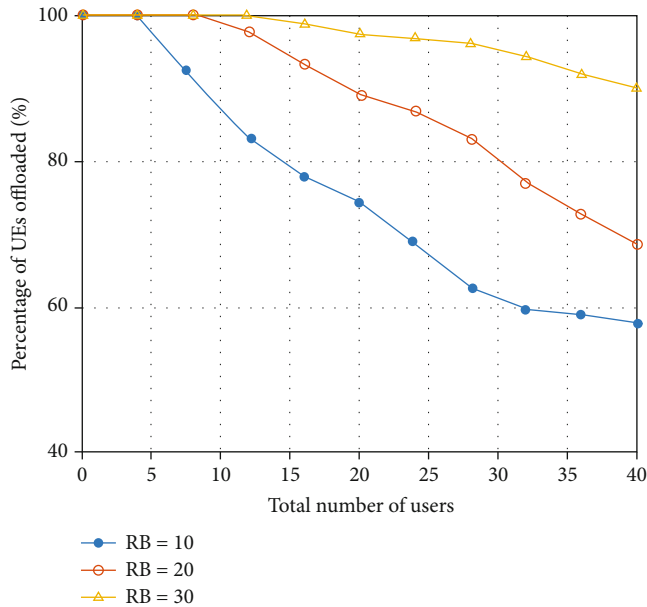


FIGURE 4: The relationship between the proportion of offloaded UEs and the number of UEs.

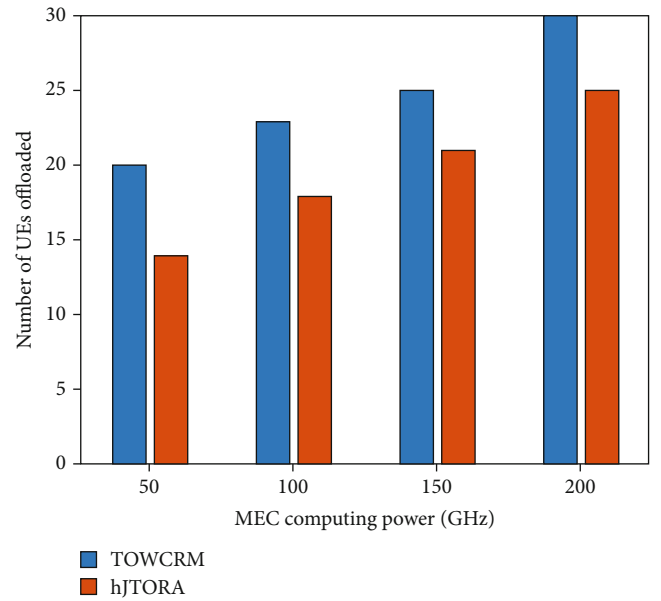


FIGURE 5: Comparison of the number of UEs offloaded against different MEC computing power.

offloading UEs. We can observe that the UE covered by the same SBS does not occupy the same RBs, and the RB is reused by the UEs belonging to different SBS, such as the 2-th and the 4-th RB. Some UEs are assigned to multiple RBs, such as  $u_1^6$  and  $u_2^6$ . The results of RB allocation show that the resource matching algorithm is effective.

By performing 1000 times of simulation, Figures 3(a) and 3(b) show the system utility performance with different  $D_{u_s^m}$  or  $C_{u_s^m}$ , respectively. From Figures 3(a) and 3(b), the system utility calculated by the proposed TOWCRM strategy is higher than that computed by hJTORA. From

Figure 3(a), it can be seen that the system performance of two strategies decreases as the tasks' input data size increases. From Figure 3(b), the system utility becomes larger as the tasks' workload increases. This means that the task with smaller input data or higher workloads will improve the value of system utility.

From Figure 4, the proportion of offloading users decreases, as the number of user increases. This is mainly because the capacity of computing resources and the RBs assigned to each offloaded users decreases, as the number of users increases. Therefore, more tasks tend to be

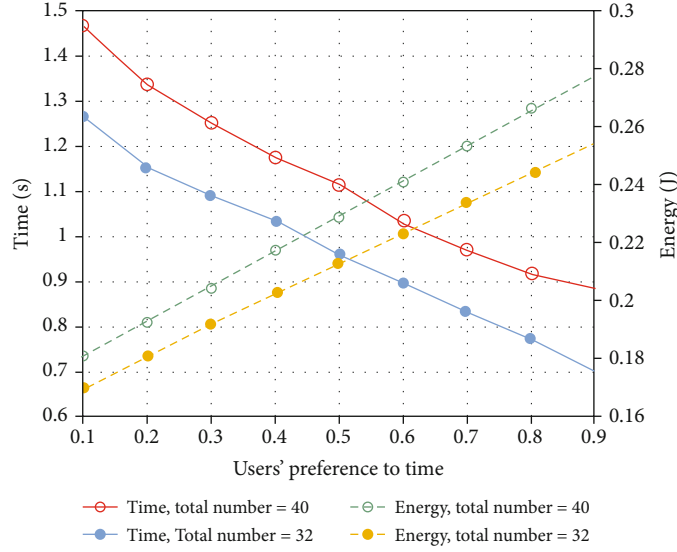


FIGURE 6: Time and energy consumption of all users obtained using TOWCRM.

processed locally. In addition, the proportion of offloaded users will increase with the larger number of RBs in the network.

The number of offloading UEs under different computing power is analyzed, as shown in Figure 5. It can be seen that the number of users offloaded by the TOWCRM and hJTORA algorithms is increasing with the enhancement of computing power. Because the computing power of MEC is stronger, the computation time of the offloading tasks becomes shorter. Therefore, more UEs will tend to offload their tasks to the MEC server to be processed. Moreover, under the same computing power of the MEC server, the number of UEs offloaded by the proposed algorithm is generally higher than that by using hJTORA.

Figure 6 shows the total time for finishing all offloading tasks and energy consumption obtained using TOWCRM when UEs' preferences to time of  $\beta_{u_s}^t$  vary from 0.1 to 0.9. It can be seen that the time is reduced, and the energy consumption is increased as  $\beta_{u_s}^t$  becomes larger. In addition, when the number of users in the system increases, the total time and energy consumption of users will be increased. This is because when more users participate in the competition for limited resources, a longer delay and higher energy consumption of all offloading UEs will occur.

## 7. Conclusion

In this article, the scenario of a multicell and multiuser mobile-edge computing network is modeled and analyzed. The optimization of the user utility and the system utility is formulated by combining task offloading and wireless and computing resource management. The original problem is decomposed into resource block allocation (RBA), computing resource allocation (CRA), and task offloading decision. The RBA is solved by using a resource allocation algorithm based on an improved graph coloring method. The optimal solution of CRA is obtained by using KKT con-

ditions. In task offloading, a semi-distributed TOWCRM strategy is proposed to optimize the system utility under the constraints of computing resources. Simulation results show the effectiveness of the scheme under different system parameters. The transmission power of every user is considered a fixed value and is equal to each other for wireless resource allocation in this work. The power control of each user will be studied to improve the system utility in the next research work.

## Data Availability

We derived the writing material from different journals as provided in the references. A MATLAB tool has been utilized to simulate our concept.

## Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

## Acknowledgments

This work was supported by the 2020 State Grid Corporation of China Science and Technology Program under Grant 5700-202041398A-0-0-00.

## References

- [1] S. Wang, M. Zafer, and K. K. Leung, "Online placement of multi-component applications in edge computing environments," *IEEE Access*, vol. 5, pp. 2514–2533, 2017.
- [2] S. Wang and S. Dey, "Adaptive mobile cloud computing to enable rich mobile multimedia applications," *IEEE Transactions on Multimedia*, vol. 15, no. 4, pp. 870–883, 2013.
- [3] J. Pan and J. McElhannon, "Future edge cloud and edge computing for Internet of things applications," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 439–449, 2018.

- [4] S. Bu and F. R. Yu, "Green cognitive mobile networks with small cells for multimedia communications in the smart grid environment," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 5, pp. 2115–2126, 2014.
- [5] R. Xie, F. R. Yu, H. Ji, and Y. Li, "Energy-efficient resource allocation for heterogeneous cognitive radio networks with femto-cells," *IEEE Transactions on Wireless Communications*, vol. 11, no. 11, pp. 3910–3920, 2012.
- [6] B. Yang, G. Mao, M. Ding, X. Ge, and X. Tao, "Dense small cell networks: from noise-limited to dense interference-limited," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4262–4277, 2018.
- [7] X. Ge, S. Tu, G. Mao, C. Wang, and T. Han, "5G ultra-dense cellular networks," *IEEE Wireless Communications*, vol. 23, no. 1, pp. 72–79, 2016.
- [8] G. Huang and J. Li, "Interference mitigation for femtocell networks via adaptive frequency reuse," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 2413–2423, 2016.
- [9] J. Zhang, W. Xia, F. Yan, and L. Shen, "Joint computation offloading and resource allocation optimization in heterogeneous networks with mobile edge computing," *IEEE Access*, vol. 6, pp. 19324–19337, 2018.
- [10] G. Yang, L. Hou, X. He, D. He, S. Chan, and M. Guizani, "Offloading time optimization via Markov decision process in mobile-edge computing," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2483–2493, 2021.
- [11] G. Peng, H. Wu, H. Wu, and K. Wolter, "Constrained multi-objective optimization for IoT-enabled computation offloading in collaborative edge and cloud computing," *IEEE Internet of Things Journal*, vol. 8, no. 17, pp. 13723–13736, 2021.
- [12] C. Yi, S. Huang, and J. Cai, "Joint resource allocation for device-to-device communication assisted fog computing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 3, pp. 1076–1091, 2021.
- [13] C. Guo, W. He, and G. Y. Li, "Optimal fairness-aware resource supply and demand management for mobile edge computing," *IEEE Wireless Communications Letters*, vol. 10, no. 3, pp. 678–682, 2021.
- [14] D. T. Nguyen, L. B. Le, and V. Bhargava, "Price-based resource allocation for edge computing: a market equilibrium approach," *IEEE Transactions on Cloud Computing*, vol. 9, no. 1, pp. 302–317, 2021.
- [15] Y. He, Y. Wang, C. Qiu, Q. Lin, J. Li, and Z. Ming, "Blockchain-based edge computing resource allocation in IoT: a deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2226–2237, 2021.
- [16] W. Feng, H. Liu, Y. Yao, D. Cao, and M. Zhao, "Latency-aware offloading for mobile edge computing networks," *IEEE Communications Letters*, vol. 25, no. 8, pp. 2673–2677, 2021.
- [17] G. Zhang, S. Zhang, W. Zhang, Z. Shen, and L. Wang, "Joint service caching, computation offloading and resource allocation in mobile edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 20, no. 8, pp. 5288–5300, 2021.
- [18] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3435–3447, 2017.
- [19] C. Wang, F. R. Yu, C. Liang, Q. Chen, and L. Tang, "Joint computation offloading and interference management in wireless cellular networks with mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 7432–7445, 2017.
- [20] H. Zhang, L. I. Hu, S. Chen, and H. E. Xiaofan, "Computing offloading and resource optimization in ultra dense networks with mobile edge computation," *Journal of Electronics & Information Technology*, vol. 41, no. 5, 2019.
- [21] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 856–868, 2019.
- [22] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: the communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.
- [23] A. Roy, S. K. Das, and A. Misra, "Exploiting information theory for adaptive mobility and resource management in future cellular networks," *Wireless Communications IEEE*, vol. 11, no. 4, pp. 59–65, 2004.
- [24] L. Ma, F. Yu, V. C. M. Leung, and T. Randhawa, "A new method to support UMTS/WLAN vertical handover using SCTP," *IEEE Wireless Communications*, vol. 11, no. 4, pp. 44–51, 2004.
- [25] F. Yu and V. Krishnamurthy, "Optimal joint session admission control in integrated WLAN and CDMA cellular networks with vertical handoff," *IEEE Transactions on Mobile Computing*, vol. 6, no. 1, pp. 126–139, 2007.
- [26] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2016.
- [27] D. Huang, P. Wang, and D. Niyato, "A dynamic offloading algorithm for mobile computing," *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 1991–1995, 2012.
- [28] J. Liu and Q. Zhang, "Code-partitioning offloading schemes in mobile edge computing for augmented reality," *IEEE Access*, vol. 7, pp. 11222–11236, 2019.
- [29] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 974–983, 2015.
- [30] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multiuser computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2016.
- [31] Y. Pochet and L. A. Wolsey, *Production Planning by Mixed Integer Programming*, Springer Science & Business Media, Berlin, Germany, 2006.
- [32] Y. Cheng, M. Pesavento, and A. Philipp, "Joint network optimization and downlink beamforming for CoMP transmissions using mixed integer conic programming," *IEEE Transactions on Signal Processing*, vol. 61, no. 16, pp. 3972–3987, 2013.
- [33] 3rd Generation Partnership Project, "Further advancements for E-UTRA physical layer aspects," Sophia Antipolis Cedex, France, 2010, 3GPP TR 36.814, E-UTRA Access, Tech. Rep..

## Research Article

# Data Integrity Time Optimization of a Blockchain IoT Smart Home Network Using Different Consensus and Hash Algorithms

**Ammar Riadh Kairaldeen** , **Nor Fadzilah Abdullah** , **Asma Abu-Samah** ,  
and **Rosdiadee Nordin** 

*Department of Electrical, Electronic and Systems Engineering, Faculty of Engineering & Built Environment, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia*

Correspondence should be addressed to Ammar Riadh Kairaldeen; aaltotanje@gmail.com

Received 27 June 2021; Accepted 5 October 2021; Published 9 November 2021

Academic Editor: Ruhui Ma

Copyright © 2021 Ammar Riadh Kairaldeen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Data security is a major issue for smart home networks. Yet, different existing tools and techniques have not been proven highly effective for home networks' data security. Blockchain is a promising technology because of the distributed computing infrastructure network that makes it difficult for hackers to intrude into the systems through the use of cryptographic signatures and smart contracts. In this paper, an architecture for smart home networks that could guarantee data integrity, robust security, and the ability to protect the validity of the blockchain transactions has been investigated. The system model is tested using various sizes of realistic datasets (30, 3 k, and 30 k to represent a small, medium, and large number of transactions, respectively). Four different consensus algorithms were considered, the conventional schemes concatenated hash transactions (CHT) and Merkle hash tree (MHT), as well as the newly proposed odd and even modified MHT (O&E MHT) and modified MHT (MMHT). Moreover, 15 hash functions were also examined and compared to understand the effects of each consensus algorithms on the data integrity verification check execution time and the time optimization provided by the proposed MMHT algorithm. The results show that even though the CHT algorithm gives the lowest execution time, it is impractical for a blockchain implementation due to the requirement to copy the entire blockchain ledger in real time. Meanwhile, the O&E MHT does not give any tangible benefit in the execution time. However, the proposed MMHT offers a minimum of 30% gain in time optimization than the conventional MHT algorithm typically used in blockchains. This work shows that the proposed MMHT consensus algorithm not only can identify malicious codes but has an improved data integrity check performance in smart homes, all while ensuring network stability.

## 1. Introduction

A large communication network of smart devices, sensors, and other consumer electronics such as a TV, refrigerator, and air conditioner in a home area network (HAN) has made communication and interaction among themselves very complicated and complex. Therefore, the communication between these devices in the network needs to ensure high security of data so that these systems' users can be provided with a high degree of privacy [1].

The changes brought by the Fourth Industrial Revolution have enabled the Internet of things (IoT) to play a significant role in bridging the gap between each of the phys-

ical industrial environments and cyberspace of computing systems. This requires multiple interconnected systems with unique identities that can communicate and interact with each another and transfer data over the network without requiring human-to-human or human-to-computer interaction, unlike the current case of physical industrial environments. In addition to this, the use of other advanced technologies such as artificial intelligence (AI), big data analytics (BDA), machine learning (ML), and other emerging tools helped in utilizing collected data effectively through different sources in the network. Therefore, through this practice, the processed data can be used to improve system efficiency and performance [2, 3]. To accomplish a highly

interactive, efficient but secure network, various elements and factors such as data privacy, authentication, ease of use and maintenance, and high security standards against possible attacks are needed. These robust and advanced features are possible using blockchain technology in the IoT system.

Various types of blockchain are used depending on the different elements and features of the system in consideration. One of the core features of blockchain-based IoT is authentication. The use of this feature helps strengthen the network against potential attacks from hackers. Also, the accessibility of data is another factor that defines the safety of data used by the system. Thus, different blockchain types are discussed further to examine their characteristics and effectiveness against data integrity threats. In general, there are two types of blockchain, namely, *permissionless* and *permissioned* [4].

A *permissionless* blockchain is a popular type of blockchain that allows anyone to participate in the network. This type of blockchain does not require participants to be authorized to be active in the network. Anyone can join the network and use their computational powers to contribute to the system. An example of a permissionless blockchain is a Bitcoin network that allows everyone to enter the system without any prior authorization. Therefore, participation is encouraged by granting entry into the network without the need for authorization. A participant's task is to ensure performance by verifying the network operation. These verifiers are important for the network as they enhance its operation. Hence, different algorithms are used by the verifiers.

The second type of blockchain is the *permissioned* blockchain, which requires the verifiers to gain authorization before taking part in the network. Verification nodes are set by a central authority, ensuring that the verifier should ask for permission before becoming involved in the blockchain. Permissioned blockchain can be further classified into *private* and *public* blockchains. Public blockchain allows anyone to read and submit the transactions, while private blockchain restricts the right to users of the organization to become involved in the network.

There are several fundamental differences between the permissionless and permissioned blockchains. These include their way of operation and the range of activities that they can perform [5]. Each of these blockchains has different benefits and limitations. One of the benefits of permissioned blockchains is scalability, as they allow the verification of all the transactions performed by the nodes. On the other hand, the permissionless blockchain provides the benefit of being highly resistant to the changes in the blocks using a single verifier. This capacity of the blockchain is highly beneficial in keeping data safe and secure. Therefore, this type of blockchain is very applicable to the HANs, making them safer for use. The large user base of HANs can find their data safe using permissionless blockchain that will prevent transactions without their consensus. Nevertheless, permissioned blockchains can also be used by the service providers who can act as a central authority to provide authorization to the users based on their requirements. Thus, through this practice, all the transactions can be monitored and controlled by the central authority. However, it has a disadvantage in that it can be very challenging to monitor and to control many transactions, which can make the networks less efficient.

Nowadays, the use of second-generation blockchains, including smart contracts, is on the rise in the industry [6, 7]. However, before the wide-scale adoption of smart contracts occurs, various factors must be considered in keeping the smart home applications private and secure. One of these factors includes the blockchain system that continuously monitors all activities from any intrusion [8]. Scalability can be very challenging as every node needs to process every transaction in the blockchain [9], resulting in a higher execution time cost required for the validator due to enormous computational power [10, 11]. Secondly, code correctness is an issue as both the developers and users must be confident about smart contracts and must not employ high fees for unnecessary computations [12]. In addition to this, the concern is about using a suitable authentication algorithm with a suitable hash function in the blockchain system [13].

In this work, the proof-of-work (PoW) is the selected algorithm to provide data integrity for smart homes [14]. The consensus algorithm is based on PoW that secures the network via the validator, which can be one or more participant nodes to verify the accomplished and submitted work, allowing them to add new transactions to the blockchain as it does not allow a single verifier to make changes to the entire network. The concept of consensus among most of the verifiers keeps the entire system safe from hackers' activity. Therefore, hackers must put in a large amount of time and money due to the massive computational efforts required. This effort has to be greater than all the power spent from the reference point that the hacker aims to change at a specific time [15]. The permissioned blockchains use different consensus protocols [16] to permit users to become authorized verifiers. Besides, this type of blockchain also uses a set of trusted parties to perform verification so that additional verifiers can become a part of the network. This can be achieved through the consensus of a current member and central authority of the blockchain. Financial settings are more likely to include this type of blockchain, which operates a Know Your Business (KYB) or Know Your Client (KYC) procedure to differentiate users that are allowed to undertake operations in a particular space [14].

This article has the following major contributions:

- (i) Design of a smart home architecture using a blockchain-based technology on specific criteria to ensure performance and security
- (ii) Selection of optimal consensus algorithms and authentication algorithms for smart homes, considering the security standards
- (iii) Comparison between different hash functions to select the most suitable for adoption into the authentication algorithm
- (iv) Propose a modified Merkle hash tree (MMHT) authentication algorithm to reduce the validation execution time

The rest of the paper is structured as follows. In Section 2, a brief background on IoT in a smart home area network,

data integrity in the blockchain system, highlighting standardizations used in security and policy, explanation on smart contract, consensus algorithms, authentication algorithms, hash functions, attacks, and smart home security attacks has been outlined. Section 3 presents the architecture components in the smart home ecosystem such as the server, hardware and software requirements, applications, smart contract structure, and design, illustrating the proposed design and architecture workflow. Section 4 describes the dataset used and the data preparation procedures. Section 5 explains the architecture of conventional and proposed consensus algorithms, the implementation steps of the proposed algorithms, and findings on the network's performance data integrity regarding transaction scalability and validation execution time. Finally, the paper is concluded in Section 6 with a detailed discussion on different issues involved in the proposed MMHT algorithm.

## 2. Related Works

*2.1. IoT in Smart Home Area Networks.* Over the architecture revolution, 5G networks promised to give credible schemes such as a high quality of service, ultralow latency, and high level of security demand [17]. Home area networks (HANs) are home-based networks that connect all the devices including computers, laptops, and smart appliances. This network is aimed at achieving energy-efficient smart homes by efficiently managing appliances and energy usage [18]. Therefore, the concept of smart homes relies on the application of HANs. HANs comprise various appliances integrated with the network and different sensing devices such as thermostats and smart meters. These sensors' primary objective is to collect data from these appliances and communicate it to the homeowners, utility providers, and other service providers. Therefore, this data flow is of key importance for HANs as it allows the homeowners and service providers to monitor and control the operation of the appliances and energy consumption. It is also important to note that most of this data communication occurs through different communication protocols such as Wi-Fi, RFID, and Zigbee [19]. However, most contemporary smart HANs are based on the Internet and Wi-Fi as they consume a large bandwidth of data transmission. Therefore, the high speed of Internet connections has made HANs very efficient and quick.

The integration of the smart devices and sensors with these networks has enabled the controllers to gain real-time information about various parameters including energy used and traffic load. Thus, the use of IoT in smart homes provides the stakeholders with a great opportunity to automate most appliances using smart systems. IoT is defined as the system of smart appliances and devices that can collect data and transfer it over the network without human interaction. Thus, the use of IoT in smart homes has reduced human interaction substantially as the appliances have become smart [20]. Smart appliances and devices are connected to the network in smart homes, which also comprise microcontrollers programmed to enable them to make decisions without human support. Thus, the entire IoT concept

in smart HANs is based on collections and effective data use. The ability of IoT-based HANs to collect, transfer, and process data has proved to be beneficial for homeowners and service providers as energy consumption has been made significantly efficient [21]. However, these systems have also increased the threat of people's data security and privacy.

Data security lies at the core of smart home networks as it is very important to keep the information being sent over the network safe [22]. For this reason, data security experts have been using different security protocols and standards to make smart home networks safer and strong against cyberattacks. Irrespective of these measures, smart HANs are still prone to the threat of data security. Various issues including credit card fraud, identity theft, and virus attacks are common for smart homes [23]. Therefore, the service providers have been using security methods such as encryption and authentication to protect people's privacy and avoid any data theft and information leak. Nevertheless, intruders have been able to decrypt the data and communication over the HANs, which requires data security experts to look for new methods to mitigate this issue. Challenges such as the complexity of the network structure in HANs and the devices' heterogeneity are major barriers to applying highly efficient and effective security standards [24]. IoT for data security has enabled researchers and experts to develop smart data security protocols, which enable high protection of the data and privacy of smart homeowners.

The use of IoT in smart HANs for data security is based on monitoring and control mechanisms. IoT's major role in smart home networks is the monitoring and data control through the application of security protocols that monitor data and prevent any suspicious activity.

Previously, blockchain is normally implemented in the public network [25]. However, with IoT there is a tendency for adoption of blockchains in the private network. Using Blockchain in the private network makes it easy to connect different systems horizontally rather than work on vertical compatibility; this will have the advantage of being easily scalable and adding more applications while considering security requirements.

IoT-based networks provide the benefit that they do not require human support in case any malicious activity is encountered [26]. These systems are smart enough to stop intruders from injecting the virus or malicious code into the network. Sensors are the major player in IoT that work to monitor the data once the network is live. The real-time monitoring of data through IoT on smart networks enables high security of all the network gateways and communication media. The data sent and received from all the sources is effectively monitored to prevent any data security attack. Besides, IoT also ensures data storage safety and manages the devices' operation status to enforce high-security standards. Through these practices, IoT has proved to be very efficient in maintaining data integrity and keeping it safe from cyber criminals [20]. Therefore, due to these security protocols offered by IoT, it is gaining high popularity among security experts.

The use of IoT in smart HANs also ensures the high availability of data and network systems. The efficient

monitoring and control processes employed by IoT have enabled the systems to reduce their downtimes as any encountered issues are handled automatically or communicated instantly to the human operators. Besides, IoT maintains a high focus on data integrity and confidentiality by using different security methods and data encryption protocols [27]. IoT-based data encryption is safer in smart home networks as the systems are continuously monitored. Thus, any activity of third-party intrusion into the network can be immediately detected. Hence, IoT in smart home networks effectively increases the data and network systems' security. The increase in IoT devices and their broad-spectrum applications in houses provides advanced ways to keep data safe. The risks from lack of transparency, auditability, and accountability in HANs are being catered using IoT through efficient application in critical areas by staying within the legal domains [28].

Attacks are one of the major threats to information systems and networks. They harm the integrity and security of data, leading to negative effects for various stakeholders of the systems [29]. Therefore, the systems' vulnerability to potential attacks must be managed so that the network systems can work with high data security standards. Different types of attacks exist to target the information systems and network, as explained by [30]. Some of these critical and most popular security attacks are examined as follows:

*2.1.1. Data Availability Attack.* This type of attack will be defeated by the data validity algorithm [31]. Different types of data availability attacks, the response of smart contract, and explanation of data validity check algorithm are as follows:

- (i) Malicious block attack: when a malicious block producer publishes a block to the blockchain, data validity will check the block inclusion to the blockchain and flag invalid transactions hash and show the fraud-proof status (attack status) to the system administrator
- (ii) Denial-of-service attacks: when a system is aware of data unavailability, it will flag an alarm without needing any kind of proof in the blockchain

*2.1.2. Access Control Impersonation Attack.* In HAN impersonation attack comprises both user and device impersonation, which is a form of fraud caused by replay, message modification, etc. [15]. The malicious attackers pose as a known or trusted person and gain admin privileges before using the smart home IoT ecosystem to share sensitive information or liability of any vicious activities. For instance, attackers abnormally manipulate IoT devices (home appliances) and increase Sauna's temperature, which is connected to a HAN, thus risking people's lives at home using the facility.

In HAN impersonation attack, it is necessary to follow the standardization protocol of communication to avoid the lack of service, using various data communication standards to eliminate impersonation attack over a HAN (e.g., ZigBee and Wi-Fi) which does not affect the blockchains efficiency [32]. The main challenge for both efficiency and speed of ensuring is that all nodes are not involved in poten-

tial impersonation attacks or fraudulent behavior. Validator nodes usually work to check consensus in the blockchain network. However, this work requires enormous computational power from those validators, and hence, in this research, we are trying to efficiently consensus algorithms that reduce computational power by decreasing the execution time and enhancing speed.

*2.1.3. Double-Spending Attack.* Double-spending attacks are one of the most popularly used threats by hackers in PoW algorithms. This type of attack occurs when the user controls more than 50% of the computing power [33]. Therefore, they can send a fraudulent transaction log to the network, enabling them to perform the same transaction multiple times by removing the record of previous transactions. However, this attack is not very easy to execute; but, it can be very harmful if hackers accomplish it.

*2.1.4. Side-Channel Attack.* The Merkle tree-based algorithm is also vulnerable to side-channel attacks, which reduces the integrity of data. A side-channel attack targets the authentication process, which reduces its reliability and effectiveness [5]. This type of attack introduces a malicious code that intrudes the authentication process, making it ineffective for testing the data's credibility. Therefore, this is one of Merkle tree-based networks' most critical threats as it takes away their ability to validate the datasets.

Attacks can be defeated by different parts of the system based on the type of attack and its effect. Table 1 summarizes the attacks considered in the implementation and design.

*2.2. Blockchain and Data Integrity.* Blockchain is one of the most advanced technologies for data security as it allows the data to be stored in blocks linked through the chain. This chain is complex enough to avoid the intrusion of cybercriminals. Blockchain is becoming popular in various parts of the world as it is a highly secure method of keeping data safe. The blocks are assigned hash values along with timestamps that depend on the data stored in each block and their link with the neighbor blocks [24]. Therefore, it is almost impossible for the hackers to intrude into the system and steal the data as it would require changing the hash value of a block, which would take high cost and time due to the dependency on other blocks. In addition, it is important to note that such an activity cannot go unnoticed as the network managers and cybersecurity experts have a close monitoring and control system over the blockchain [34]. Therefore, breaching the security of the blockchain is the most challenging task for any cybercriminals. Also, blockchain technology in data safety is based on a consensus algorithm, which prevents malicious activities from becoming successful. Hence, most organizations are shifting their database to blockchain to ensure high security and integrity of data.

Moreover, blockchain is based on cryptography that uses encryption through advanced algorithms to hide the real data [10]. The use of encryption in the blockchain is very important for data integrity as it keeps the data safe during communication, processing, and storage. Thus, the use of blockchain in IoT systems adds a major benefit to the latter

TABLE 1: Summary of attacks on HANs.

Attack name	Effect	Defended by
Malicious block (cite)	Attackers produce a malicious block	Authentication algorithm and smart contract
Denial-of-service (cite)	Data unavailability	Access authority and management
Access control impersonation (cite)	Fraudulent behavior	Consensus algorithm
Double-spending (cite)	Same transaction multiple times	Consensus algorithm
Side-channel (cite)	Authentication process	Authentication algorithm

as it substantially improves the data security standards. However, it is one of the riskiest areas in IoT due to a large volume of data [11]. IoT-based smart home networks are highly vulnerable to data security threats as they involve collecting, transferring, and storing household users' personal information. Blockchain can prevent both data tampering and spoofing [35], recording all node transactions in the blockchain, which is a complete managing and securing of the industrial IoT and operational technology (OT) devices, in which the transacted data of the sensor, device, or controller after it is deployed and starts working cannot be changed [36]. Another important benefit of using blockchain in IoT-based systems is that it will take intruders a large amount of time to break into the system and data.

The researchers conducted a study over blockchain for the security of data in smart home networks. It was noted that the increasing use of smart home networks is raising different challenges of privacy and authentication. Therefore, the authors have proposed an IoT-based blockchain for smart homes to ensure the safety and integrity of data. The authors proposed network architecture based on key blockchain elements such as smart contracts and tokens to perform strong verification checks. These verification checks are the primary function of the blockchain that enables them to perform authentication checks. Using these security protocols, smart home networks can be strongly secured by blockchain technology. The authors of [37] have used an IoT-based network to conduct tests on the use of blockchain to apply highly secure standards for the transaction of information and data. Thus, the result found the model to be highly effective in preventing any attack from external forces.

**2.3. Blockchain Standardizations.** Security standards are a set of policies and methods to keep the smart home system protected. Security standards allow the systems to become safer as the standards have been tried and tested before. Thus, the use of the developed standards is very effective in making secure networks and data systems. These standards are developed by internationally acclaimed organizations such as ISO, NIST, IEEE, ITU, and W3C, which work to introduce standardization at all levels and areas of the firms. Therefore, organizations and industries can keep themselves protected from the existing threats and challenges [38].

The use of network security standards is aimed at preventing, detecting, and rectifying network challenges and threats. The use of these network standards is critical in ensuring the security and integrity of data. Hence, in our research, we aim to comply with the standards as they are

the factors that ensure our research's security to be applicable [39]. There are several standards available in the market. Most of them are authorized by the governments and used in different private and public sectors.

Some of these standards focusing on blockchain and distributed ledger technologies are as follows:

- (i) Permitted distributed ledger (PDL) provides the foundations for the operation of permitted distributed ledgers, which is not limited to standardization activities. Furthermore, it includes research activities and initiatives concerning blockchain and the distributed ledger [40]
- (ii) Focus group on application of distributed ledger technologies (FG DLT) provides the process and technologies to synchronize the distributed ledger across the network's nodes to undertake the updates and validate the network's nodes securely

Other standards play an important role in facilitating business interaction, communication, measurement, and manufacturing [41]. For example, ISO 27001 is an information security management standard based on the assessment of organization management of its data security systems. The implementation and control of data security are the major requirements of this standard [10].

Thus, its use is significant for home networks as it can help the service providers to maintain standardized security systems that can be applied universally to all households. Similarly, in ISO 7498-2, there are seven layers in the security architecture:

- (1) The authentication layer
- (2) The access control layer
- (3) The nonrepudiation layer
- (4) The data integrity layer
- (5) The confidentiality layer
- (6) The assurance/availability layer
- (7) Notarization/signature layer

ISO 7498-2 standard is based on using a reference model to secure different basic layers of the information system model. This standard has different security protocols for each of the layers. ISO 7498-2 focuses on the communication used by the information systems and ensures that each layer communicates the data in a very secure manner.



Therefore, this standard provides highly efficient security of networks involving various clients of smart home networks. Hence, this standard is examined in detail in this research to ensure data security for IoT smart home networks based on blockchain technology.

*2.4. Smart Contract.* Smart contracts are digital contracts that are self-executable without the need of a third party. The use of smart contracts is very crucial in decentralized networks such as blockchain that do not have a central authority, allowing all the parties to interact with each other without any third party. Therefore, smart contracts ensure that all the transactions between the nodes are credible and reliable [42]. A smart contract is very efficient in avoiding conflicts and keeping the cost of transactions minimal. The safety of blockchain is highly dependent on smart contracts because they allow the users to share information, money, shares, etc., without any middleman [43]. The effectiveness of a blockchain is incomplete without smart contracts, as it is the most important tool to ensure that all the transactions are carried out fairly.

Smart contracts are a major part of the second-generation blockchains. Previously, the experts found that the blockchains are less effective due to the tools' inability to handle the conflicts [44]. Recently, the industry's application interest that focuses on digital assets has increasingly moved to the second generation of blockchain applications, including digitizing asset ownership, smart contract, and intellectual property. It must be noted that smart contracts act like real business contracts, which ensure that all the parties comply with the contract terms. Therefore, the increasing trend of blockchain applications worldwide is enabling organizations to use smart contracts. This practice has the benefit that smart contracts are strong enough to control the behavior of parties that are part of the blockchain. They are more efficient at keeping the transactions fair as compared to physical contracts. A smart contract is also advantageous because it can be encoded as computer code rules, which can then be replicated and executed across all the blockchain nodes. Therefore, this saves the time and cost of making an individual contract for all the blockchain nodes.

IoT devices generate a huge amount of sensitive data with limited resources. Using blockchain technology with a decentralized smart contract, the network will be more secured and efficient by improving different factors like error handling, monitoring, analysis, and data and identity issues. However, this is outside the scope of this paper, where our work focuses on the design and configuration of the smart contract as a part of the blockchain.

Moreover, another benefit of smart contracts is that they are self-enforcing. They do not require any external authority to manage the contractual terms and monitor external inputs from trusted sources, such as a financial exchange or meteorological service [14]. The complexity of the network due to a large number of users and the structure of the conventional contracts makes them very ineffective. Hence, smart contracts are very critical as it allows the blockchain users to ensure their safety. Smart contracts are

incumbent upon all the nodes to perform according to their responsibilities. Through this practice, blockchains have become very reliable for making transactions. Bitcoin and Ethereum are common blockchain examples that use a smart contract for keeping all the money transactions safe and fair [45]. Hence, through this practice, blockchains are becoming a regular part of various organizations.

Smart contracts can also be effectively used for smart home networks that use blockchain. In smart home networks, data fairness and safety management are one of the major challenges that can be overcome by using smart contracts in blockchains [46]. Also, in smart home networks, the users are preferably not to be managed manually. Therefore, smart contracts allow the entire network to be managed digitally without the need for any external or third-party authority. The smart contracts work based on verification as they ensure that the terms are fully satisfied by the users. This practice can accomplish high data safety and integrity for the data producers and consumers [47].

There are various types of smart contracts that are used in blockchain systems. The following are the fundamental types of smart contracts [48]:

- (1) Decentralized autonomous organizations (DAOs): this type is based on the general rules that are made by the developers who designed the blockchain. These rules apply to all the participants of the chain. Therefore, all the users have to comply with the rules to ensure that they can work effectively in the blockchain community. One of the key points about DAOs is that they are usually handled manually as the programmers and developers have a strong influence over the control of the smart contracts
- (2) Application logic contracts (ALCs): this type is usually engraved in the program's code. These contracts work with the program code and other smart contracts to enforce the rules for blockchain users. Therefore, they are completely independent of the external forces as they can make decisions based on the programmed code. ALCs are highly effective in handling the entire network transactions independently as they can be replicated at all the nodes automatically without human assistance. Hence, such smart contracts are gaining high popularity among network designers and developers

*2.5. Consensus Algorithms.* Consensus algorithms are the core part of the blockchain network as they allow the network to function without any central authority. Consensus algorithms are based on the mechanism of consensus, in which all peers need to arrive at a common agreement about the states of a ledger [49].

Therefore, this mechanism of the blockchain networks ends the need for any centralized power. Through this practice, the peers in the network can build trust and carry out secured transactions among themselves. Various types of consensus algorithms exist in the blockchain network, which is applied according to the network's objectives and the

users' needs. Different consensus algorithms are used in blockchain; they are generally categorized as proof-based and voting-based consensus algorithms. The classification is shown in Figure 1. The proof-based consensus algorithms require the nodes joining the verifying network to show that they are more qualified before having the right to append a new block to the chain. Meanwhile, voting-based consensus algorithms consider the number of votes cast by nodes on the network to achieve consensus on transactions and key network decisions.

The comparison of these consensus algorithms is shown in Table 2 [50]. The overview for each consensus algorithms is further explained as follows.

- (i) *Proof of work (PoW)*: the concept of this algorithm is to produce a new block to the blockchain and confirm the transaction. This process responsibility bears special nodes called miners, and a process is called mining. In PoW, miners compete against each other to complete transactions on the network and get rewarded [44]. This algorithm offered multi-signature transactions and multichannel payments over an address to enhance blockchain security. It also has strong support to increasing numbers of nodes in the network. This type of consensus algorithm is highly effective in an environment where there is a lack of trust among the nodes (e.g., public networks of home networks) that involves multiple data collectors and communicators, where each user sends each other digital tokens [51]
- (ii) *Proof-of-stake (PoS)*: this algorithm works on an incentive mechanism, where every block is validated through a betting system [53]. If a consensus is made between the majority of the blocks about adding another block, the stakes of all the blocks are raised, which is a strong support to increasing numbers of nodes in the network. Therefore, this algorithm has a drawback that it can cause major stakeholders' monopoly, influencing the transactions' efficiency. While PoS solved various issues earlier associated with PoW, two popular PoS variations are DPoS and LPoS
- (iii) *Delegated proof of stake (DPoS)*: this type of algorithm is based on a positive relationship; the more the coins and vote to invest, the more the weightage to receive, providing the semicentral control benefits to the network. For instance, to increase the speed while maintaining the features of the decentralization network and enhance the efficiency [59], this algorithm has strong support to increasing numbers of nodes in the network
- (iv) *Leased proof of stake (LPoS)*: this type of algorithm operates on the wave platform, which is considered as an advanced version of the traditional PoS. The users will add the next block to the blockchain by releasing a larger amount of a cryptocurrency to the full node; as a part of the process, the leaser will gain a percent of a transaction fee [54]. Also, this algorithm has strong support to increasing numbers of nodes in the network
- (v) *Proof of activity (PoA)*: this algorithm is a hybrid approach of PoW and PoS blockchain consensus models designed through the convergence of both of them. The validator races to solve cryptographic mathematical challenges at the earliest using special hardware and electric energy, similar to PoW. The mechanism could end up to Prove of PoS when the blocks added the network and hold only the information about block winner and reward the transaction identity. [30]. This algorithm has strong support to increasing numbers of nodes in the network
- (vi) *Proof of identity (PoI)*: the concept in this consensus algorithm is just the same as that of the authorized identity. To ensure the authenticity and integrity of the created data by network users using cryptographic confirmation of the private key attached in any transaction, any identified user in the network works under a consensus that the PoI algorithm can create and then manage a block of data that can be presented to others in the network [54]. This algorithm has strong support to increasing numbers of nodes in the network
- (vii) *Proof of importance (PoI)*: this algorithm developed NEM; PoI is a variation of the PoS consensus algorithm that considers the mechanisms of validators for its operation. However, this algorithm is used to determine which network nodes are eligible to manage, when adding a new block to the blockchain and which are not affected by the size and balance only but also other factors similar to the number of transactions between network nodes; reputation and overall balance also play a role in it [30]. In this algorithm, scalability is considered and support increasing numbers of nodes in the network
- (viii) *Proof-of-capacity (PoC)*: this algorithm is a less popular consensus algorithm [13], where the validators invest their hard disk space into the network, rather than adding any money or hardware. The more space a validator has in the network, the bigger his chance to mine the next block and get rewarded. Therefore, this system also promotes monopoly, making the smart home network less effective and unsafe. It relies on the computing power of the miners to their disk capacity [43], which is significantly energy efficient. The miners store huge data to mine the next block. The drawback of this technique is high latency, especially with the smart home network in which the devices have limited storage capacity. Hence, PoW is the most suitable consensus algorithm for the home network [60]. In this research, the PoW has been used in the system model. Scalability?

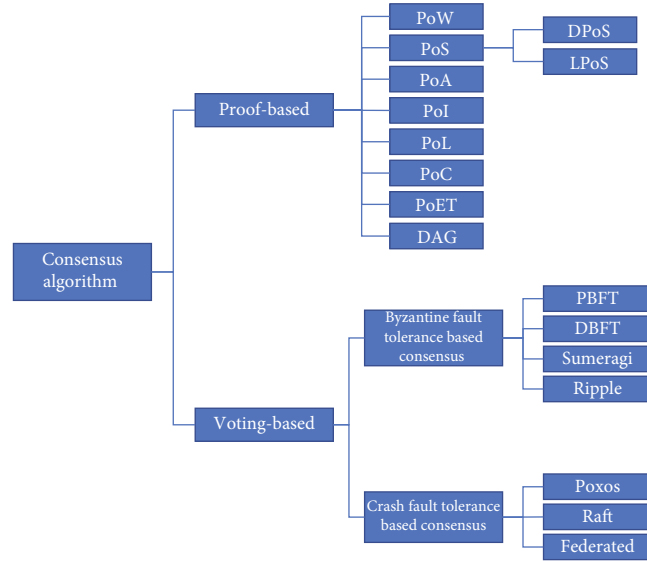


FIGURE 1: A summary of the consensus algorithms.

TABLE 2: Consensus algorithm comparison.

Algorithms	Designing goal	Advantages	Disadvantages	Scalability
PoW [44, 51]	Sybil-proof	(i) Security improvements (ii) Minimize the attacks up to 50% or less [52]	(i) More power consumption (ii) Centralized miners	Strong
PoS [53]	Energy efficiency	(i) Energy efficient (ii) More decentralized	(i) Nothing-at-stake problem	Strong
DPoS [53]	Organize PoS effectively	(i) Energy efficient (ii) Scalable (iii) Increased security	(i) Partially centralized (ii) Double spend attack	Strong
LPOS [54]	Distributed PoS	(i) Fair usage (ii) Lease coins	(i) Decentralization issue	Strong
PoA [29]	Benefits of both Pos and PoW	(i) Reduces the probability of the 51% attack (ii) Equal contribution	(i) Greater energy consumption (ii) Double signing	Strong
PoL [54]	Improve PoS	(i) Vesting (ii) Transaction partnership	(i) Decentralization issue	Strong
PoC [13]	Less energy than PoW	(i) Cheap (ii) Efficient (iii) Distributed	(i) Favoring bigger fishes (ii) Decentralization issue	Strong
PoET [55]	Decide the mining rights	(i) Cheap participation	(i) Need for specialized hardware (ii) Not good for public blockchain	Low
DAG [44, 56]	Speed and scalability	(i) Low-cost network (ii) Scalability	(i) Implementation gaps (ii) Not suited for smart contracts	Strong
BFT [30]	Failures of system	(i) Energy efficiency (ii) Transaction finality	(i) Number of replicas in the network (ii) Message complexity	Low
PBFT [30]	Remove software errors	(i) No need for confirmation (ii) Reduction in energy	(i) Communication gap (ii) Sybil attack	Low
DBFT [30]	Faster PBFT	(i) Scalable (ii) Fast	(i) Conflicts in the chain	Medium
Sumeragi [57]	Reputation system.	(i) Distributed across many clusters	(i) The more nodes that exist on the network, the more time it takes to reach consensus	Medium
Ripple [58]	Same FBFT	(i) Reduce the latency	(i) Few nodes required to vote, not really distributed network	Strong

- (ix) *Proof of elapsed time (PoET)*: this algorithm developed by Intel enhances the PoW mechanism in view that the CPU architecture and the validator hardware identify when and at what frequency a validator deserves the reward block. It is based on fair network distribution and expanding the odds for a bigger fraction of participant nodes in the blockchain. The task for every participating node is to wait for a particular time to participate in the following mining process. In this case, the miner is randomly chosen to solve the hash problem based on a random wait time [55]. Network validator nodes with the shortest hold-up time have the authority to offer a block. Simultaneously, every network node similarly comes up with its own waiting time. After the sleep mode, the node gets active and a block is available. This network node is considered as a validator. In the end, the validator can spread the information throughout the blockchain network, even though maintaining the property of decentralization in the network and then receiving the shared reward. This technique is aimed at reducing energy consumption
- (x) *Direct acyclic graph (DAG)*: this algorithm is familiar to every blockchain mobile app development service company. In this model, all nodes in the network can be a “miner” and validate transactions by themselves and reduce fees to zero, making the process easy, faster, and secure. This algorithm is used in Tangle [56] that reduces the computational time and does not use blocks to store transactions. With DAG, each transaction must approve two older transactions to provide a fast, scalable supports and no transaction fee framework for data integrity [44]. The drawback of this technique is the storage caused by imposing the rule to choose two-order transactions for approval. It can be solved by running a node named “coordinator” to perform this rule. However, this can be in conflict with the decentralization’s basics of the blockchain architecture
- (xi) *Byzantine fault tolerance (BFT)*: (or called Byzantine Generals Problem) this algorithm is used to deal with the Byzantine fault in situations where the system’s actors have to agree on an effective strategy to circumvent catastrophic failure of the system, but some of them are dubious. The two variations of BFT models that are prime in the blockchain arena are PBFT and DBFT [30]. This algorithm has low support to increasing numbers of nodes in the network
- (xii) *Practical Byzantine fault tolerance (PBFT)*: this algorithm is a lightweight algorithm that solves the Byzantine Generals Problems by enabling the user to confirm the messages delivered to them by performing a computation to evaluate the decision about the message’s validity. This algorithm

has low support to increasing numbers of nodes in the network [30]

- (xiii) *Delegated Byzantine fault tolerance (DBFT)*: (presented by NEO) this algorithm is similar to the DPoS mode, besides being more effective by countering unreliable or untrustworthy participants to the blockchain. This algorithm has medium support to increasing numbers of nodes in the network. Moreover, the NEO token holders can vote for the delegates [30]

According to [41], the consensus algorithm is used in the smart home network to prevent anybody from playing the network and to secure the communicated devise and stored data in this network against malicious acts that desire to wreak havoc with someone’s home. This work is focused on the PoW based on optimizing data integrity for the time required to secure the blockchain of the smart home network, taking into consideration the total storage required and executing several hash algorithms.

Moreover, the benefits gained from PoW, not covered in this work, are solving the cooperation tracking, collective behavior, and discrete opinion [61]. These are based on using statistical methods and mathematical calculation. Furthermore, this work is not attempting to change the structure of the PoW and disrupt the core characteristic of the blockchain, which should be always secure, decentralized, and peer to peer. The proposed system [62] works on improving the transaction speed and scalability by modifying the structure of PoW and introducing a parallel PoW in which the miners work together in such task to validate the transactions.

Additionally, it is suitable when working with the mathematical challenges in the digital ledger, such as recording and maintaining the unalterable transactions [63]. Typically, PoW is a computationally expensive consensus approach. However, such techniques reduce the computation requirements to achieve data integrity of the network without harming the data protection criteria [28]. Hence, the smart home networks’ high security can be ensured by the PoW consensus algorithm.

**2.6. Verification Algorithms.** The data structure verification algorithm is also known as the hash tree [64], which includes transaction blocks. Every node connected to IoT devices in a smart home network, including miners in a blockchain network, has a memory pool (Mempool) that contains all current transactions that are waiting to be added to the blockchain to produce a new block. This memory pool contains all the current transactions in wait, which must be added to the blockchain to create a new block.

The verification and summarization of all the transactions after hashing are performed by the different algorithms [34]. In consideration of the particularity and complexity of the chained hash database, the concatenated hash transaction (CHT) strong component is collision resistance but not more than collision resistance because it is feasible for an attacker to find two messages with the same hash

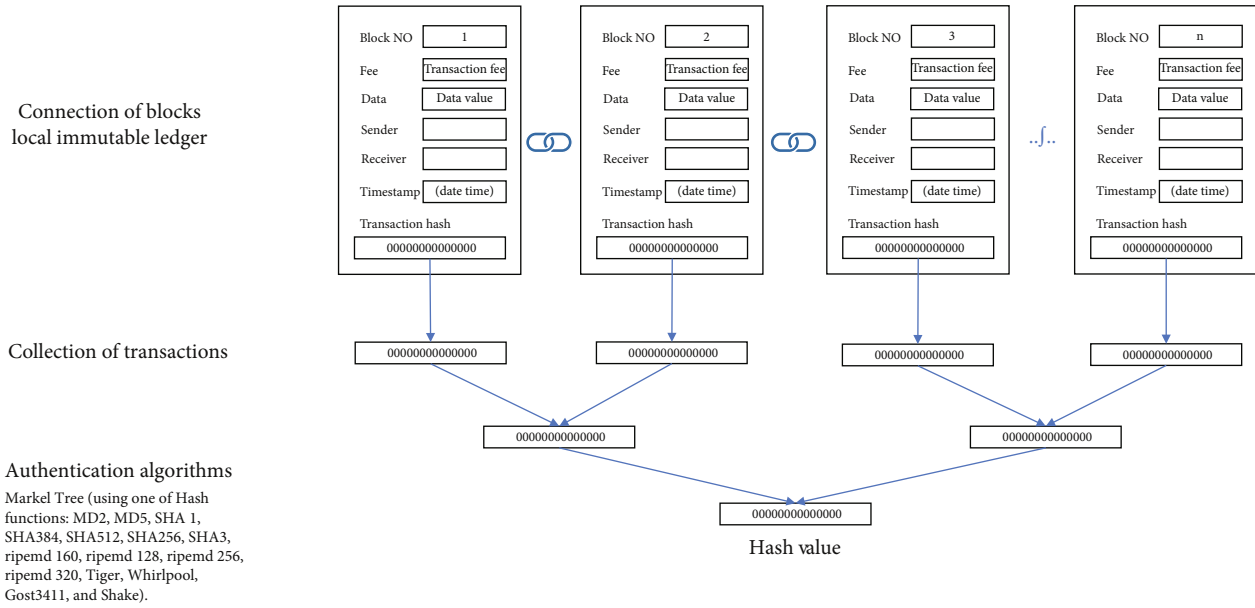


FIGURE 2: Merkle tree algorithm visualization.

functions. They can find as many additional messages with that same hash function as they desire, with no greater difficulty. Using Merkle hash tree (MHT) in blockchain is more effective than CHT by increasing complexity, and validators can calculate hashes progressively as they receive transactions from their peers.

MHT summarizes all the transactions in a block by using a mathematical data structure composed of hashes of different data blocks. It allows an efficient and secure verification in the blockchain of a large transaction hash. As a fundamental part of blockchain, MHT benefits by providing a means to maintain the integrity and validity of data and helping in saving the memory or disk space as the proof, computationally easy and fast, and their proofs and management require tiny amounts of information to be transmitted across networks.

Figure 2 shows the MHT block connection and the algorithm structure. If the transactions are valid, they are included in the new block that miners on the home network can mine. The miners produce a hash of a block by the process of changing the nonce and time stamp. In the blockchain, the system then compares the generated hash with the target. As soon as validating of the new block is finished, this block becomes part of the chain. After checking the hash's value below the target value, the PoW is verified as a successful transaction and then added to the block [65]. Subsequently, an update notification concerning this change of the blockchain size is broadcasted to the whole network for the ledger to inform every connected nodes [66].

Similarly, another research is conducted in [67] by the researchers over the existing surveillance systems. It was noted that the current surveillance system has a high risk of data security, which demands more safety protocols to protect the privacy of the users. The current use of cloud and big data-based surveillance systems has not been very efficient at handling the data and information as they are

vulnerable to possible attacks by intruders. Therefore, [67] has proposed blockchain technology in the network of surveillance systems to ensure high security of the data. The results noted that using a MHT algorithm made the transactions safer as it imposed effective monitoring and control of the security. This proposed method is also applicable to the home networks that have a high number of users. The MHT algorithm enabled the blockchain to conduct verification checks at all the nodes to prevent any data attacks [67]. Therefore, the proposed model is very efficient at handling data security due to its high convenience of design and implementation.

**2.7. Hash Functions.** One-way hash functions are used to map any data of the arbitrary size to fixed-size values, also referred to as message integrity check (MIC), message digest, contraction functions, compression functions, cryptographic checksum, fingerprint, and manipulation detection code (MDC). In the Merkle tree, the hash value is used as a Merkle root as the tree is created bottom-up using the individual transaction hashes. 15 popular hash functions were used in the implementation, which includes the following: MD2, MD5, SHA 1, SHA384, SHA512, SHA256, SHA3, RIPEMD-160, RIPEMD-128, RIPEMD-256, RIPEMD-320, Tiger, Whirlpool, GOST3411, and SHAKE (SHA with KEccak) [68].

### 3. Home Area Network Blockchain-Based Security System Model

**3.1. Architecture Components.** Node.js version (v12.16.2) and NPM version (6.14.7) have been used at the server side, where the local blockchain operates. At this place, the smart contract is also developed and deployed with the help of NPM. In addition to this, various tools and plugins such as the Truffle framework version (5.1.37) for deployment, migration, and management of smart contracts are used.

MetaMask add-on in the Chrome browser has been used for visiting the distributed web in the browser. Besides, the Solidity programming language has been used to develop smart contracts. Figure 3 and Table 3 show the components used in the local blockchain along with the network and server elements and versions.

After setting up all the server nodes, all the network components are connected in a local blockchain provided by Ganache. The Ganache version (2.5.4) has been used in this study. The accounts provided by this blockchain network are added to MetaMask to start transactions and make the blockchain network fully operational.

The simulation is implemented on a laptop with specifications given in Table 4. An i7 8<sup>th</sup> generation processor has been used along with 16 GB RAM to enhance the network’s performance. Also, SSD storage is used to ensure that the data can be processed and stored at high speed.

**3.2. Smart Home Ecosystem.** The smart home IoT ecosystem data is majorly generated from the sensors connected to different devices and electronics. The computing nodes with the central processing unit have to process these data collected from the sensors. These nodes then send the processed data to the transceivers, allowing the information’s transfer to other nodes or other associated devices. In addition to this, the actuators work according to the decisions made by the competing nodes. These actuators may be electromechanical in nature that allow them to receive data from the nodes and use it to operate different devices on the network. Therefore, through this practice, the data collected from the sensors can be used to trigger different devices’ function based on the decision made by the computational nodes [15].

On the other hand, the IoT smart home control systems can be messaging based, such as Telegram, Blynk, and web or they can work as a voice command, for instance, Google Assistant, Apple kit, and Amazon Echo [69]. A decentralized structure is implemented in the current study along with smart contracts using Solidity in a Truffle framework. This structure is then deployed into Ganache, a blockchain network. The nodes in this network are designed to perform a transaction through web-based services on NodeJS server, the frontend of web3js with a combination of HTML, CSS, and JavaScript. The network’s backend is designed using a combination of Truffle environment for blockchain development and management with Ethereum Virtual Machine (EVM), smart contract, deployment, and binary management, as well as NodeJS web servers with a node package manager (NPM). The NPM manages the users’ requests and calls for transactions outside the context of a frontend in the migration of ABI bytecode for the compiler to deploy the smart contract. Figure 4 shows a smart home ecosystem, where all the devices are connected to a single network that controls all the devices’ operations.

The nature of blockchain technology is usually decentralized. However, in smart homes, the blockchain has a central authority to ensure efficient control and monitoring of all the devices on the network [7]. The decentralized nature of blockchain is based on the distributed transactions between

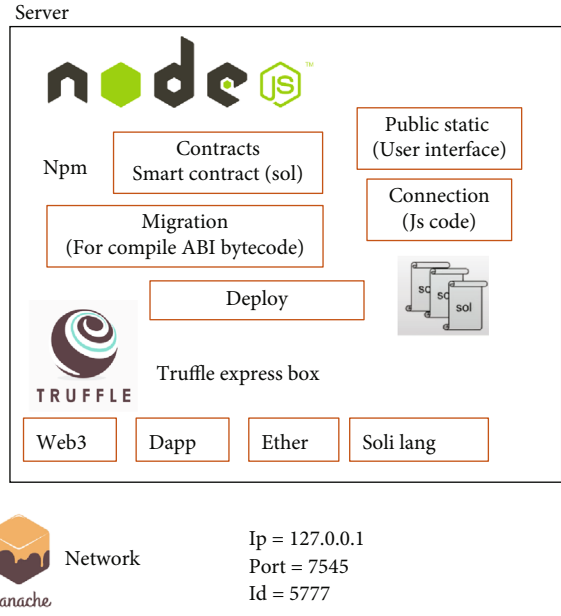


FIGURE 3: Local server blockchain components.

TABLE 3: Summary of server component versions.

Component	Version	Role
Node.js	12.16.2	Blockchain backend
NPM	6.14.7	Package management
Truffle	5.1.37	Smart contract development tools
Ganache	2.5.4	Distributed local network

TABLE 4: Summary of local server specifications.

Component	Description
CPU	Intel(R) Core (TM) i7-8550U CPU @ 1.80 GHz 1.99 GHz
RAM	16.0 GB Speed 2133 MHz
OS	Windows 10 Pro, version 20H2, 64-bit operating system, x64-based processor
Disk type	SSD SAMSUNG MZVLB512HAJQ-000L7

the blockchain network-participating nodes. These transactions are not stored in a single node or a storage device. Also, there is no central authority to approve the transactions as they are assessed according to the blockchain algorithm’s specific rules. Therefore, this removes the need for a central authority in a smart home network to carry out transactions or reach a consensus in the network. In addition to this, blockchains allow only new verified blocks to be added to the old chain. The existing blocks in the blockchain are already public and distributed, which makes them openly verifiable. Hence, they cannot be changed or revised. Thus, the blockchain security is another major benefit for the home networks as they allow the data to be kept safe [70].

**3.3. Smart Contract Design.** A smart contract is an essential part of the blockchain network used to ensure fair and

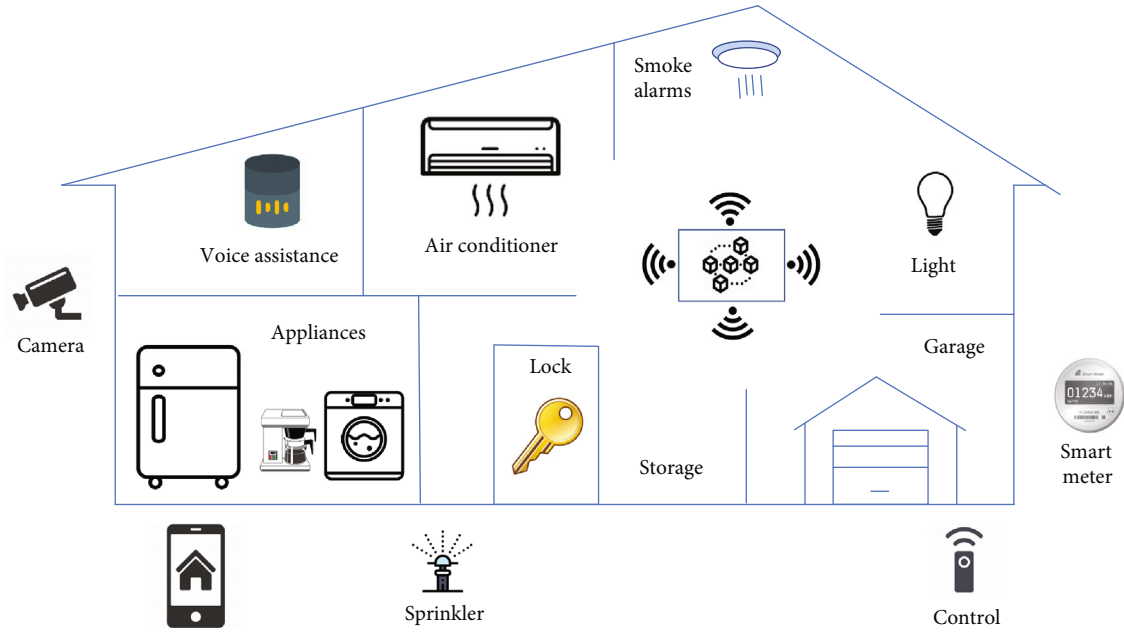


FIGURE 4: Example of a smart home network.

secure transactions among users. Therefore, a smart contract is also used in this research to make the network highly efficient. Smart networks are usually written in a domain-specific language such as Solidity so that they can be used to reach a consensus among all the peers in the network [42]. The smart contract serves a variety of functions in the blockchain, including registering all network and node components, receiving queries and transactions from various peers and applications, and allowing them to access the blockchain network's private ledger and update the ledger database. The different functions in the smart contract system is shown in Figure 5.

The hash value of the transactions is the key to the smart contract's verification process as the latter uses it to ensure all the terms have been complied. The smart contract includes the functions with a specific requirement of the input parameters [43]. The peers must fulfil these requirements on the network to make changes to the private ledger. In the nonfulfillment of the input parameters, the smart contract automatically rejects the peer's query. The smart contract takes decisions based on the code stored in it, which are also referred to as the rule of the network [38]. Therefore, a similar smart contract has also been used in our proposed research work to gain highly efficient results. Figure 5 shows the smart contract design used in this study used to communicate with the IoT blockchain network private ledger and client application.

The smart contract mechanism is illustrated in Figure 6 as an executable code stored in the blockchain that defines and manages the operation between smart devices and the identity of all kinds of users, from homeowners to local miners and normal users.

**3.4. Process Flowchart.** Figure 7 shows the process workflow of the PoW consensus algorithm. The workflow consists of

five layers: the authentication layer, access control layer, data integrity layer, availability layer, and signature layer. According to the authentication layer, the registration of all the devices on the IoT network is necessary to ensure that all the devices are part of the system. This practice can ensure that all the devices are ready to track and perform the transactions.

The access control layer contains the core system components explained previously in the HAN system model section, which determines the process workflow into different scenarios and is linked to other model layers. The registration process takes place at the sponsor, which is usually a server or many servers in the case of a distributed system. In addition to this, a smart contract is the core of the blockchain system to achieve data integrity for all transferred data between the nodes in the network. The transacted data is managed by a data integrity algorithm, which will be explained in Section 5.

Subsequently, validity check is made to approve the transaction handled in the data integrity layer. Next, in the availability layer, the private ledger is managed and updated based on the consensus algorithm's results. Finally, the users and validators can use and verify the performance based on the framework and the used technology in the signature layer.

## 4. Dataset

The dataset is a collection of various transactions between ten nodes in the network, sorted in tables. These datasets include transactions along with data about transaction hash, block no., timestamp, date and time, data value, and transaction fee, as shown in Figure 8. The total number of hashes for implementation is (30,703), compiled in ten groups.

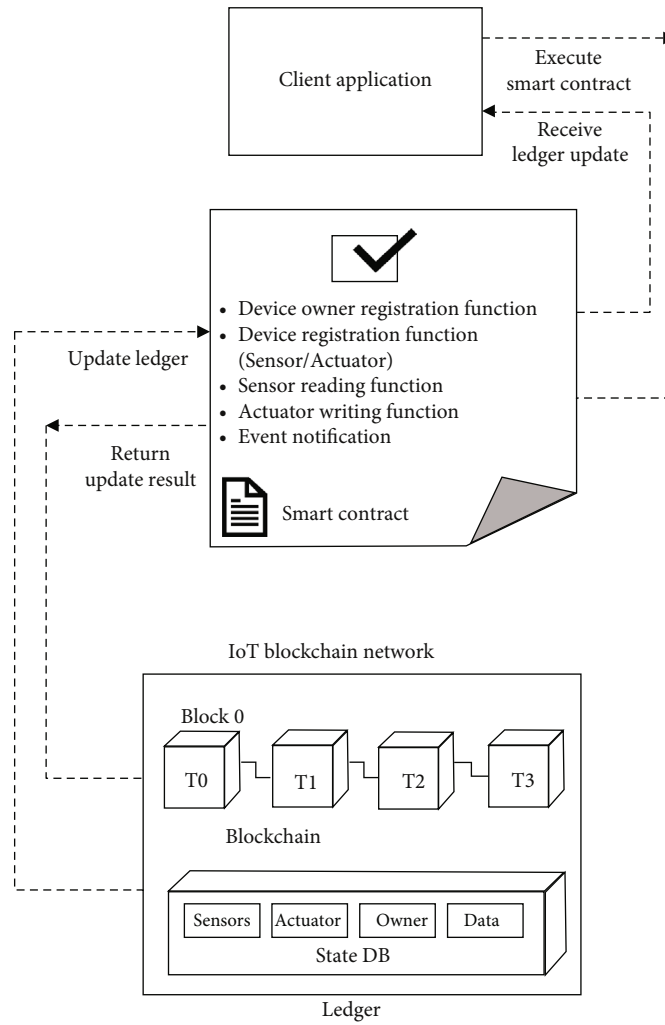


FIGURE 5: General smart contract mechanism.

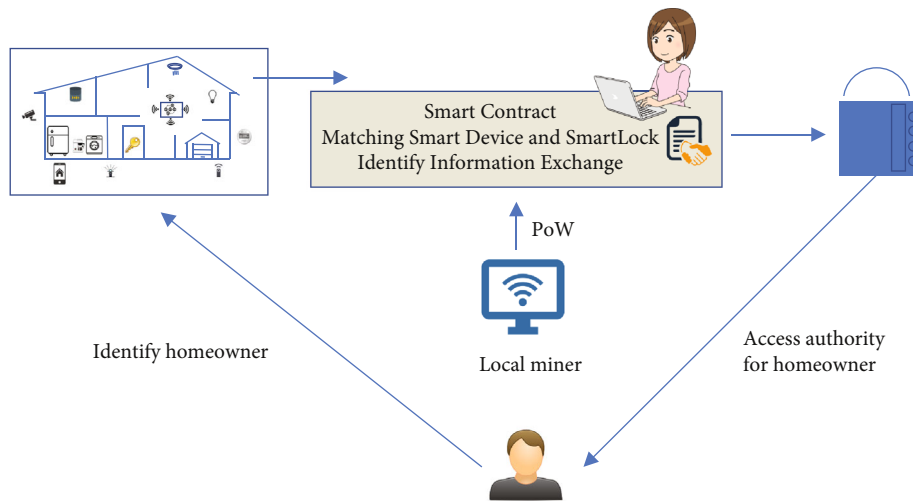


FIGURE 6: Smart contract mechanism for HAN.



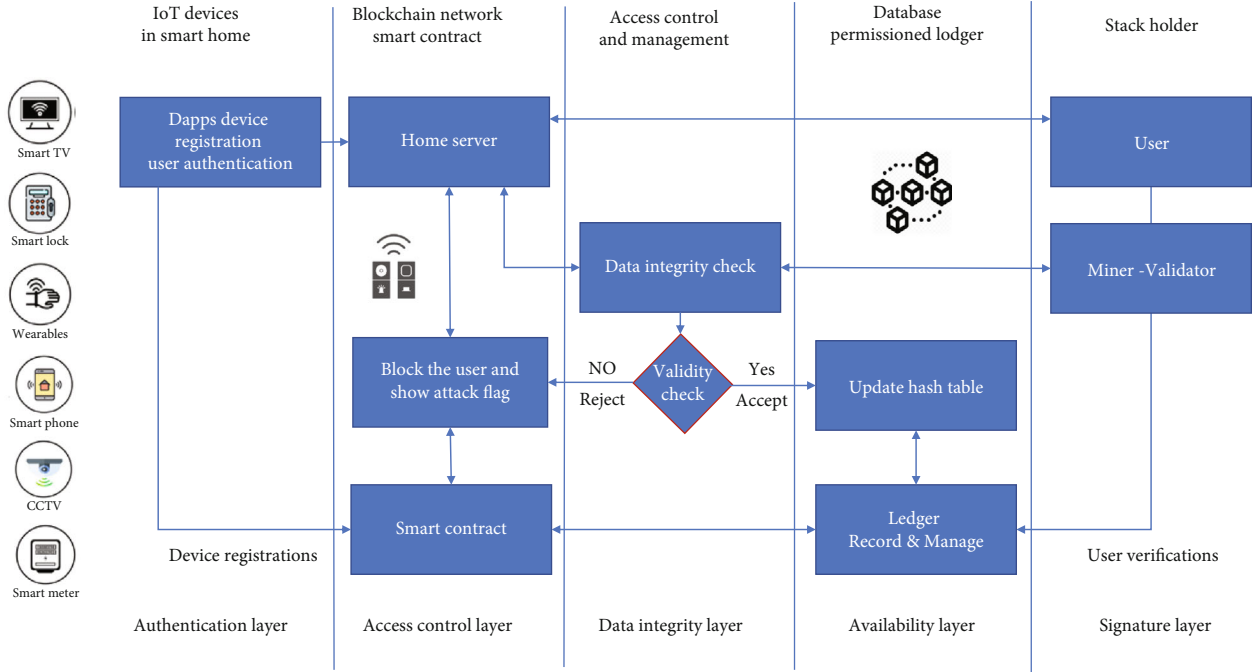


FIGURE 7: Process workflow of the PoW consensus algorithm.

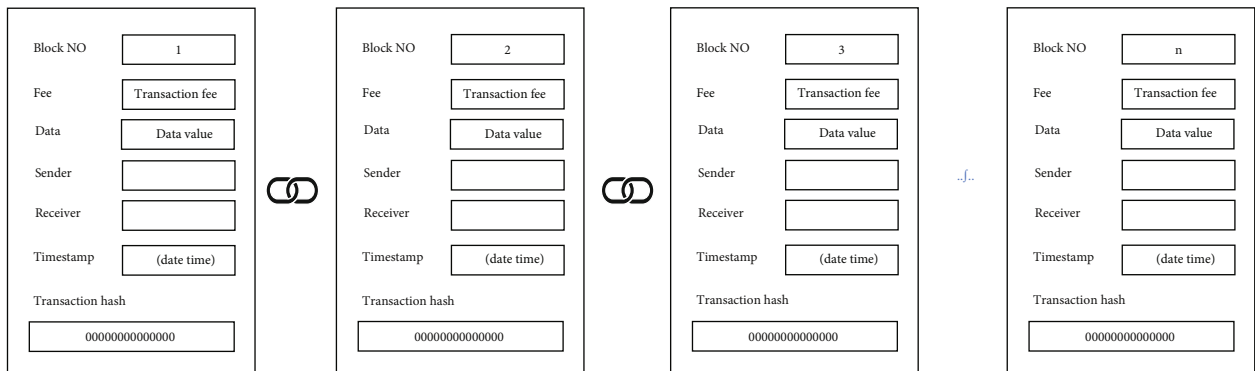


FIGURE 8: Dataset block structure.

The blockchain dataset structure and size of transactions are shown in Figures 8 and 9 [71].

Data cleaning and normalization have been performed in this work for removing unwanted records such as double spending and errors. Through this practice, an effective evaluation of the proposed system model has been conducted to produce accurate and realistic results.

In the initial stage after obtaining the selected dataset, the data is preprocessed to remove invalid data. Next, they are stored data of blockchain network transactions across a distributed architecture. One of the major objectives includes testing the system’s data integrity procedures to examine the effect of each consensus algorithms based on applying different hash functions.

Next, the dataset was split into three different sizes to check the network’s performance in data integrity and transaction scalability. Hence, the datasets were based on 30, 3000, and 30000 transactions.

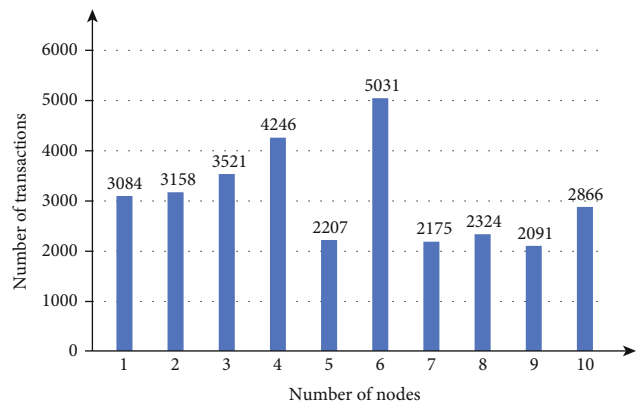


FIGURE 9: Number of transactions in the dataset.

The researchers conducted a similar evaluation in [17], which performed a similar test on the blockchain network using real home situations. Two smart home networks were

set up in different rooms, and data was collected from three different sensors in each of the rooms. The data was collected and processed through an IoT-based blockchain system that used web3.js to interact, while the smart contract was designed on Solidity. Their work is different from our current framework because it used a small volume of data and reliance on a single hash function and PoW consensus algorithm. Therefore, their results are significantly different from our current findings. The work in [17] tested the network's ability to prevent any attack on the network. It has experimented on a potential attack on the network by inserting correct and incorrect data parameters for the smart contract. The results found that the blockchain network peers were able to make effective transactions only when the correct parameters were provided. However, in the case of wrong parameters, the smart contract blocked the query of the peers. Therefore, it can be noted from [17] that smart contracts can be highly efficient in handling the security of smart home networks.

Moreover, another previous work conducted by the researchers in [18] used the Merkle tree to analyze security data of a blockchain network. The analysis was based on a network of surveillance cameras connected through a blockchain network. The work mainly relied on using the Merkle tree algorithm for authorization of the transactions between nodes rather than smart contracts. They tested the efficiency of the Merkle tree by comparing it with a similar SM tree. The results of both the trees were used to identify the network's ability to identify possible attacks on the blockchain. Some of the tested critical aspects included testing the network against data falsification attacks, message tapping attacks, and privacy masking. The results found that the use of blockchain technology in the surveillance network can make it very secure. It was noted that using the Merkle tree algorithm, the blockchain could perform rigorous authorization checks, which help in the protection against different data security threats. However, they did not use different consensus algorithms for testing the systems and relied on a single hash function and consensus algorithm. Therefore, the results of [18] are less reliable than our current work that uses multiple consensus algorithms and hash functions for testing the effectiveness of blockchain against potential data security threats.

## 5. Performance Evaluation of Proposed Consensus Algorithm

Different consensus algorithms were used in our proposed system model to compare the results and test the effectiveness and efficiency. Hence, this helped identify the most efficient consensus algorithm for the blockchain network and the ability to enhance data security and integrity. Some modifications were also made to the algorithms to increase their performance and overcome their complexity.

In the first scenario, the transactions' values were hashed while the chain of transactions was concatenated to form the concatenated hash transactions (CHT). Through this method, the final transaction value was obtained, as shown in Figure 10.

In the second scenario, the Merkle hash tree (MHT) [72] algorithm shown in Figure 11 was used for hashing the trans-

action values. MHT is also referred as a hash tree because it is used in data verification and synchronization. Therefore, it is one of the most used methods for hashing the data. MHT has a tree-like structure in which all the nodes are of the same depth and as far left as possible. The input of the tree is mapped onto the fixed output, which is known as a hash. Thus, through this mechanism, MHT can hash large and complex data with high efficiency. MHT has also been used by previous researchers in [67] to enhance the authentication procedure of the blockchain network. MHT can be represented mathematically for  $n$  blocks as follows:

$$H_{0-n} = \text{Hash}(H_{0-1} \| H_{2-3} \| H_{4-5} \| \dots \| H_n). \quad (1)$$

Next, we attempt to modify the sequence of the values of the MHT algorithm to add complexity to the traditional MHT algorithm by considering the odd and even value sequences together, as shown in Figure 12.

Odd and even modified Merkle hash tree (O&E MHT) algorithm can be represented mathematically for  $n$  blocks as follows:

$$H_{0-n} = \text{Hash}(H_{0-2} \| H_{1-3} \| H_{4-6} \| \dots \| H_n). \quad (2)$$

Finally, a modified MHT algorithm (MMHT) shown in Figure 13 is applied to the transaction chain by combining it with two algorithms (CHT & MHT). The blockchain was divided into two groups during the analysis. The first group of the blockchain was given a specific size as it included the initial blocks until block  $(X - 1)$ . The second group began from block number  $(X)$  and ended at the last transaction block  $(n)$ . The value of  $X$  that achieves the best performance will be selected. The first group of the chain used the CHT algorithm. When a new block was added to the chain during the experiment, then block number 1 was removed from the second group and added to the first group. Throughout the test, this process was performed during the transactions; the blocks were removed from the second group and added to the first group. MHT algorithm has been used in the second group of the chain. At the end of the test, groups one and two of the chain were combined and a final hash value was used to validate the transaction as explained in Figure 13.

The main purpose of dividing the original blockchain into two groups was to increase the network's efficiency by accelerating the validation execution time required to find the total hash of the blockchain and detect any partial changes. Through this practice, the process of hashing the transactions was performed with high efficiency and speed, allowing the tests to be conducted accurately.

MMHT can be represented mathematically for  $n$  blocks as follows:

$$H_{0 \rightarrow n} = \text{CHT } H_{(0 \rightarrow (n-(x+1)))} \parallel \text{MHT } H_{((n-x) \rightarrow n)},$$

$$H_{0 \rightarrow n} = \left( H_{0 \rightarrow 1} \| H_{2 \rightarrow 3} \| \dots \| H_{(n \rightarrow x-2) - (n \rightarrow x-1)} \right), \quad (3)$$

$$\parallel H_{(n \rightarrow x) - (n \rightarrow x+1)} \| \dots \| H_n.$$

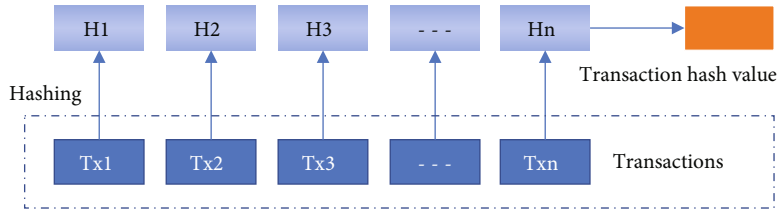


FIGURE 10: Concatenated hash transaction (CHT) algorithm.

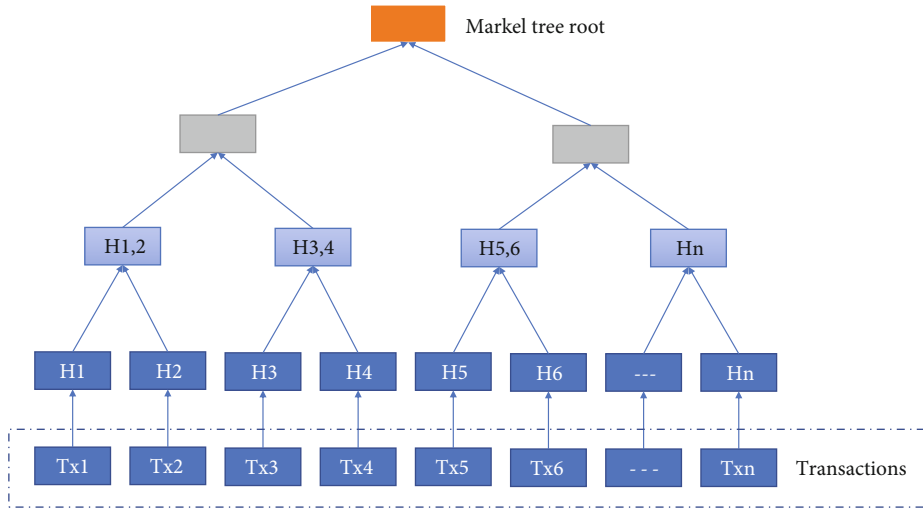


FIGURE 11: Merkle hash tree (MHT) algorithm.

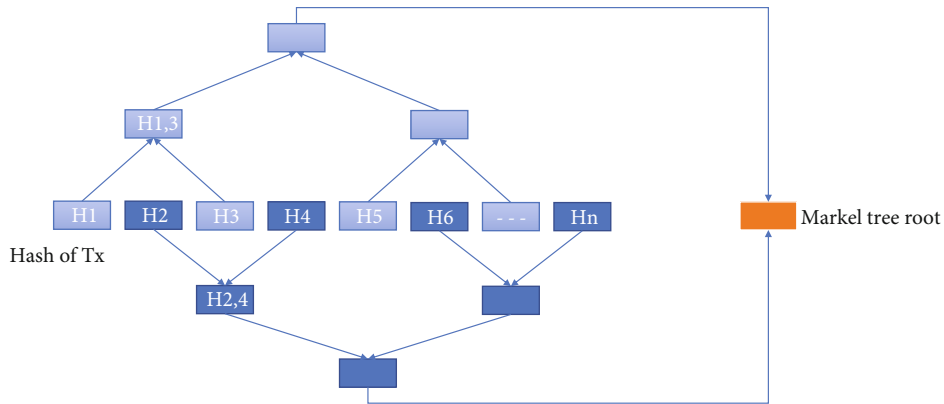


FIGURE 12: Odd and even modified Merkle hash tree (O&E MHT) algorithm.

Furthermore, another major benefit of using two different blockchain groups was that it allowed the use of a modified algorithm. This helped update the private ledger after knowing the new block’s index that was newly added to the blockchain. Therefore, only the new blocks were used to update the hash table. The benefit of this method can be seen in testing the algorithms’ effectiveness for the security of data and help in comparing the results of the two algorithms. The trigger number ( $n$ ) is used to define the first group size by knowing the significant number of changes in the execution time used in the MHT algorithm. In Figure 14, the flow of the MMHT algorithm is presented, and then, Algorithm 1 describes the process steps towards the development of the MMHT algorithm.

In this work, the evaluation of all the above consensus algorithms (CHT, MHT, O&E MHT, and MMHT) with 15 different hash functions was conducted. Furthermore, three different dataset sizes (30, 3 k, and 30 k) to check the data integrity performance of the network at different transaction scalability were investigated. This represents different models of blockchain transactions for a specific system. The results on the validation execution time using different consensus algorithms and different transaction sizes are presented in Tables 5 and 6, considering various hash functions with a length of 128 bits to 512 bits.

The analysis is performed based on an average of 11 simulation runs to obtain an accurate result with a significant

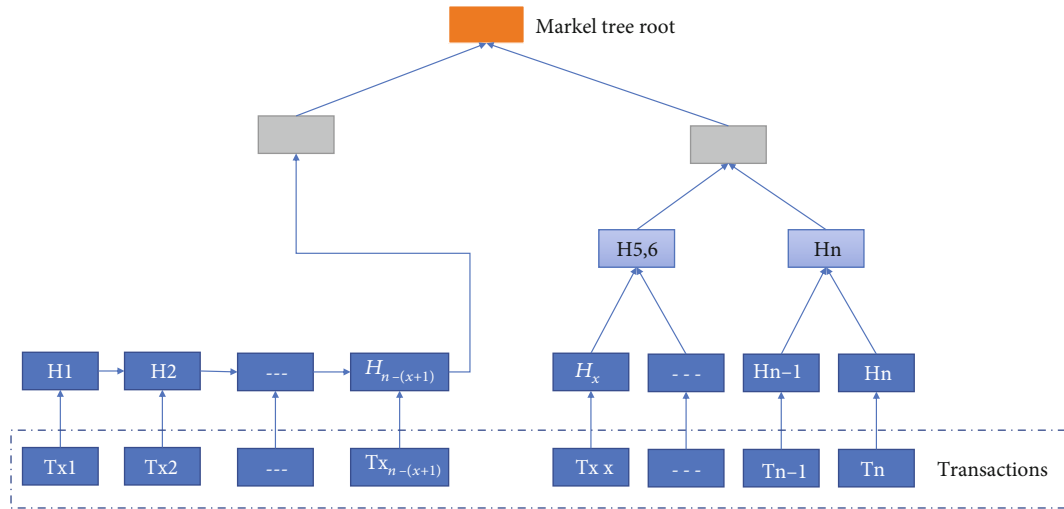


FIGURE 13: Modified Merkle hash tree (MMHT) algorithm.

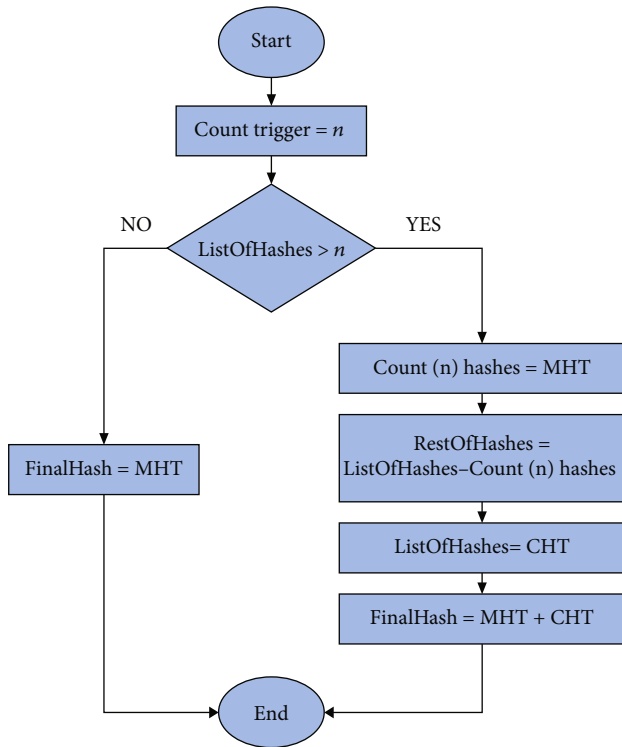


FIGURE 14: MMHT algorithm flowchart.

confidence interval of 90%. The method of averaging the results of the last ten runs was used to ensure the results' high accuracy and credibility. This practice was used as it was noted that all the execution gave different results and were not fixed. Therefore, to gain more reliable results, the program runs were averaged to produce efficient results.

For the  $n$  dataset size of transactions, it is shown that the CHT algorithm has the lowest execution time compared to any tree structure MHT, O&E MHT, and MMHT. However, it is impractical for a blockchain implementation due to the

requirement of the entire copy of the blockchain ledger in real time [47]. By using any hash functions, the test runs sequentially from one block to the other.

The results from 30 transactions recorded in Table 5 show the execution time in milliseconds (ms) and the improvement percentage mentioned in the column (%) compared to the proposed MMHT algorithm against the conventional MHT algorithm. Note that the table is categorized into hash length because it has a direct impact on the execution time of the authentication consensus algorithm and for a fair comparison between different hash functions of the same length. The higher the hash length, the higher the theoretical execution time. Then, the same hash length can be compared between hash families. Values in green highlight the best execution time in each family, while the grey ones show the overall best value.

Under the category of a 128-bit hash length, MD5 gives the best performance for all consensus algorithms. Meanwhile, SHA1 for CHT and MMHT has the best execution time compared to other hash functions regardless of hash length. However, the newer SHA2 family (SHA256, SHA384, and SHA512) do not have good execution time performance. This is due to the increase in the computational processing and number of rounds applied in the more complex hash algorithm. RIPEMD-256 gives the best performance for the 256-bit hash length category. However, most improvement (65%) in time optimization between the proposed MMHT and a conventional MHT is observed for GOST3411 hash function. The higher processing time required by the GOST3411 is based on the HMAC (hashed message authentication code) protocol. The results showed a significant benefit of using the proposed MMHT algorithm compared to the MHT algorithm. However, the proposed O&E MHT does not offer much advantage over the conventional MHT.

The analysis also considered the storage size, which is an important factor affecting the blockchain network's performance. It was noted that the storage size is fully dependent on the hash function length used in the chain. Therefore, there is a trade-off between security robustness and the low-capacity smart home IoT devices.

<p><b>Input:</b></p> <ol style="list-style-type: none"> <li>1. List of Hashes (listOfHashes)</li> <li>2. Count trigger = n</li> <li>3. Selected algorithm</li> </ol> <p><b>Algorithm Steps:</b></p> <ol style="list-style-type: none"> <li>1. Start Process, Initialize Count (n) hashes</li> <li>2. Condition, listOfHashes &gt; Count (n) hashes, if YES go to step 3, if NO go to step 7</li> <li>3. Set Count (n) hashes by taking top count(n) transactions from listOfHashes. The remaining transactions are set to restOfHashes.</li> <li>4. Use Merkel Root to hash Count (n) hashes based on selected algorithm and add it to the restOfHashes.</li> <li>5. Use CHT to hash the restOfHashes based on selected algorithm, getting the final hash. Go to Step 8.</li> <li>7. Use MHT to hash listOfHashes based on selected algorithm, getting the final hash. Go to Step 8.</li> <li>8. End of Process</li> </ol>
--

ALGORITHM 1: MMHT algorithm process steps

TABLE 5: Consensus algorithm execution time using a dataset size of 30 transactions.

Hash length	Algorithm	Total storage (bits)	CHT (ms)	O&E MHT (ms)	MHT (ms)	MMHT (ms)	MMHT/MHT (%)
128 bits	MD5	3840	0.05006	0.46151	0.45941	0.20045	56.4
	RIPEMD-128	3840	0.05277	0.50815	0.50605	0.24363	51.9
	SHAKE	3840	0.2185	1.50799	1.50589	0.72076	52.1
160 bits	MD2	3840	0.44345	2.7373	2.7352	1.18904	56.5
	RIPEMD-160	4800	0.07338	0.62818	0.62608	0.36012	42.5
	SHA1	4800	0.04245	0.501	0.4989	0.18341	63.2
192 bits	Tiger	5760	0.05415	0.5399	0.5378	0.29054	46.0
	RIPEMD-256	7680	0.05246	0.48365	0.48155	0.24179	49.8
	SHA256	7680	0.06199	0.57151	0.56941	0.25043	56.0
256 bits	SHA3	7680	0.26925	1.65981	1.65771	0.66007	60.2
	GOST3411	7680	1.18739	8.26976	8.26766	2.8954	65.0
	RIPEMD-320	9600	0.07626	0.70223	0.70013	0.30498	56.4
320 bits	SHA384	11520	0.04286	0.48769	0.48559	0.30443	37.3
384 bits	SHA512	15360	0.04588	0.46818	0.46608	0.22264	52.2
	Whirlpool	15360	0.37904	2.64535	2.64325	1.1103	58.0

TABLE 6: Consensus algorithm execution time using a dataset size of 30 k transactions.

Hash length	Algorithm	Total storage (bits)	CHT (ms)	O&E MHT (ms)	MHT (ms)	MMHT (ms)	MMHT/MHT (%)
128 bits	MD5	3840000	12.21466	183.1979	180.5979	103.8136	42.5
	RIPEMD-128	3840000	15.93768	211.553	208.953	101.2142	51.6
	SHAKE	3840000	105.5813	823.7975	821.1975	517.4801	37.0
160 bits	MD2	3840000	256.6967	1635.536	1632.936	1074.465	34.2
	RIPEMD-160	4800000	27.38025	281.9626	279.3626	168.3616	39.7
	SHA1	4800000	18.12102	210.1011	207.5011	123.1716	40.6
192 bits	Tiger	5760000	14.70333	191.2121	188.6121	110.3094	41.5
	RIPEMD-256	7680000	17.44287	208.334	205.734	121.6099	40.9
	SHA256	7680000	23.85505	274.2761	271.6761	160.9931	40.7
256 bits	SHA3	7680000	130.8481	825.7562	823.1562	543.7262	33.9
	GOST3411	7680000	512.3987	3650.143	3647.543	2337.47	35.9
	RIPEMD-320	9600000	26.5973	288.7437	286.1437	170.9692	40.3
320 bits	SHA384	11520000	17.62926	249.641	247.041	142.4498	42.3
384 bits	SHA512	15360000	18.74704	264.5122	261.9122	151.0031	42.3
	Whirlpool	15360000	178.0506	1251.731	1249.131	803.9161	35.6

TABLE 7: Consensus algorithm execution time using a dataset size of 3 k transactions.

Hash length	Algorithm	Total storage (bits)	CHT (ms)	O&E MHT (ms)	MHT (ms)	MMHT (ms)	MMHT/MHT (%)
128 bits	MD5	384000	1.94123	34.92615	34.10615	22.11161	35.2
	RIPEDM-128	384000	2.93489	44.16066	43.34066	20.65914	52.3
	SHAKE	384000	18.99856	152.072	151.252	18.78511	87.6
160 bits	MD2	384000	47.86639	329.0667	328.2467	20.32121	93.8
	RIPEDM-160	480000	4.69918	54.79416	53.97416	18.92513	64.9
	SHA1	480000	3.53118	38.42909	37.60909	19.87732	47.1
192 bits	Tiger	576000	2.3327	29.43644	28.61644	19.47202	32.0
	RIPEDM-256	768000	3.49019	38.84966	38.02966	17.26856	54.6
	SHA256	768000	4.26395	48.45686	47.63686	18.63771	60.9
256 bits	SHA3	768000	25.84644	161.3161	160.4961	20.6792	87.1
	GOST3411	768000	84.64851	652.9856	652.1656	19.39326	97.0
	RIPEDM-320	960000	5.27818	45.66796	44.84796	18.72902	58.2
384 bits	SHA384	1152000	3.24072	44.06398	43.24398	21.01735	51.4
512 bits	SHA512	1536000	2.94544	46.2301	45.4101	19.64456	56.7
	Whirlpool	1536000	31.0449	237.9127	237.0927	18.40894	92.2

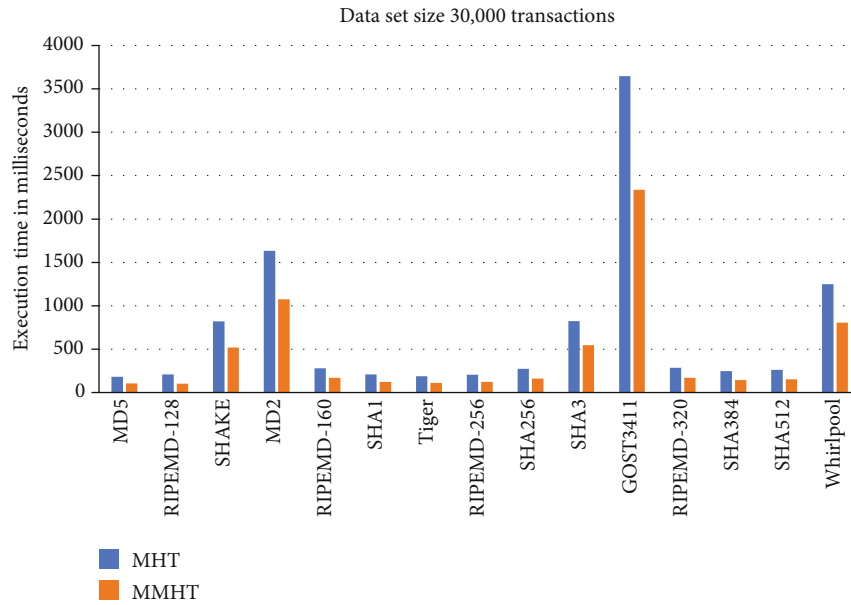


FIGURE 15: Execution time comparison graph of MHT & MMHT.

The next scenario was then conducted by increasing the size of the dataset to 3000 transactions to represent a medium-sized blockchain smart home network. The MD5 hash function was used along with the CHT consensus algorithm due to the low execution time compared to other algorithms and hash functions. The results of this medium-sized transactions have been recorded in Table 7. Similar to small transaction results in Table 5, the GOST3411 hash function gives the most time optimization gain (97%) against conventional MHT. A new trend to note, the Tiger hash function gives the best conventional MHT performance and RIPEDM-256 giving the best proposed MMHT performance. These results prove that the relationship between the number of transactions, consensus algorithm, and hash

function is not straightforward. The blockchain network needs to be designed so that it can be adaptive to different conditions in the network.

The third and final scenario was performed using 30000 transaction dataset. The results of these large-sized transactions are shown in Table 6. In the table, a more expected and stable result where the smallest 128-bit MD5 and RIPEDM-128 hash functions have the best MHT and MMHT execution time, respectively. The proposed MMHT consensus algorithm using RIPEDM-128 hash function also gives the highest time optimization gain of 51.6% compared to conventional MHT. Meanwhile, SHA1 offers the lowest execution time for the 128-bit category and RIPEDM-256 for the 256-bit category and SHA512 category. SHA3, the

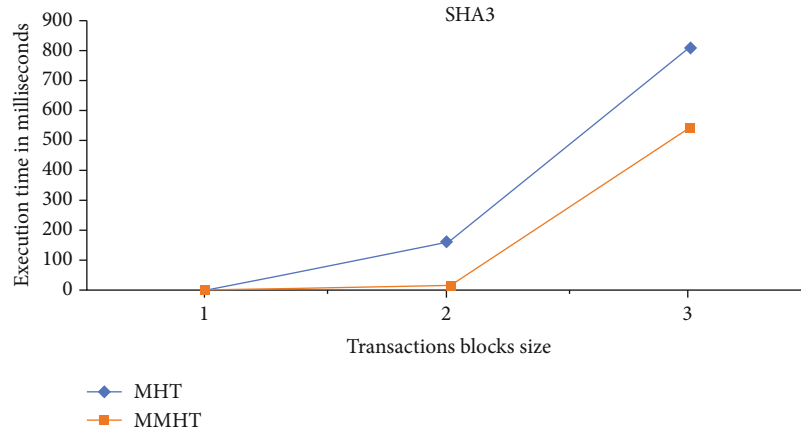


FIGURE 16: An example of execution time comparison using SHA3 with three levels of datasets.

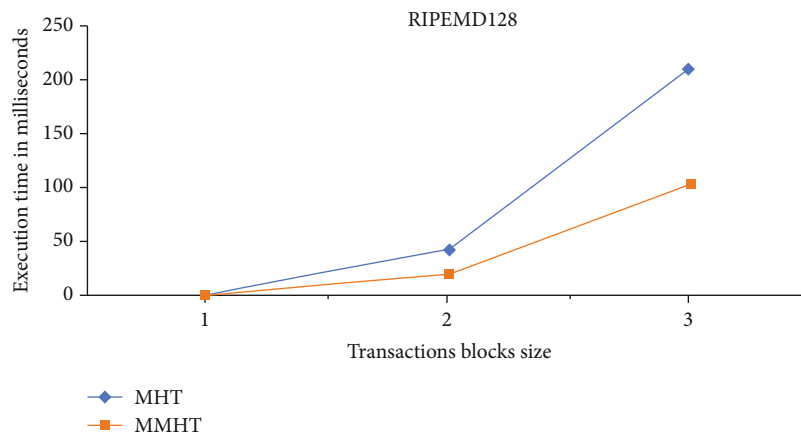


FIGURE 17: An example of execution time comparison using RIPEMD128 with three levels of datasets.

most recent version of the hash function that is frequently debated and proposed for usage presently, is slower in software implementation of an algorithm but more suitable for hardware implementation [5]; hence, it is not ideal for blockchain architecture.

As a result, we used 15 different hash functions with three different dataset sizes to show that the consensus authentication process can determine which Hash function is the best in terms of performance.

To illustrate the results, one of the comparisons of MHT and MMHT consensus algorithm execution time using 30000 transactions is shown in Figure 15.

The results shows that using the SHA3 hash function with the CHT consensus algorithm has a low execution time compared to other algorithms and hash functions, but not low enough; the use of these hash functions and consensus algorithms has a lower execution time than those of MHT and O&E MHT using the same hash function, which is not suitable for blockchain because of its complexity. Figure 16 shows an example of execution time with MMT and MMHT when comparing three different datasets using SHA3.

The RIPEMD-128 hash function with the MMHT consensus algorithm also has a low execution time, whereas the improvement percentage is 51.6% compared with that

using MHT, shown in Figure 17. It is noted that we did not consider the data maintenance functions [25] when manipulating the structure of the transaction chain. While changing of the block order in the proposed MMHT algorithm comes with an advantage of reduced execution time, it also increases the complexity of executing data maintenance functions, especially data recovery compared with the original MHT.

## 6. Conclusion

The use of blockchain-based IoT systems for smart homes can provide high security against possible data security threats. Recent studies have found that blockchain networks are highly effective and secure due to their advanced features like smart contracts, which keep a strong check over the activities and transactions over the network. The consensus algorithm based on PoW secures the network via a validator responsible for handling all communication verifications between the blockchain network nodes within the smart homes. This work is unique because to the best of our knowledge, there are no previous studies that has attempted an investigation of multiple hash functions for a consensus algorithm, as well as different data sizes for testing

blockchain performance. In terms of data integrity verification check, the results show that the proposed modified Merkle hash tree (MMHT) consensus algorithm used in the blockchain has a very efficient execution time.

This system can be effectively used by smart homes to provide the safest systems for high data security and integrity.

However, the results have shown that the relationship between the number of transactions, consensus algorithms, and hash functions is not straightforward. The blockchain network needs to be designed so that it can be adaptive to different conditions in the network. Even though the proposed MMHT algorithm gives a significant advantage in the simulation, the current study also has limitations in terms of the lack of testing of the proposed system in a real environment. In addition to this, the concern about the relationship between a smart electronic contract and its legal counterpart can cause inefficiencies and barriers to the networks' operation. The lack of a legal status for smart contracts in the current laws is a significant issue that needs further investigation.

## Data Availability

The dataset used in the implementation of this study is included and explained within the article and referenced in reference [71].

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

Part of this work is supported by the Malaysian Ministry of Higher Education and Universiti Kebangsaan Malaysia (Grant number: GUP-2021-023).

## References

- [1] P. Ii, *Part II Cryptocurrencies and Blockchain Applications Applications with Blockchain*, Scrivener Publ. LLC, 2020.
- [2] M. Dopico, A. Gomez, D. De la Fuente, N. Garcia, R. Rosillo, and J. Puche, "A vision of industry 4.0 from an artificial intelligence point of view," in *Proc. 2016 Int. Conf. Artif. Intell. ICAI 2016- WORLDCOMP 2016*, pp. 407–413, Athens, 2016.
- [3] Z. Mubeen, M. Afzal, Z. Ali, S. Khan, and M. Imran, "Detection of impostor and tampered segments in audio by using an intelligent system," *Computers and Electrical Engineering*, vol. 91, article 107122, 2021.
- [4] H. N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for Internet of things: a survey," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8076–8094, 2019.
- [5] H. Cho, "ASIC-resistance of multi-hash proof-of-work mechanisms for blockchain consensus protocols," *IEEE Access*, vol. 6, no. c, pp. 66210–66222, 2018.
- [6] M. Maslin, M. Watt, and C. Yong, "Research methodologies to support the development of blockchain standards," *Journal of ICT Standardization*, vol. 7, no. 3, pp. 249–268, 2019.
- [7] U. Bodkhe, S. Tanwar, K. Parekh et al., "Blockchain for industry 4.0: a comprehensive review," *IEEE Access*, vol. 8, pp. 79764–79800, 2020.
- [8] D. Guha Roy, P. Das, D. De, and R. Buyya, "QoS-aware secure transaction framework for Internet of things using blockchain mechanism," *Journal of Network and Computer Applications*, vol. 144, pp. 59–78, 2019.
- [9] Q. Nasir, I. A. Qasse, M. Abu Talib, and A. B. Nassif, "Performance analysis of hyperledger fabric platforms," *Security and Communication Networks*, vol. 2018, 14 pages, 2018.
- [10] J. Cynthia, H. P. Sultana, M. N. Saroja, and J. Senthil, *Security Protocols for IoT*, no. 2020, Springer International Publishing, 2019.
- [11] K. Hao, J. Xin, Z. Wang, and G. Wang, "Outsourced data integrity verification based on blockchain in untrusted environment," *World Wide Web*, vol. 23, no. 4, pp. 2215–2238, 2020.
- [12] E. Androulaki, A. Barger, V. Bortnikov et al., "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proc. 13th EuroSys '18: Thirteenth EuroSys Conference 2018*, Porto Portugal, 2018.
- [13] S. Alsaqqa and S. Almajali, "Blockchain technology consensus algorithms and applications: a survey," *International Journal of Interactive Mobile Technologies (ijIM)*, vol. 14, no. 15, pp. 142–156, 2020.
- [14] G. W. Peters and E. Panayi, "Understanding modern banking ledgers through blockchain technologies: future of transaction processing and smart contracts on the Internet of money," in *Banking Beyond Banks and Money*, Springer, Cham, 2016.
- [15] M. Moniruzzaman, S. Khezr, A. Yassine, and R. Benlamri, "Blockchain for smart homes: review of current trends and research challenges," *Computers and Electrical Engineering*, vol. 83, article 106585, 2020.
- [16] M. Salimitari, M. Chatterjee, and Y. P. Fallah, "A survey on consensus methods in blockchain for resource-constrained IoT networks," *Internet of Things*, vol. 11, article 100212, 2020.
- [17] P. M, M. Malviya, M. Hamdi et al., "5G based Blockchain network for authentic and ethical keyword search engine," *IET Communications*, vol. 2021, 2021.
- [18] S. N. Makhadmeh, M. A. al-Betar, Z. A. A. Alyasseri et al., "Smart home battery for the multi-objective power scheduling problem in a smart home using grey wolf optimizer," *Electronics*, vol. 10, no. 4, p. 447, 2021.
- [19] S. A. Maghdid, H. S. Maghdid, S. R. HmaSalah, K. Z. Ghafoor, A. S. Sadiq, and S. Khan, "Indoor human tracking mechanism using integrated onboard smartphones Wi-Fi device and inertial sensors," *Telecommunication Systems*, vol. 71, no. 3, pp. 447–458, 2019.
- [20] H. M. Kim, H. Turesson, M. Laskowski, and A. F. Bahreini, "Permissionless and permissioned, technology-focused and business needs-driven: understanding the hybrid opportunity in blockchain through a case study of insolar," *IEEE Transactions on Engineering Management*, pp. 1–16, 2020.
- [21] S. Beg, A. Anjum, M. Ahmad et al., "A privacy-preserving protocol for continuous and dynamic data collection in IoT enabled mobile app recommendation system (MARS)," *Journal of Network and Computer Applications*, vol. 174, article 102874, 2021.
- [22] H. Hosseinian, H. Shahinzadeh, G. B. Gharehpetian, Z. Azani, and M. Shaneh, "Blockchain outlook for deployment of IoT in distribution networks and smart homes," *International*



- Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 3, pp. 2787–2796, 2020.
- [23] P. Sandner, J. Gross, and R. Richter, “Convergence of blockchain, IoT, and AI,” *Frontiers in Blockchain*, vol. 3, 2020.
- [24] B. Ali and A. I. Awad, “Cyber and physical security vulnerability assessment for IoT-based smart homes,” *Sensors*, vol. 18, no. 3, pp. 817–817, 2018.
- [25] M. U. Hassan, M. H. Rehmani, and J. Chen, “Privacy preservation in blockchain based IoT systems: integration issues, prospects, challenges, and future research directions,” *Future Generation Computer Systems*, vol. 97, pp. 512–529, 2019.
- [26] S. Shetty, C. Kamhoua, and L. Njilla, *Blockchain for Distributed Systems Security*, John Wiley & Sons, Inc., 2019.
- [27] M. Liu, K. Wu, and J. J. Xu, “How will blockchain technology impact auditing and accounting: permissionless versus permissioned blockchain,” *Current Issues in Auditing*, vol. 13, no. 2, pp. A19–A29, 2019.
- [28] D. Minoli, “Positioning of blockchain mechanisms in IOT-powered smart home systems: a gateway-based approach,” *Internet of Things*, vol. 10, article 100147, 2020.
- [29] M. Yahuza, M. Y. I. B. Idris, A. W. B. A. Wahab et al., “Systematic review on security and privacy requirements in edge computing: state of the art and future research opportunities,” *IEEE Access*, vol. 8, pp. 76541–76567, 2020.
- [30] M. S. Ferdous, M. J. M. Chowdhury, M. A. Hoque, and A. Colman, “Blockchain consensus algorithms: a survey,” 2020, <http://arxiv.org/abs/2001.07091>.
- [31] Brilliant, *Merkle Tree*, Springer, 2016.
- [32] S. Khan, A. Gani, A. W. Abdul Wahab et al., “Towards an applicability of current network forensics for cloud networks: a SWOT analysis,” *IEEE Access*, vol. 4, pp. 9800–9820, 2016.
- [33] A. Niakanlahiji and J. H. Jafarian, “WebMTD: defeating cross-site scripting attacks using moving target defense,” *Security and Communication Networks*, vol. 2019, 13 pages, 2019.
- [34] H. Wang and J. Zhang, “Blockchain based data integrity verification for large-scale IoT data,” *IEEE Access*, vol. 7, pp. 164996–165006, 2019.
- [35] A. M. Sagheer, M. S. Al-Ani, and O. A. Mahdi, “Ensure security of compressed data transmission,” in *2013 Sixth International Conference on Developments in eSystems Engineering*, pp. 270–275, Abu Dhabi, United Arab Emirates, 2013.
- [36] A. Basil Ghazi, O. Adil Mahdi, and W. Badee Abdulaziz, “Lightweight route adjustment strategy for mobile sink wireless sensor networks,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 1, pp. 313–320, 2021.
- [37] T. L. N. Dang and M. S. Nguyen, “An approach to data privacy in smart home using blockchain technology,” in *Proc. -2018 Int. Conf. Adv. Comput. Appl. ACOMP 2018*, pp. 58–64, Ho Chi Minh City, Vietnam, 2018.
- [38] L. Hang and D. H. Kim, “Design and implementation of an integrated iot blockchain platform for sensing data integrity,” *Sensors*, vol. 19, no. 10, p. 2228, 2019.
- [39] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, “Security, privacy and trust in Internet of things: the road ahead,” *Computer Networks*, vol. 76, pp. 146–164, 2015.
- [40] L. König, Y. Korobeinikova, S. Tjoa, and P. Kieseberg, “Comparing blockchain standards and recommendations,” *Future Internet*, vol. 12, no. 12, p. 222, 2020.
- [41] R. Leszczyna, *Cybersecurity in the Electricity Sector*, Springer, 2019.
- [42] I. Karamitsos, M. Papadaki, and N. B. Al Barghuthi, “Design of the blockchain smart contract: a use case for real estate,” *Journal of Information Security*, vol. 9, no. 3, pp. 177–190, 2018.
- [43] B. K. Mohanta and D. Jena, “An overview of smart contract and use cases in blockchain technology,” in *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–4, Bengaluru, India, 2018.
- [44] M. Maximilien, A. Vallecillo, J. Wang, and M. Oriol, “Service-oriented computing,” in *15th International Conference, ICSOC 2017*, pp. 229–237, Malaga, Spain, November, 2017.
- [45] F. Daniel and L. Guida, “A service-oriented perspective on blockchain smart contracts,” *IEEE Internet Computing*, vol. 23, no. 1, pp. 46–53, 2019.
- [46] H. Wang, X. A. Wang, S. Xiao, and J. S. Liu, “Decentralized data outsourcing auditing protocol based on blockchain,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 2703–2714, 2021.
- [47] R. Kalis and A. Belloum, “Validating data integrity with blockchain,” in *2018 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pp. 272–277, Nicosia, Cyprus, August, 2018.
- [48] R. Zambrano, “Taming the beast: harnessing blockchains in developing country governments,” *Frontiers in Blockchain*, vol. 2, pp. 1–15, 2020.
- [49] B. Bhushan, A. Khamparia, K. M. Sagayam, S. K. Sharma, M. A. Ahad, and N. C. Debnath, “Blockchain for smart cities: a review of architectures, integration trends and future research directions,” *Sustainable Cities and Society*, vol. 61, article 102360, 2020.
- [50] S. J. Alsunaidi and F. A. Alhaidari, “A survey of consensus algorithms for blockchain technology,” in *2019 International Conference on Computer and Information Sciences (ICIS)*, pp. 1–6, Italy, 2019.
- [51] A. Meneghetti, M. Sala, and D. Taufer, “A survey on pow-based consensus,” *Annals of Emerging Technologies in Computing*, vol. 4, no. 1, pp. 8–18, 2020.
- [52] T. Alam, “IoT-Fog: a communication framework using blockchain in the Internet of things,” 2019, <http://arxiv.org/abs/1904.00226>.
- [53] K. S. Gorniak and A. M. Kudin, “Aspects of blockchain reliability considering its consensus algorithms,” *Theoretical and Applied Cybersecurity*, vol. 2, no. 1, pp. 5–9, 2020.
- [54] S. M. H. Bamakan, A. Motavali, and A. Babaei Bondarti, “A survey of blockchain consensus algorithms performance evaluation criteria,” *Expert Systems with Applications*, vol. 154, article 113385, 2020.
- [55] L. Chen, L. Xu, N. Shah, Z. Gao, Y. Lu, and W. Shi, “On security analysis of proof-of-elapsed-time (PoET),” in *Stabilization, Safety, and Security of Distributed Systems*, pp. 282–297, Springer, Cham, 2017.
- [56] W. F. Silvano and R. Marcelino, “Iota Tangle: a cryptocurrency to communicate Internet-of-things data,” *Future Generation Computer Systems*, vol. 112, pp. 307–319, 2020.
- [57] M. Salimitari and M. Chatterjee, “A survey on consensus protocols in blockchain for IoT networks,” 2018, <http://arxiv.org/abs/1809.05613>.
- [58] M. Salimitari, M. Chatterjee, and Y. P. Fallah, “A survey on consensus methods in blockchain for resource-constrained IoT networks,” *Internet of Things*, vol. 11, article 100212, 2020.
- [59] M. Bartoletti, S. Lande, and A. S. Podda, “A proof-of-stake protocol for consensus on bitcoin subchains,” in *Financial*

- Cryptography and Data Security*, pp. 568–584, Springer, Cham, 2017.
- [60] X. Fu, H. Wang, and P. Shi, “A survey of Blockchain consensus algorithms: mechanism, design and applications,” *Science China Information Sciences*, vol. 64, no. 2, pp. 1–15, 2021.
- [61] G. Kumar, R. Saha, M. K. Rai, R. Thomas, and T. H. Kim, “Proof-of-work consensus approach in blockchain technology for cloud and fog computing using maximization-factorization statistics,” *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6835–6842, 2019.
- [62] S. S. Hazari and Q. H. Mahmoud, “A parallel proof of work to improve transaction speed and scalability in blockchain systems,” in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 916–921, Las Vegas, NV, USA, 2019.
- [63] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Čapkun, “On the security and performance of proof of work blockchains,” in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 3–16, Vienna Austria, 2016.
- [64] Y. Ren, Q. Zhao, H. Guan, and Z. Lin, “A novel authentication scheme based on edge computing for blockchain-based distributed energy trading system,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, 2020.
- [65] Y. Zhao, “Research on personal credit evaluation of Internet finance based on blockchain and decision tree algorithm,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, 2020.
- [66] A. Ju, Y. Guo, Z. Ye, T. Li, and J. Ma, “HeteMSD: a big data analytics framework for targeted cyber-attacks detection using heterogeneous multisource data,” *Security and Communication Networks*, vol. 2019, 9 pages, 2019.
- [67] D. Lee and N. Park, “Blockchain based privacy preserving multimedia intelligent video surveillance using secure Merkle tree,” *Multimedia Tools and Applications*, 2020.
- [68] B. Schneier and B. Schneier, “Cryptography in context,” in *Secrets and Lies: Digital Security in a Networked World*, pp. 102–119, Wiley Publishing, Inc., 2015.
- [69] H. Isyanto, A. S. Arifin, and M. Suryanegara, “Design and implementation of IoT-based smart home voice commands for disabled people using Google Assistant,” in *2020 International Conference on Smart Technology and Applications (ICoSTA)*, Las Vegas, NV, USA, 2020.
- [70] M. H. Miraz and M. Ali, “Integration of blockchain and IoT: an enhanced security perspective,” *Annals of Emerging Technologies in Computing*, vol. 4, no. 4, pp. 52–63, 2020.
- [71] B. Podgorelec, *Dataset of Transactions of 10 Ethereum Addresses Controlled by a Private Key, Each Has At Least 2000 Output Transactions, Which Include a Transfer of Cryptocurrency, and All Transactions Are Performed within No Longer Than Three Months Period*, 2019.
- [72] M. S. Niaz and G. Saake, “Merkle hash tree based techniques for data integrity of outsourced data,” *CEUR Workshop Proceedings*, vol. 1366, pp. 66–71, 2015.

## Review Article

# A Review of Big Data Resource Management: Using Smart Grid Systems as a Case Study

Muhammad Fawad Khan,<sup>1</sup> Muhammad Azam,<sup>2</sup> Muhammad Asghar Khan ,<sup>3</sup>  
Fahad Algarni ,<sup>4</sup> Mujaddad Ashfaq,<sup>5</sup> Ibtihaj Ahmad,<sup>6</sup> and Insaf Ullah<sup>3</sup>

<sup>1</sup>School of Computer Science and Engineering, Kyungpook National University, Daegu, Republic of Korea

<sup>2</sup>School of Energy and Power Engineering, Jiangsu University, Zhenjiang 212013, China

<sup>3</sup>Department of Electrical Engineering, Hamdard University, Islamabad, Pakistan

<sup>4</sup>College of Computing and Information Technology, University of Bisha, Saudi Arabia

<sup>5</sup>Department of Electrical Engineering, Foundation University Islamabad, Rawalpindi Campus, Pakistan

<sup>6</sup>Department of Computer Science Engineering, Northwestern Polytechnical University, Xi'an, China

Correspondence should be addressed to Muhammad Asghar Khan; [m.asghar@hamdard.edu.pk](mailto:m.asghar@hamdard.edu.pk)

Received 31 August 2021; Revised 2 October 2021; Accepted 12 October 2021; Published 22 October 2021

Academic Editor: Suleman Khan

Copyright © 2021 Muhammad Fawad Khan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Big data has recently been a prominent topic of research due to the exponential growth of data every year. This massive growth of data is causing problems with resource management. The available literature does not address this problem in depth. Therefore, in this article, we aim to cover the topic of resource management for big data in detail. We addressed resource management from the perspective of smart grids for a better understanding. This study includes a number of tools and methods, such as Hadoop and MapReduce. Large data sets created by smart grids or other data-generating sources may be handled using such tools and approaches. In this article, we also discussed resource management in terms of various vulnerabilities and security risks to data and information being transmitted or received, as well as big data analytics. In summary, our comprehensive study of big data in terms of data creation, processing, resource management, and analysis gives a full picture of big data.

## 1. Introduction

Over the past 20 years, data has been increasing tremendously in different fields. According to the International Data Corporation (IDC), the total copied and created data volume all over the world was 1.8 ZB (zettabytes), which has increased by approximately nine times within five years [1]. And there is a prediction that in the near future, this figure will double at least every other two years. Considering these statistics, one can well imagine about the drastic growth of big data and the issues related to it. Big data deals with huge data sets mostly in exabytes, zettabytes or yottabytes. Figure 1 can give a better estimate of these mega units that represent an enormous scale of volume. In Figure 1, these higher units of volume are converted to bytes in order to get a better esti-

mation and clear picture of the huge volume of data sets in big data.

The enormous increase of data is generating resource management issues. Resource management is a technique which is used to utilize the resources in efficient way by improving the network throughput, capacity, robustness, and efficiency. Importance of management of resources in data applications is growing day by day.

Big data is linked with a huge amount of data sets. Most of the big data comprise a huge amount of unstructured data sets as compared to traditional data sets that require more real-time data analysis [2]. Additionally, big data help us in categorizing the data looking at different aspects considering the value of the data. It creates new opportunities for effectively organizing and managing such enormous data sets

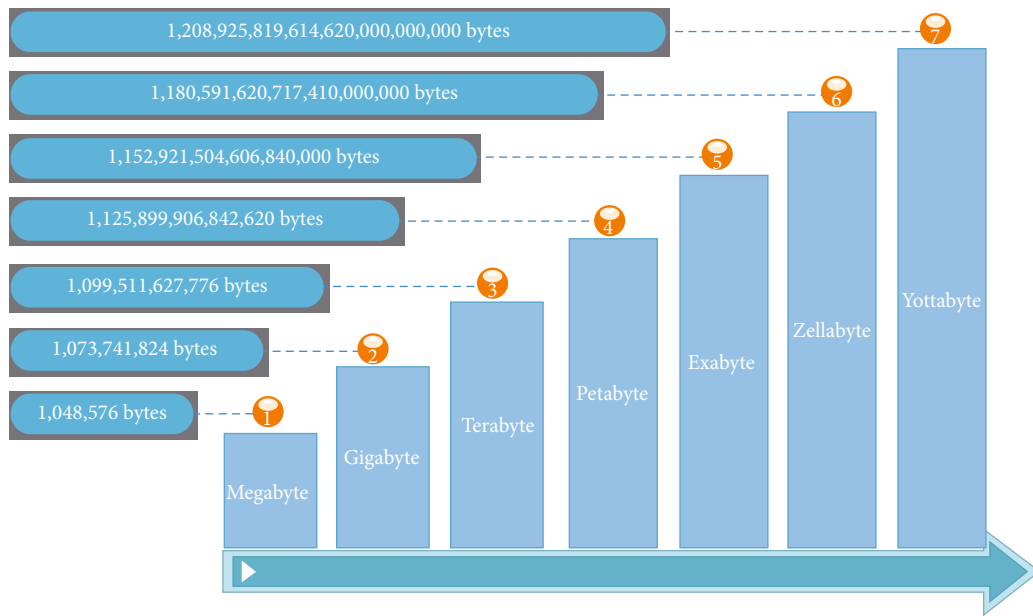


FIGURE 1: Bigger units of volume converted to bytes [4].

according to their value [1]. There are many applications of big data in the smart world. Everything is shifting from off-line to online and cloud systems. Smart grid systems are a technological innovation whose adaptation has recently increased all over the globe. Two-way communication flow makes it more efficient, reliable, sustainable, and cost effective as compared to traditional power grid systems. But two-way data flow of a huge number of sensors is a big challenge for data scientists [3]. Due to limitation of resources, resource management is needed to get the effective results in every field. In many different fields of communication and IT, work on resource management has been done to a great extent. Different aspects of big data have been highlighted in the existing literature as shown in Table 1.

There are four contributions in this article. To begin, we will go through data generation sources. The second contribution is a consideration of the relevance of resource management in the context of smart grids, as well as the sources of big data. Third, we go through the strategies and tools that go into analyzing various cases. Finally, we discuss unresolved difficulties and obstacles in large data analysis in general.

**1.1. Data-Generating Sources.** Big data includes large data sets produced by different applications and devices. The umbrella of big data covers various fields; some of them are given as follows.

- (i) *Black box data*: the black box of jets, airplanes, helicopters, etc. captures voices of the flight crew, performance information, and recordings of microphones of the aircraft

- (ii) *Social media data*: social media websites like Facebook, LinkedIn, and Twitter hold the information of millions of people across the globe [5].
- (iii) *Stock exchange data*: it also contributes in generating a huge number of data sets comprising the information regarding buying and selling of shares of different companies
- (iv) *Smart grid data*: one step ahead of a typical power grid is the smart grid that also generates information at an enormous scale [6]
- (v) *IoT*: the internetworking of devices, sensors, and applications works on the principle of information exchange among the devices and hence is a leading contributor towards big data [7]. Issues in the IoT-based smart grid are that it uses internet-based protocols and infrastructure of public communication which are more exposed to security threats [8].
- (vi) *Search engine data*: search engines like Google, Yahoo, and Bing also create a huge amount of data

**1.2. Why Big Data?** Big data, an emerging and one of the most important technologies in the world of internet, IoT, mobile networks, wireless sensor networks (WSN), smart grid systems, medical and health monitoring systems, etc. Big data have several benefits:

- (i) The limitation of fossil fuels and natural resources has raised the demand for efficient energy generation, distribution, and monitoring systems. In response to requirements, the smart grid is a

TABLE 1: Comparison with existing related surveys.

Ref	Research area	Year	Remarks	Issues identified	Possible solutions
[9]	Scope and privacy	2013	This paper presents scope and privacy concerns in big data.	Privacy and security issues	✗
[4]	Applications and challenges	2014	This comprehensive survey covers communication and business applications as well as challenges and technologies.	Volume of data	✓
[10]	Clustering algorithms	2014	This survey provides a number of algorithms related to clustering. Comparison of existing clustering algorithms has been included.	Limitations of data clustering algorithms	✗
[11]	Platforms for big data analytics	2015	This study offers a survey on available platforms for big data analytics. Pros and cons of each platform are explored.	Drawbacks of different data processing platforms	✗
[12]	Mining algorithm	2015	It presents brief introduction of data analytics and the mining algorithm to extract the useful information from big data.	Issues related to platform, framework, security, privacy, and data mining perspective have been highlighted.	✗
[13]	Parallel processing	2016	This survey paper presents an overview of parallel processing and highlighted the processing efficiency of different cases.	Novel data, processing model, energy efficiency, and large-scale machine learning	✗
[14]	Networking for big data	2017	This survey provides the introduction of networking in big data as well as networking features, challenges, and opportunities.	Big graph mining, dynamic representation, time evolution, security, privacy, and scheduling for big data related to networking perspective	✗
[15]	Modern computing paradigms	2017	New computing paradigms are discussed for big data in the IoT case and limitation of cloud computing for the IoT applications. Data base management systems based on NoSQL are investigated for different authorizers.	Storage, management, security, privacy, computation, and resource performance	✗
[16]	Big data issues in smart grids	2019	This article highlights issues related to big data analytics, technologies, and architectures in next-generation power systems.	System, data management, and analysis	✗
This article	Big data resource management in smart grids	2021	Big data in the domain of a smart grid is explored and the resource management for smart grid applications is discussed. Techniques, tools, and challenges are also elaborated.	Volume, data integration, storage and visualization from multiple sources, data backup, privacy, security, confidentiality, energy management, and quality	✓

technological advancement that is a solution to the energy crisis. The generated big data from the smart meter in terms of volume, variety, and velocity would be very much beneficial for efficient utilization of energy as well as for better energy planning [17]

- (ii) Different companies of marketing agencies use big data resource management strategies in order to improve the response of their campaigns, promotions and other advertising mediums [18], and information of the social network like Facebook [5]
- (iii) Hospitals are providing quick and better services using the information regarding the previous medical history of patients [19] and predicting the future health conditions using big data analysis in the domain of health care [20, 21]

*1.3. 5 Vs of Big Data. Volume:* the first “V” is the large volume of the data clusters of big data. The data is so large that it cannot be analyzed by any conventional methods. Five versions of big data are shown in Figure 2. The data is increasing at a very fast rate, and according to experts, 78% of the current data on social websites has been produced in 5 years since 2011 making it the largest data generated to date. Other examples include the following: Facebook produces 500 terabytes of data on a daily routine, according to a report of the IDC USA; the increase of data will be 400 times by now in 2021. The explosion of data which has been collected in e-commerce is 10 times more in quantity of an individual’s data transaction [22].

*Variety:* variety targets the type of data that we have. It may be a structured, semistructured, or unstructured data set. However, the majority of big data is unstructured that is randomly generated by multiple sources. Big data is not

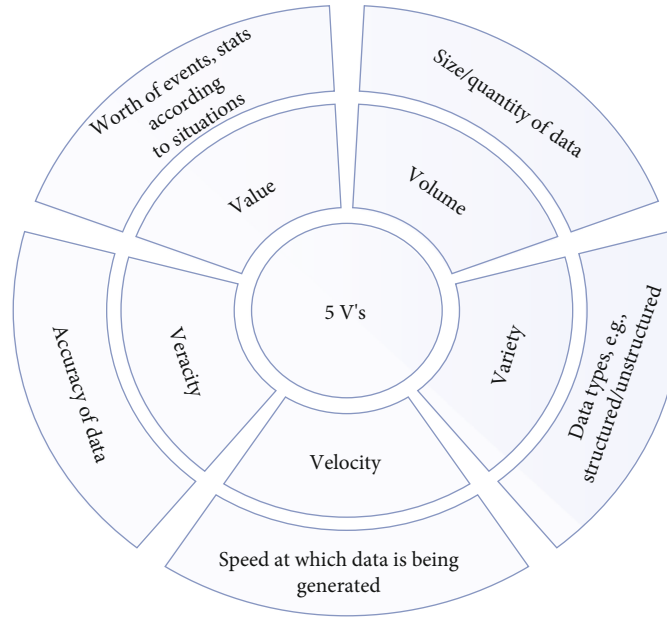


FIGURE 2: 5 Vs of big data.

just bits and pieces; it is much more than that. Big data includes audio, video, 3D data, and unstructured text, including log files and social media. The traditional data includes lower volumes of consistent and structured data.

*Velocity*: the third V is velocity, which deals with the pace of data that is being generated by different sources like machines, mobile networks, business processes, and human interaction with things like social media sites and internet banking. The information flow is continuous and massive as well. Handling this rate at which data is being generated provides a strong basis for valuable decisions. It leads toward rapid interpretation and strategic competitive advantages to help businesses and researchers from this real-time data.

- (i) Clickstreams and ads capture a large amount of data, e.g., millions of events per second
- (ii) It takes a fraction of seconds to reflect market changes for high frequency stock trading algorithms
- (iii) Online gaming produces huge amount of data from millions of concurrent users producing multiple inputs per second

*Veracity*: uncertainty or inaccuracy of data can be dealt under the 4<sup>th</sup> V of big data that is termed as veracity [23]. Data veracity refers to the abnormality and noise in the data. It also deals with whether the stored data is meaningful to the analysis or not. As compared to velocity and volume, it is the observation of the community that veracity in data analysis is the supreme challenge.

*Value*: value is also very important when business models are considered. Data should be analyzed in accordance to the value of data to get the best result out of the analysis. Value is critical for business initial phases, because it is the matter of investing money and reducing the risks. Still, many companies are not using this application of big

data in effective way. Better use of this application will not only be effective for revenue generation but also help to avoid fraud.

*1.4. 5 Vs and Smart Grid*. Smart grid incorporates conventional power systems with a bidirectional infrastructure that integrates electricity and information flow. The smart grid is a complex interconnected system that generates a diversified variety of data with huge volume, high velocity, and veracity. These 5 Vs are worthy of importance when we discuss automated electric grid systems [24].

*1.5. Major Contribution*. Our focus is to provide a comprehensive survey on big data for smart grid applications. The contributions of this work are summarized as follows:

- (i) General overview of big data and smart grid systems
- (ii) Big data-generating sources in the smart grid
- (iii) Importance of resource management
- (iv) Tools and techniques for the analysis of big data
- (v) Research challenges

Most of the existing literature on big data and smart grid mainly focus on its applications, issues, tools, technologies, and techniques separately, but big data resource management in the context of the smart grid has not been explored so far. References [16, 25] targeted the issues related to SG, but the management perspective is missing in the literature. Different domains of big data are being targeted in various reviews, but a comprehensive survey is not present in the existing literature that covers a holistic picture of big data. This paper has been written in such a way that it clears the complete picture of big data for the beginner in this field. Unlike other available literature on the big data smart grid,

TABLE 2: Comparison with existing work.

Ref	Techniques/tools	Smart grid	Issues in smart grid	Resource management discussed	Challenges and opportunities discussed
[1]	×	×	×	×	✓
[3]	×	✓	×	✓	×
[8]	×	✓	✓	×	✓
[17]	×	✓	×	×	✓
[22]	✓	×	×	×	✓
[26]	×	✓	✓	×	✓
[27]	×	×	×	✓	✓
[28]	×	✓	×	✓	×
[29]	×	×	×	✓	✓
[30]	×	×	×	×	✓
[31]	✓	✓	×	×	✓
[32]	×	×	×	✓	✓
[33]	✓	✓	×	×	✓
[34]	✓	✓	✓	×	✓
[16]	✓	✓	✓	×	✓
[25]	✓	✓	✓	×	✓
This article	✓	✓	✓	✓	✓

it not only covers the main topic but gives a clear holistic picture of the importance of big data and resource management in smart grid networks. Table 2 represents a comparative analysis of this article with existing literature available. Uniqueness or real contribution of this article is clearly judged by Table 2.

*1.6. Article Structure.* The paper's organization is shown in Figure 3 and is described in the later sections. In Section 2, we discuss the smart grid systems and sources generating big data like sensor and information flow of different applications. The motivation for deploying resource management is presented in Section 2.1. In Sections 3 and 4, we have discussed different techniques and tools like Hadoop/MapReduce.

## 2. Smart Grid Systems

Smart grid power systems are new innovative power systems which will not only provide more electricity to meet the increasing demand but also improve reliability, efficiency, and quality. This system will allow other individuals to add their energy in the national grid which includes many energy sources like renewable energy resources (solar, biogas, wind, etc.) [35]. Traditional power distribution systems transport energy to the consumer side from a central power plant using transmission lines [24]. Major stakeholders of smart grid systems are the distribution, transmission, consumption, and communication networks. The communication network is actually the main portion that converts the conventional grid to a smarter one. There is a two-way communication between the distributor and the consumer in smart grid networks. This information exchange and continuous

monitoring of energy enables efficient utilization of power in emerging smart grid networks [36, 37].

Smart grid systems enable the grid to observe and control the power parameters accurately. This system also offers to make decisions on time as well as allows us to integrate renewable systems. In this advancement of the grid system, communication technology plays a pivotal role as depicted in Figure 4. It establishes a strong link between the distributor and the consumer to make the network more efficient.

Reliable and efficient distribution of electricity is a basic requirement with essential energy production units. The power grid infrastructure was deployed in early ages and is now reaching full life. For more competition, there is a need for strong political and regulatory push, lower energy prices and more energy efficiency, and greater use of renewable energy like biomass, water, solar, and wind to keep the environment clean. The load demand has remained the same or has slightly increased in the previous years in industrial countries. Some of the developing countries show a rigorous increase in load demand. But now, load demand is increasing exponentially due to more industries and increasing population [38]. On the other side, aging equipment may lead to shortfall of electricity during peak hours. In different parts of the world, regulators are advising utilities to find the cost-effective solution for transmission and distribution of electricity. That is why new techniques like the smart grid (based on modern communication) are emerging to operate power systems which guarantee a secure, sustainable, and competitive energy supply. The important goals of an advanced electrical grid are to ensure an environment-friendly, transparent, and sustainable system. Utilization of renewable energy resources are worthy of importance in order to meet the above-mentioned goals of smart grid systems.

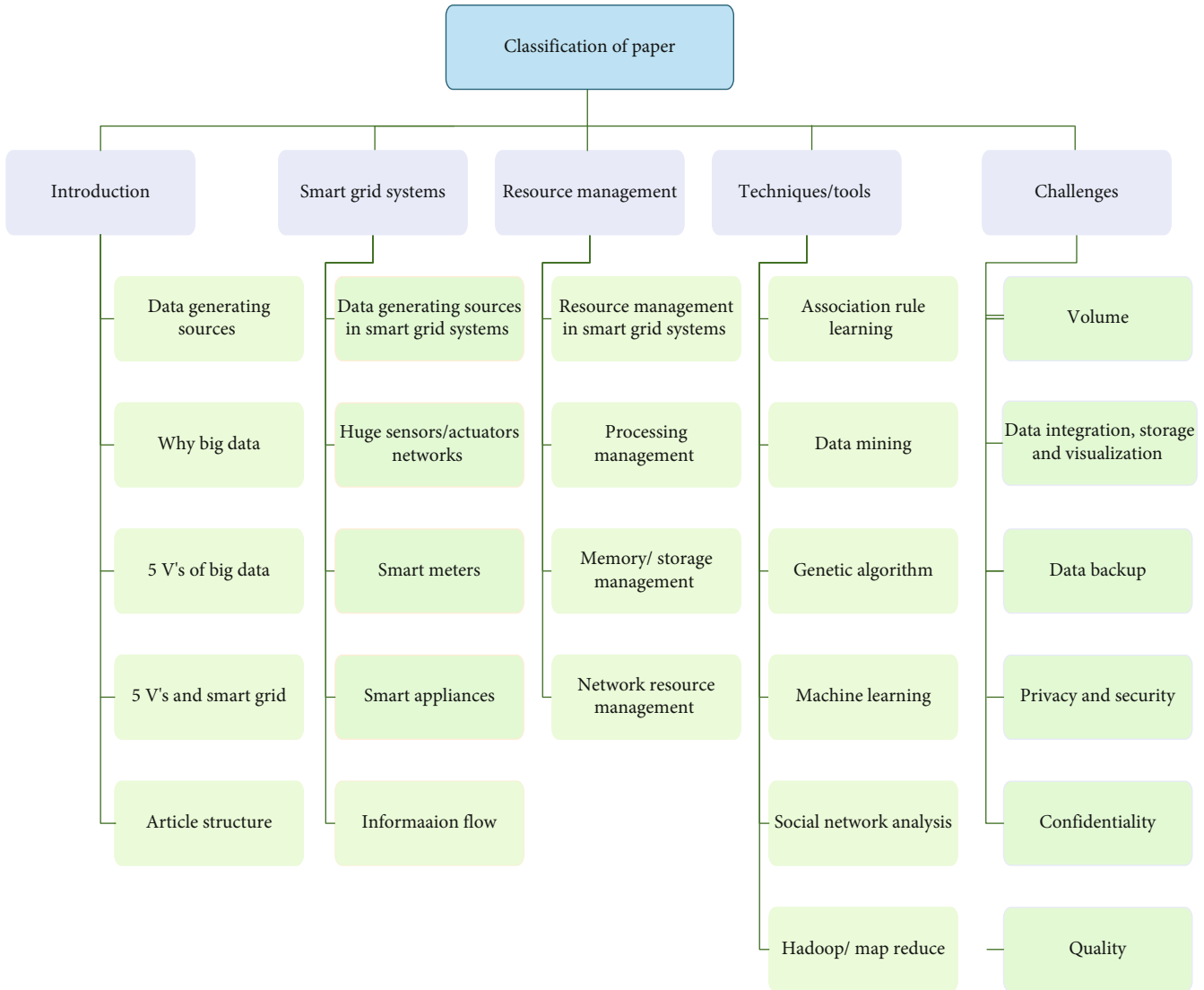


FIGURE 3: Organization of paper.

2.1. Types of Data-Generating Sources in Smart Grid Systems

2.1.1. *Huge Sensor/Actuator Network.* A smart monitoring system is actually a strong source that generates huge data sets. It is impossible to implement a smart monitoring infrastructure without using low-cost but intelligent devices. A new scheme of sensors termed as smart sensors has recently been introduced that fulfils the criteria discussed above, i.e., low cost, ultralow power, and more intelligence [39, 40]. The importance of smart sensors has been discussed in detail in [41], and a new type of sensor termed as “stick on” was investigated. These sensors do not even need physical contact with the utility asset for some applications. They have the capability to monitor different parameters of interest only by getting close to utility assets. Fang et al. have also discussed the self-powering smart sensors and challenges related to that domain.

2.1.2. *Smart Meters.* A smart grid comprises smart meters that play a very important role. A grid, by definition, is an

electric system that includes electricity generation, transmission, distribution, and consumption. A traditional power grid system comprises a typical setup that supplies electricity to users and consumers by carrying that power from a few central generators [42]. One main advantage of the Smart Meter System is their simple operation of the overall process even if they are varied in technology and design. These intelligent meters gather information from end consumers every 15 minutes or once a day and transmit that valuable information to the data collector through the Local Area Network (LAN). Arif et al. [43] developed a smart meter based on GSM and ZigBee. These meters are capable enough to update the information of the service provider about the energy measurements. The service provider can use this information to notify their consumers via Short Message Service (SMS) or using the internet. A hardware architecture is presented in [44] which discussed the adapted communication protocol and monitored the energy using web-based application. Managing the energy in smart grid systems using a mobile application is investigated in [45] to improve



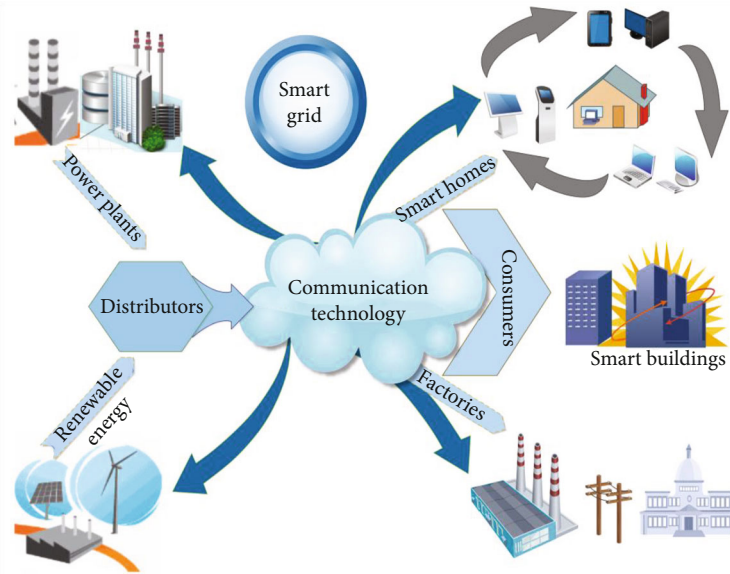


FIGURE 4: Smart grid look.

the availability and data exchange. Policies and security concerns vary from country to country which has been discussed in [46], and the smart grid development in different countries has been compared.

**2.1.3. Smart Appliances.** The smart grid system has done a lot in aligning the electricity demand and supply during peak hours by promoting small-scale renewable energy generation [47]. Talking about smart houses, the key element is the smart cards that are responsible for communication between the smart meter and appliances. These smart cards act as a communication link for the transference of information. The town server holds the connection of the number of such smart houses and is responsible for controlling the power provided by the service provider and the power generated by regenerative sources. A town server network has been discussed in [48] which manages the communication and the whole power consumption between the systems. A smart house architecture is presented in [49] which is proposed for a demand-responsive energy management system based on Information and Communication Technology (ICT).

**2.1.4. Information Flow.** In smart grid systems, communication or information plays a crucial role in making decisions. Normally, decisions are based on the collected information. In power systems, most of the time, information plays a very critical role. The grid is becoming smarter with the passage of time by the use of modern technologies which facilitate bidirectional information sharing between customers and the utility [50]. The smart grid consists of sensors, actuators, smart meters, control units, computers, etc. The information of all these sources flow from one point to another. Effective management systems are necessary to manage the information of these heterogeneous complex and bulk data networks. In [51], Suci et al. examined the cyberphysical system (CPS) from an information flow perspective. A

method is presented to analyze the leakage of information by using the advice tape concept in the field of algorithms.

### 3. Management Perspective

**3.1. Resource Management.** Resource management in every field is very important to optimize many parameters. In Figure 5, the variety of resource management processes are shown. Resource optimization is a supreme parameter to minimize the cost and improve efficiency. Normally, the resource is in the form of a spectrum which is sparse due to the exponential increase in the user devices. Resource management is an effective and efficient allocation of resources in any platform. User devices are increasing exponentially with the times and generating a lot of distributed data in various forms. Data handling is a big challenge for researchers. Without efficient management in big data applications, it is very hard to tackle such huge data. A huge research space is available for exploring resource management in big data.

In big data, resource management in the sense of memory and complexity is rarely explored. Different applications create 2.5 quintillion bytes of data every day [52]. The amazing thing is that 90 percent of the data in the world has been created in the last two years. This data includes all applications like sensors used to gather information about climate, posts to social media sites, cell phone GPS signals, digital pictures and videos, purchase transaction records, etc. Different aspects in big data like resource management, processing, analytics for social media, database technique packing algorithms, security, and privacy concerns are covered in [53]. Speed of information technology growth is increased from Moor's law at the beginning of the 21st century. Excessive data is creating more challenges in data science. On the other side, data science is extremely important to produce productivity in businesses which will create a lot of opportunities. Reference [4] discussed a closed-up

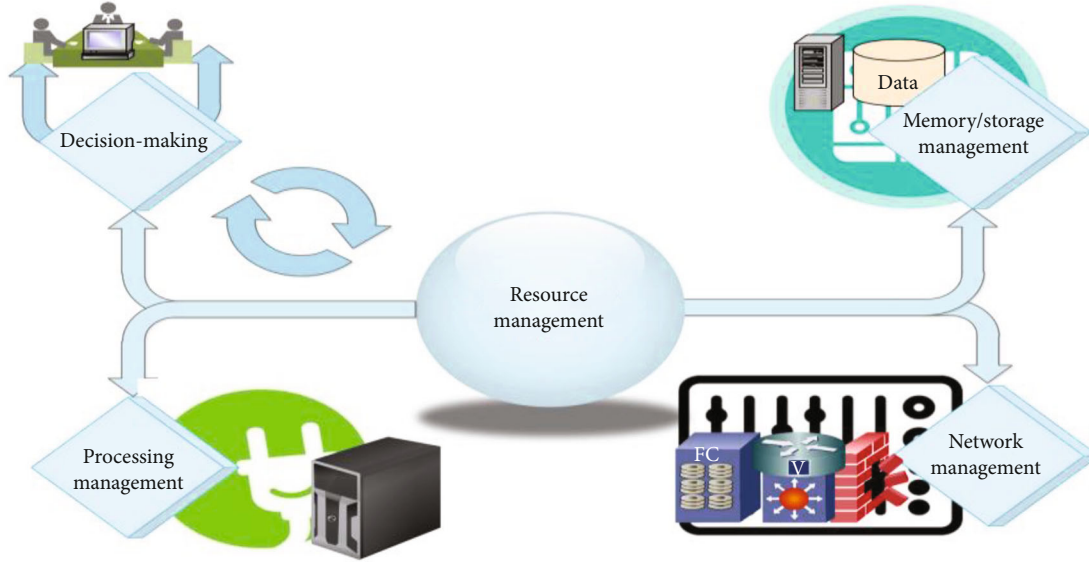


FIGURE 5: Resource management.

view about big data including opportunities, applications, and challenges. Chen et al. have also discussed techniques and technologies used to deal with big data problems. In [1], Chen et al. explained in detail about the background of big data, related technologies, data storage, applications, issues, and practical applications. In [54], different aspects of resource management for big data platforms have been examined. Pop et al. also discussed the importance of resource management for smart cities. The smart grid is the part of smart cities which will enable the use of energy in more efficient ways. Resource management for big data applications is an open issue for current development in the era of the smart world. Predictive resource planning and allocation discussed in [55] is energy saving and will ultimately save on costs. Won et al. in [56] investigated the advance resource management for multiple tenants using access control to share the computing resources. In this environment, multiple tenants having different demands can share computing resources like data, storage, network, memory, and Central Processing Unit (CPU). Researchers claimed that multitenancy reduces cost and offers highly effective saving computing resources to acquire a similar environment for data management and processing. The novel approach is used to support the multitenancy features for Hadoop. It is understood that Hadoop is a large-scale distributed system which is commonly used for the processing of data. Resource management in big data is rarely covered in the community although popular literatures and academia have many examples on initiatives of big data. Senior managers are hesitant to commit resources on data sciences on a sustainable basis. Reference [57] covered the theme of improving organizational resource management and created a concentration to attain a positive capability with initiatives of big data. The relationship between dynamic capabilities and big data is of great significance because processes of data need to be developed step by step

as organizations want new insights from big data. The smart grid is a growing technology in the power system which also needs data or information management to efficiently utilize the resources which is investigated in [3]. They discussed how to manage different types of front-end intelligent devices like smart meters and power assets.

*Resource management in smart grid systems:* the exponential increase in the data has prompted many challenges to develop systems to manage the resources. It is required to manage resources to analyze the huge amount of data efficiently. It is impossible to manage the big data in traditional ways. There is a need to manage the resource like processing, memory, and network resources so that we could be able to process the complex data systems in comfortable ways in emerging smart grid networks.

**3.2. Processing Management.** Processing in big data plays a pivotal role to analyze the data to extract the required results. Big data-processing techniques process data sets of terabytes or even more than that. Processing is further divided into distributed and parallel processing using traditional application frameworks like OpenMP and MPI which are still playing an important role. Newly investigated big data processing and cloud computing frameworks like Spark, Hadoop, and Storm are becoming popular. But in the parallel application framework, it requires a physical cluster to run the system efficiently. A resource-sharing approach using a cluster as a service for a private cloud has been discussed in [58]. The ClaaS model is proposed to make the implementation simple. Authors claimed that it is an effective model for sharing a cluster between several frameworks. Parallel processing systems like batch, graph, stream, and machine learning techniques have been discussed in [13] in which optimization and extensions for the MapReduce platform are also discussed. There are many platforms that are developed for processing purpose, but

[59] developed a pipeline structure for the heterogeneous execution environment to integrate data jobs. To integrate data jobs in a heterogeneous execution environment, developers need to write a long glue code to get data sets into and out of those jobs. For the data pipelining and integration support, some frameworks are also proposed such as Crunch, Cascading, and Pig, but these frameworks are built on top of a single data-processing execution environment. Resource management in big data regarding business purpose is an important aspect that either these initiatives are helping managers to grow their business and make it more profitable or not. It is invested in [57] which classifies the organizational resource management in three aspects. First is to establish a business process archetype and second to create a dynamic capability and identify the drawbacks of the resource-based theory. Lessons are learned, and the implications for business research and practice are sorted out. An applied example to apply the data techniques to smart cities had been investigated in [60], and an IoT-based architecture was proposed. Some services implemented in the smart campus of Murcia University and some services are focused on tram service scenarios where thousands of transaction data are considered.

*3.3. Memory/Storage Management.* Growing memory capacity has accelerated the development in memory of big data processing and management [27]. Real-time data analytics require intelligent memory or storage systems which have the least latency to read or write the data. Initially, the need of this type of performance was encountered by well-known global companies like Google, Amazon, and Facebook, but now, it is becoming an obstacle for other organizations which are looking for a meaningful real-time service like social gaming, advertising, and real-time bidding. To meet the requirements in real time for analysis of large data sets in milliseconds requires RAM memory. Bandwidth, capacity, and memory storage have been doubling after every three years while its price is dropping by a factor of ten every 5 years. Noteworthy advances have been observed in non-volatile memory (NVM) e.g., SSD. Hardware technology advancement in recent times has generated interest in hosting the whole database as well as overturned many earlier works [61] in memory to provide real-time analytics [62, 63] and faster access. Comprehensive memory management and some key factors to achieve efficient memory data management and processing are investigated in [27]. There are privacy and security implications [64] of pervasive memory augmentation which effect what and how humans radically change the scale and nature of external cues. The presence of ubiquitous displays in personal devices and environment provides new opportunities for showing memory cues to trigger recall.

*3.4. Network Resource Management.* Recent findings show that human behavior is highly predictable [65]. Improving the performance in wireless systems by exploiting the predicted information draws an attention which is known as anticipatory, context aware, and predictive resource allocation in the literature [66, 67]. Context awareness is not a

new concept in the computing science. Context is any information that can be used to characterize the situation of an entity. Entity can be anything like a place, object, or person that is contemplate relevant to the interaction between an application and a user. Energy-efficient predictive resource allocation and planning is presented in [55] based on predictive analysis and results that show that the proposed policy can drastically reduce the energy consumed by the BSs. Won et al. [56] introduced an advanced resource management (e.g., network, memory, storage, and data) with an access control in a multitenant environment. Multitenancy facilitates management of multiple users who use similar systems. Using this concept, the system is able to permit multiple users to maintain and develop their own environment; otherwise, an application is required to provide their products by manual anthropology to address the requirements of each company. The manual approach resulted in an excessive maintenance cost because it desires the management of each company separately. Hadoop Apache was used as a base platform for providing features of multitenancy.

## 4. Techniques

There are different techniques defined by the researcher to analyze big data. Researchers are continuously working on the development of new techniques as well as focusing on the improvement of existing ones. Some of the most common and regularly used techniques for the analysis of big data are

- (1) Association rule learning (ARL)
- (2) Data mining
- (3) Genetic algorithm
- (4) Social network analysis
- (5) Classification tree analysis

*4.1. Association Rule Learning.* With the evolution of data generation, new methods of data analysis are needed to carry out in-depth analysis of the clusters. One such rule is ARL. It is a method related to rule-based machine learning and used to discover interesting relations between variables in large databases. With the increase in the quantity of big data, ARL is being implemented across the globe in a number of fields to study relationships between variables to sort data according to desired variables. Its applications range from consumer markets to modern-day communication.

With the increase in the internet users, the data generation across the different levels of the World Wide Web has sky rocketed, but the internet still works on criteria and the protocols of the past. The internet cannot keep up with the recent increase in the data generation and storing. The modern-day data supports a large number of elements which can be used to create semantics to understand the trends of data. Reference [68] introduces the design of a human mind-based semantics to improve

internet decision-making associated with an analysis technique for accurate reasoning about the internet and to compare the current algorithms. As the data volume has increased, the complexity of the mobile network, furthermore, the user-based data generation and complex interrelations, has also grown. Reference [69] uses ARL to produce a deep network analyzer (DNA) for anomaly detection in order to make further improvement in the network and make an accurate gain prediction to address a wide range of problems faced by internet service providers (ISP). With the increased span of the internet and complexity of the networks, the dark network has also increased. Since its dawn, the dark net has been the cornerstone of illegal activity across the global networks resulting in huge loss of value from patents of companies to breaching of the defense networks of countries. This new dawn of internet volume explosion has made it easier for cyberattacks on critical infrastructure. ARL can be used for the analysis of the data clusters of the dark net, predicting relationships between a number of factors such as malusers and beneficiaries of such activities.

Talking about the smart grid, it is understood that it uses a vast network of smart sensors. These sensors are responsible of generating a huge volume of data that must be categorized in a mathematical or scientific way to make this advanced network more efficient. References [3, 28] explored numerous applications as well as the techniques used for managing the big data of smart grids. Reference [70] uses ARL-based learning to draw a pattern between malware and cyberattack activities and draw a rule-based diagram to point infected machines and routes as well as probing the dark net. Similarly, cloud computing has taken the main stage after the recent internet revolution [71]. The increased data and information flow through a wireless medium between intermediate devices in smart grids has also increased the concern of its privacy and security detail. Reference [72] introduces an ARL-based analysis technique for sifting through mined data in order to prepare routines for improving privacy and security along with guaranteed result from the data mining operations.

The recent increase in the large amount of data generation has been problematic for a number of reasons. It takes a toll on services to store and make it accessible; furthermore, to sift through the data, to find relationships, and to make it efficient for the user are a challenge. The data sorting is very important from a number of views like market investment to national security concerns. ARL helps to narrow down the study criteria to a limited pool of variables making it easy to analyze large smart grid data clusters from the point of view of the concerned. Reference [73] focuses on the discovery of these routines in the big data stacks and a post analysis of these rules to arrange them in a better fashion for a more efficient process. Similarly, [74] addresses the problem by an algorithm to highlight the mined rules by assigning them weights in binary digits. Reference [75] uses a number of ARL-associated trees to improve the efficiency of the mined rules in order to keep up with the rapidly increasing data clusters. References [76, 77] proposed a data mining algorithm based upon MapReduce to sift through

the data and produce a more efficient rule-based tree for decision making. Reference [78] is the implementation of the ARL mining on the data in order to improve the failure rate of products and predict the market variables concerning it by studying the trends across the social networks. Also, with the evolution of the internet, it also offers a number of concerned parties a chance to evaluate the customer psyche. Reference [79] is used to mine according to ARL the activities of the users related to their user's concerning factors like when, where, and what for in order to get a better understanding of the response of the customer community.

Increase in the development of a smarter network in service sectors and usage of internet services in many areas of application have further increased the ease of access and maintenance for a number of complex networks. One such example is the new power networks. Power networks are also very important and help in the distribution of the electricity to domestic and industrial use making them an essential part of modern-day life. A failure can lead to disaster. Reference [80] uses a number of algorithms to rule mine the data from power station differential equations. The mined data projects the values regarding the system helping in their maintenance as well as further development of the networks.

*4.2. Data Mining.* Data mining software permits users to make the analysis of data from different dimensions, summarize it, and categorize the relationships identified. The concept of data mining is gaining popularity in the modern era of information and technology. In the information economy, data is being downloaded, uploaded, and extrapolated. So data mining is the incorporation of mathematical methods and algorithms including classification to extract patterns regarding desired data.

A dynamic power grid not only focuses on energy storage but is also concerned about the value of information [81]. According to IEA (International Energy Agency), out of our total final consumption of energy, 32% is consumed by residential and commercial buildings [82]. So it requires more intelligent strategies for processing and analyzing the big data related to the smart grid and residential and commercial buildings. The data mining technique can be used for categorizing smart data into useful information.

Data mining is also used for a number of purposes in the daily civil services ranging from engineering to finance. In Public Structural Development (PSB), data is collected from various aspects and sensors, providing information such as the structural integrity of various structures forming recognition-based patterns on the statistics and is known as structural health monitoring (SHM). The data does not provide the parameters like acceleration and displacement velocity but actually provides the change in the parameters of the structure; a number of mathematical models compute and provide the output according to [83]. Signal sorting of radar communication is an important factor in modern warfare electronics. Modern radars are highly advanced and provide a number of challenges like using multiple emitters eliminating conventional algorithms and producing a ton

of data. A number of developed sorting methods are discussed in [84] in order to sift through the data.

In finance, data mining plays an important role in operation planning. In large-scale operations, business, routines, and processes, it is very important to decompose itself into smaller multiple units for the multiple objective optimizations of the entity also known as role-based access control (RBAC). It uses data mining techniques to discover rules from user permissions from access lists. Reference [85] uses the said technique to further optimize the entity in discussion using RBAC and edge concentration. On the other hand, customer database and trend understanding is very important for service providers. It involves a large database of customers and their daily activities, practices, and behavioral traits. Reference [86] involves the utilization of the multivariate data collected from ends like phones and services. The process involves receiving data and updates from a large number of devices and nodes, sorting and production of desired characteristics and trends. As technologies continue to improve in use and experience, similarly, Facebook-like applications have attracted large user bases linking the virtual space with the real world. In [87], the authors used data mined from the geosocial networks to understand traits and response of the users to provide better statistical analysis for concerned parties providing peoples opinion regarding decisions.

Tax evasion is a common felony practiced at a large scale. Due to the high data volume, it is impossible to detect such a large amount of tax theft, so the data must be sorted and analyzed and the results extrapolated as [88] used the color network-based model (CNBM) for the construction of a pattern tree providing a link between tax evasion techniques and behavior trends. Similarly, electric power is a basic necessity and very important to the modern-day life sustenance. A large number of energy frauds are committed around the world. Reference [89] introduces a technique involving data mining through the advanced metering infrastructure (AMI) plotting the data to provide a number of plausible suspects without including field inspections by constructing a cluster using homogeneous data and constructing prototypes.

*4.3. Classification Tree Analysis.* Classification tree analyses are used to generate the prediction regarding the membership of cases or objects in multiple classes using one or more predictor variables through the help of categorical measurements. Classification tree analysis is one of the main techniques used in data mining.

A decision tree helps the routine in classifying the best option out of the members and their classes presented in the tree which helps in sifting through a large amount of data. For having accurate classes and objects, the training data provided to the tree must be closely related to the analysis data for a comprehensive decision. Reference [89] involves multiple methods and approaches to improve the accuracy of the sample or the training data using multiple attributes of the data. In big data, sometimes, it is needed to compute numerical and mixed data which has to be made discrete, as many of the convention methods and algorithms

are not suitable for big data computation. Reference [90] involves the development of an algorithm to perform discretization and to be further structured into fragments to contain one data each in each object of the classification tree.

With the emergence of high-speed internet to the masses, cloud services are used across the globe from domestic to industrial use. The number of cloud users has reached a peak above any other service comparable like email and social networks. From storage of personal items to office use, the cloud has replaced a number of services but is also giving rise to such things as security and privacy which increases the data multifold. As the user database is increasing, so is the need of betterment in the current cloud structure and implementation. Reference [91] proposes a number of implementations in order to answer the challenges faced by cloud computation.

Data stream is a constant influx of infinite data continuously in a nonstationary manner. In the stream, such algorithms are placed that are learning so that they can overcome the limitations of time and hardware. Reference [92] involves an algorithm to deal with the issue of

- (1) What and how data to present
- (2) The display of the recent data by manipulating the nodes of the classification tree; [93] uses MapReduce in collaboration with tree analysis to sift through the data

*4.4. Genetic Algorithm (GA).* GA are an adaptive and stage-dependent search algorithm. It is based on the evolutionary ideas of genetics and natural selection process. GA is an intelligent optimization algorithm which uses sifting and sorting of a random search. Genetic algorithms (GAs) are designed in such a way so that they can simulate processes in natural systems necessary for evolution. It is obtained solving both limited and nonlimited optimization sets that grow taking the best characteristics like biological evolution, each time replacing the previous with the next.

GA is used in a number of applications along with big data tools like Hadoop. Hadoop is a cluster which consists of thousands of servers and tens of thousands of CPUs which queue up a number of jobs which require multimode scheduling using software. It is needed to improve their efficiency which has been done in depth in [94, 95] keeping in mind real-time system status.

With the increase in the amount of data-generating sources, a better system of mining the data from multiple sources is required. One such source is the modern communication system, which is complex due to the user base. Reference [96] performs an analysis on the problem through classification and regression analysis by studying the big data clusters to detect anomalies. With introduction of cloud computing, it is needed to efficiently mine data for which the big data cloud is used with the help of a number of tools like Hadoop and MapReduce. So it is important to optimize the work of the routine. Reference [97] does in-depth analysis of the process of optimization. With the introduction of cloud computing and software as a service, a number

of web services continue to increase, resulting in increased business value and routines. Reference [98] is based on an analysis on improving the scheduling criteria using MapReduce.

As optimization techniques are used in multiple fields, for instance, big data is generated in the medical field with the help of the high-res scanning technology available. In [99], genetic analysis is used to propose a distinct classification analysis for a number of variables than build a table or tree which could help to develop values for a number of conditions like diseases and infections.

**4.5. Social Network Analysis (SNA).** SNA actually measures the relationships and flows between groups, people, organizations, URLs, computers, and other connected information/knowledge entities.

Nowadays, the internet data has increased multifold and is changing at an alarming rate. The analyses of mined data with algorithms and traditional methods are costly for the large amount of data. So [100] performs the big data tools to map a tree of the traffic nodes on the internet regarding social websites. The internet with its complexity is a collection of large databases with multiple modes. This ranges from text to pictures, videos, etc. However, there is no sorting process for the multitudes of data mined from the internet. So [101] does depth analysis to provide a better approach to the sorting of the data types in order to make the process of data mining more accurate.

Microblogging sites like Facebook, LinkedIn, and Twitter are very important to modern-day communication and play an important role in the daily lives of a large customer database. References [102, 103] use a number of analysis techniques to separate conversation and posts regarding certain data types and construct a tree about relationships interacting with the desired data which in turn can be used for a number of purposes and by many concerned parties for an advertisement-like process. The mined data is used to construct a tree based on the interest of users and recommending them items using a recommender system based on the user activities. On the other hand, with the increase of the microblogging and social websites, social events have been arranged and invitations are sent out via the internet. People attend and get awareness to it, and it further includes the coverage of the event by the people as well. References [104, 105] make the use of algorithms to plot a tree from the information on these sites to detect events and related social activities. With the growth of the internet, social websites have also increased the number of social event coverages on the internet. These events produce multimode data such as videos and images that can be used by concerned parties. Reference [106] proposes an algorithm for multimode tracking of the event for getting a varied form of data.

With the large number of data appearing on the internet, it is also needed to compute and sort the mined data in order to further maximize the use. Reference [107] uses in-depth analysis to compute and arrange the data to produce and maximize the results for use in a number of fields like busi-

ness and media. A lot of reviews and feedback are found on the social websites. Innovation diffusion deals with the response a new product receives in the customer user base like [108] uses a number of algorithms to do in-depth analysis of the data mined through social websites and networks for a concerned product or party.

## 5. Tools of Big Data

The survey covers two universal tools used for the analysis of big data generated by the smart grid, social media, IoT, stock exchange, etc.

- (1) Hadoop
- (2) MapReduce

With all the Vs of big data, conventional means are not enough to tackle the problems and challenges presented by big data and its handling. So for a better analysis method, a number of tools were developed on the techniques mentioned above to work on big data handling. The survey will include Hadoop and its algorithm of MapReduce.

**5.1. Hadoop and Its Importance.** Hadoop was developed as an Apache top level project. It was an open-source implementation of frameworks which provided qualities like reliable, scalable, and distributed computing and data storage. It is a flexible and highly available architecture [109]. The following were goals of the Hadoop Apache Project.

- (i) Facilitate the processing and storage of large and rapidly growing data sets, e.g., unstructured and structured data
- (ii) Simple programming models
- (iii) High availability and scalability
- (iv) Use commodity hardware with little redundancy
- (v) Fault can be tolerated
- (vi) Move computation rather than data

In 2003, Hadoop was bought and implemented by Google, and in 2004, the Hadoop MapReduce Algorithm was developed. Hadoop has the following three important features.

- (1) Hadoop is based around analyzing big volume data in large amounts that are further analyzed by breaking it according to one of the analysis techniques like ARL and CTA. One application is the analysis of large data clusters provided by RFID sensors in a large number of applications such as the Geographic Information System and earth observation with the help of ARL and genetic analysis to filter out the required data from the cluster
- (2) Hadoop is also being used in the analysis of large number of servers ranging from cloud servers to app-related services. These servers get a constant

influx of data from a large number of devices like smart phones having a large array of sensors providing a continuous stream of data. In order to sift through the data, ARL is used with Hadoop to develop relationships and links but with the data in the clusters. Furthermore, the analysis of the mobile networks also yields a large amount of mined data which cannot be handled via conventional means. So Hadoop is used to sift through garbage data and get the required relationships

- (3) With such volumes of data, a large amount of text is also generated. CTA is used with Hadoop to analyze the relationships in the text to arrange them

**5.2. MapReduce.** It is important to differentiate between MapReduce and an implementation of MapReduce, in order to fully understand the capabilities of Hadoop MapReduce. It is an implementation of the algorithm maintained and developed by the Hadoop Apache Project. If you take MapReduce as an engine, then it is an efficient engine which takes data as fuel converting it into energy in a quick and efficient manner.

**5.3. Advantage.** The major advantage is that data processing over multiple computing nodes is made easier using MapReduce.

**5.4. Working.** It can be implemented in three stages, namely, map stage, shuffle stage, and reduce stage.

- (i) *Map stage:* the mapper's job is to process the input data and create small chunks of data, and that is stored in the Hadoop file system (HDFS) in the form of a file or directory. Then, line by line, the input file is passed to the mapper function
- (ii) *Reduce stage:* shuffling and reducing both combine to form the reduce stage. The data that came from the mapper is then processed by the reducer. It gives a new version of the output after processing, which will be stored in the HDFS

MapReduce is based around the analysis of the large amount data inputs to make it very applicable on the modern data and communication network of smart power grids. With the complexity of the modern network, it has to be analyzed for anomalies and dark net trenches which target the data. Furthermore, it is used to analyze the network to maintain and upkeep the internet speed. It also analyzes the cloud network relationships with the internet tracking the big data associations with the cloud network. MapReduce is also used in the database analysis to analyze large data clusters of XML, structured query language (SQL) based on CTA or genetic analysis. It helps a lot in financial and administrative sectors, tracing and locating relationships between data. It had already helped a lot in the federal tax audit for tracing culprits and identity theft. It is also used in the power and domestic services from computing power network algorithms of a city to the traffic patterns in certain hours of the city workload.

## 6. Challenges and Open Issues of Big Data

With the constant evolution of the internet and its related data-generating sources, the volume of big data is increasing at an alarming rate making it necessary for the developers and researchers to keep coming up with new means and analyses to handle big data. It also involves the development of new technologies to look after the hardware prospect of big data computation. So out of the multifold challenges, the following were surveyed.

- (i) Volume
- (ii) Data integration, storage, and visualization from multiple sources
- (iii) Data backup
- (iv) Privacy and security
- (v) Confidentiality
- (vi) Energy management
- (vii) Quality

**6.1. Volume.** The volume of the big data in a smart grid is increasing daily. With the increase in the complexity of the data-generating sources, it is impossible for the conventional data manipulation and sources to deal with big data. By entering smart devices into the mix, the big data clusters are also increasing with higher velocity than the previous 5 years of big data. So using ARL and CTA in collaboration with MapReduce, new developments are being done in new protocols [110] to handle the flood of data across the cloud servers. Reference [111] involves the development of new internet protocols for 5G based on the data accumulated from the study of 3G and 4G internet. With the emerging trends, there is a need of a proper big data computing architecture which is proposed in [112] for smart grid analysis. This communication architecture involves resources of data generation, storage, transmission, and analysis of data. Similarly, [29, 111] established development in the analysis of large RFID and sensor array networks.

So, volume will always remain one of the big challenges in big data as any restriction or limitation on increasing the size of the data cannot be made. Proper data compression methods and continuous research and improvement in big data-handling tools and techniques are the only way to tackle this regularly increasing flood of volume.

**6.2. Data Integration, Storage, and Visualization from Multiple Sources.** Conventional data analysis mostly deals with data generated from a single point. For the case of power grids, data is being generated by distributed grid stations in different areas. It is difficult to store, process, correlate, and visualize data from multiple sources at the same time. HDFS is no doubt a reasonable storage file system, but it needs to be tailored when the data is collected in different representations and formats [25].

**6.3. Data Backup.** Maintaining the backup of collected data is important, but it is very challenging to implement. There are always limited resources for storage and processing of data, and data is being generated at unprecedented scales. There is a need for specifying a life cycle of data. Backup data should be discarded from the storage after completion of the life cycle. The data life cycle management system is itself an open issue because it is very difficult to decide which data shall be discarded without defining a standard principle for removal of stored data from the memory [1].

**6.4. Privacy and Security.** The bulk of information flow and advancement in technology have made living easy, but this advancement in the conventional grid system has serious security issues. Ensuring privacy and securing end-to-end communication in big data are a real challenge for researchers. Considering these security threats, [17] discussed some new findings regarding privacy and security of big data. Internet-based protocols and public communication infrastructure are used in the smart grid which is the cause of arising vulnerabilities that are discussed in [8] in detail. ICN (information-centric networking) is also a strong network architecture for smart grid systems with self-security and congestion control enabled. Reference [112] applies the ICN approach on advanced metering infrastructure to tackle the vulnerabilities regarding data security. New protocols are discussed in [111] to protect large data clusters of XML and SQL from cyberattacks and nonassociated data mining, while [113] deals with the new protocol development of cloud computation based around ARL and CTA. There is a two-way communication between the supplier and the consumer in the smart grid network. Bill payments and transactional data generally include confidential information of the customers. This personal information of the consumer is under serious threat and is one of the most important areas that must be monitored and improved on regular basis.

**6.5. Blockchain.** Blockchain technology is considered the most famous technology based on its high-level data transparency and security. This technology helps to meet the system requirements of smart grids effectively. A blockchain comprises a series of blocks that helps to keep the records of the data in different hash functions with the timestamps. This is beneficial as the data cannot be altered or tampered with. Since the data cannot be changed, data manipulation is impossible, thus protecting the data and reducing the chances of cybercriminals attacking the data.

**6.6. Energy Management.** Efficient utilization of energy is among the most focused topics of discussion all over the globe since the 20th century. The increasing demand of devices and computing systems for storage, processing, and transference of big data has also increased the energy consumption. Therefore, a concrete mechanism for power consumption control and management is worthy of importance for a clean environment and economic stability.

**6.7. Quality.** The quality of the data mined from large data clusters is a crucial factor in a number of applications of

big data analysis. With the ever-increasing data volume and variety, it is necessary to develop algorithms to highlight the relationships between large data clusters. In [114], a variety of data clusters are investigated to improve the mining efficiency of ARL- and SNA-based network algorithms and are applicable in the e-commerce industry. CTA-based decision-making data mining from RFID networks with more accuracy has been discovered in [111].

Data generated from multiple sources, real-time processing, storage, and management of bulky data sets in different modalities and representation, and real modelling are some of the important reasons that restrict giving a fix or one-time solution plan for implementation of big data analytical systems [16]. For the above-mentioned challenges, various data scientists have suggested different solutions that have been cited in the paper. Despite the evolution of data science, this huge amount of collected data is still prone to real threats like cyberattacks, information leakage, personal privacy, and security threats. This is a vast domain that requires advanced solutions and regular improvements with the evolution of big data technologies.

## 7. Conclusion

Based on empirical data, discussion, and literature, it can be concluded that resource management for big data applications emphasizes effective information utilization and analysis. Communities can use smart grid technology to exchange energy in order to meet demand. This paper also addresses the concept of resource management for smart grid applications. It explains what this contemporary idea is and what its features are. The management of various resources, like memory and processors, has also been explored. In a nutshell, resource management is critical in this era of limited resources. Despite the fact that prior surveys have revealed a number of research gaps, there is still a lack of discussion on big data resource management and its recent problems. Our study not only goes over the tools and techniques used in big data analysis in great depth but also covers over the most recent challenges in this field. The growing volume, as well as security and privacy concerns, is underlined. This study provides a comprehensive overview of big data while also revealing unresolved challenges for researchers in the field.

## Abbreviations

RES:	Renewable energy resources
SG:	Smart grid
ICT:	Information and communication technology
GPS:	Global positioning system
CPS:	Cyberphysical system
NVM:	Nonvolatile memory
ARL:	Association rule learning
GA:	Genetic algorithm
DNA:	Deep network analyzer
ISP:	Internet service provider
IEA:	International Energy Agency
PSB:	Public structure development



SHM: Structural health monitoring  
 RBAC: Role-based access control  
 CNBM: Color network-based model  
 AMI: Advanced metering infrastructure  
 ML: Machine learning  
 SNA: Social network analysis  
 CTA: Classification tree analysis  
 RFID: Radio frequency identification  
 HDFS: Hadoop distributed file system  
 ICN: Information-centric networking  
 SQL: Structured query language.

## Data Availability

All data generated or analyzed during this study are included in this published article.

## Conflicts of Interest

The authors declare no conflict of interest.

## References

- [1] M. Chen, S. Mao, and Y. Liu, "Big data: a survey," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 171–209, 2014.
- [2] Y. He, F. R. Yu, N. Zhao, H. Yin, H. Yao, and R. C. Qiu, "Big data analytics in mobile cellular networks," *IEEE access*, vol. 4, pp. 1985–1996, 2016.
- [3] J. Baek, Q. H. Vu, J. K. Liu, X. Huang, and Y. Xiang, "A secure cloud computing based framework for big data information management of smart grid," *IEEE transactions on cloud computing*, vol. 3, no. 2, pp. 233–244, 2015.
- [4] C. L. Philip Chen and C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: a survey on big data," *Information Sciences*, vol. 275, pp. 314–347, 2014.
- [5] S. Peng, G. Wang, and D. Xie, "Social influence analysis in social networking big data: opportunities and challenges," *IEEE Network*, vol. 31, no. 1, pp. 11–17, 2017.
- [6] X. He, Q. Ai, R. C. Qiu, W. Huang, L. Piao, and H. Liu, "A big data architecture design for smart grids based on random matrix theory," *IEEE Transactions on Smart Grid*, vol. 8, no. 2, pp. 674–686, 2017.
- [7] H. Cai, B. Xu, L. Jiang, and A. V. Vasilakos, "IoT-based big data storage systems in cloud computing: perspectives and challenges," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 75–87, 2017.
- [8] W.-L. Chin, W. Li, and H.-H. Chen, "Energy big data security threats in IoT-based smart grid communications," *IEEE Communications Magazine*, vol. 55, no. 10, pp. 70–75, 2017.
- [9] S. Sagiroglu and D. Sinanc, "Big data: a review," in *2013 International Conference on Collaboration Technologies and Systems (CTS)*, pp. 42–47, San Diego, CA, USA, 2013.
- [10] A. Fahad, N. Alshatri, Z. Tari et al., "A survey of clustering algorithms for big data: taxonomy and empirical analysis," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 3, pp. 267–279, 2014.
- [11] D. Singh and C. K. Reddy, "A survey on platforms for big data analytics," *Journal of Big Data*, vol. 2, no. 1, p. 8, 2015.
- [12] C.-W. Tsai, C.-F. Lai, H.-C. Chao, and A. V. Vasilakos, "Big data analytics: a survey," *Journal of Big Data*, vol. 2, no. 1, p. 21, 2015.
- [13] Y. Zhang, T. Cao, S. Li et al., "Parallel processing systems for big data: a survey," *Proceedings of the IEEE*, vol. 104, no. 11, pp. 2114–2136, 2016.
- [14] S. Yu, M. Liu, W. Dou, X. Liu, and S. Zhou, "Networking for big data: a survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 531–549, 2017.
- [15] Y. Zhang, J. Ren, J. Liu, C. Xu, H. Guo, and Y. Liu, "A survey on emerging computing paradigms for big data," *Chinese Journal of Electronics*, vol. 26, no. 1, pp. 1–12, 2017.
- [16] M. Ghorbanian, S. H. Dolatabadi, and P. Siano, "Big data issues in smart grids: a survey," *IEEE Systems Journal*, vol. 13, no. 4, pp. 4158–4168, 2019.
- [17] J. Hu and A. V. Vasilakos, "Energy big data analytics and security: challenges and opportunities," *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2423–2436, 2016.
- [18] D. C. Marinescu, A. Paya, and J. P. Morrison, "A cloud reservation system for big data applications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 3, pp. 606–618, 2017.
- [19] Y. Zhang, M. Qiu, C.-W. Tsai, M. M. Hassan, and A. Alamri, "Health-CPS: healthcare cyber-physical system assisted by cloud and big data," *IEEE Systems Journal*, vol. 11, no. 1, pp. 88–95, 2017.
- [20] P. K. Sahoo, S. K. Mohapatra, and S.-L. Wu, "Analyzing healthcare big data with prediction for future health condition," *IEEE Access*, vol. 4, pp. 9786–9799, 2016.
- [21] H. Attaullah, T. Kanwal, A. Anjum et al., "Fuzzy logic-based privacy-aware dynamic release of IoT-enabled healthcare data," *IEEE Internet of Things Journal*, 2021.
- [22] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. Ullah Khan, "The rise of "big data" on cloud computing: review and open research issues," *Information Systems*, vol. 47, pp. 98–115, 2015.
- [23] R. Kune, P. K. Konugurthi, A. Agarwal, R. R. Chillarige, and R. Buyya, "The anatomy of big data computing," *Software: Practice and Experience*, vol. 46, no. 1, pp. 79–105, 2016.
- [24] J. B. Ekanayake, N. Jenkins, K. Liyanage, J. Wu, and A. Yokoyama, *Smart Grid: Technology and Applications*, John Wiley & Sons, 2012.
- [25] C. Tu, X. He, Z. Shuai, and F. Jiang, "Big data issues in smart grid - a review," *Renewable and Sustainable Energy Reviews*, vol. 79, pp. 1099–1107, 2017.
- [26] H. Jiang, K. Wang, Y. Wang, M. Gao, and Y. Zhang, "Energy big data: a survey," *IEEE Access*, vol. 4, pp. 3844–3861, 2016.
- [27] H. Zhang, G. Chen, B. C. Ooi, K.-L. Tan, and M. Zhang, "In-memory big data management and processing: a survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 7, pp. 1920–1948, 2015.
- [28] M. Jaradat, M. Jarrah, A. Bousseham, Y. Jararweh, and M. Al-Ayyoub, "The internet of energy: smart sensor networks and big data management for smart grid," *Procedia Computer Science*, vol. 56, pp. 592–597, 2015.
- [29] Y. Mengke, Z. Xiaoguang, Z. Jianqiu, and X. Jianjian, "Challenges and solutions of information security issues in the age of big data," *China Communications*, vol. 13, no. 3, pp. 193–202, 2016.
- [30] A. Cuzzocrea, "Privacy and security of big data: current challenges and future research perspectives," in *Proceedings of the First International Workshop on Privacy and Security of Big Data - PSBD '14*, pp. 45–47, 2014.

- [31] E. Hossain, I. Khan, F. Un-Noor, S. S. Sikander, and M. S. H. Sunny, "Application of big data and machine learning in smart grid, and associated security concerns: a review," *IEEE Access*, vol. 7, pp. 13960–13988, 2019.
- [32] A. Siddiqua, I. A. T. Hashem, I. Yaqoob et al., "A survey of big data management: taxonomy and state-of-the-art," *Journal of Network and Computer Applications*, vol. 71, pp. 151–166, 2016.
- [33] P. D. Diamantoulakis, V. M. Kapinas, and G. K. Karagiannidis, "Big data analytics for dynamic energy management in smart grids," *Big Data Research*, vol. 2, no. 3, pp. 94–101, 2015.
- [34] B. P. Bhattarai, S. Paudyal, Y. Luo et al., "Big data analytics in smart grids: state-of-the-art, challenges, opportunities, and future directions," *IET Smart Grid*, vol. 2, no. 2, pp. 141–154, 2019.
- [35] B. Speer, M. Miller, W. Schaffer et al., *The role of smart grids in integrating renewable energy*, Tech. Rep., National Renewable Energy Laboratory (NREL), Golden, CO (United States), 2015.
- [36] S. Bruno, S. Lamonaca, M. La Scala, G. Rotondo, and U. Stecchi, "Load control through smart-metering on distribution networks," in *2009 IEEE Bucharest PowerTech*, pp. 1–8, Bucharest, Romania, 2009.
- [37] V. C. Gungor, D. Sahin, T. Kocak et al., "Smart grid technologies: communication technologies and standards," *IEEE transactions on Industrial informatics*, vol. 7, no. 4, pp. 529–539, 2011.
- [38] G. A. Boyd and J. X. Pang, "Estimating the linkage between energy efficiency and productivity," *Energy Policy*, vol. 28, no. 5, pp. 289–296, 2000.
- [39] M. Batty, "Big data, smart cities and city planning," *Dialogues in Human Geography*, vol. 3, no. 3, pp. 274–279, 2013.
- [40] M. Batty, *Smart cities, big data*, 2012.
- [41] R. Moghe, F. C. Lambert, and D. Divan, "Smart stick-on sensors for the smart grid," *IEEE Transactions on Smart Grid*, vol. 3, no. 1, pp. 241–252, 2012.
- [42] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart grid — the new and improved power grid: a survey," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 944–980, 2012.
- [43] A. Arif, M. Al-Hussain, N. Al-Mutairi, E. Al-Ammar, Y. Khan, and N. Malik, "Experimental study and design of smart energy meter for the smart grid," in *2013 International Renewable and Sustainable Energy Conference (IRSEC)*, pp. 515–520, Ouarzazate, Morocco, 2013.
- [44] F. Clarizia, D. Gallo, C. Landi, M. Luiso, and R. Rinaldi, "Smart meter systems for smart grid management," in *2016 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, pp. 1–6, Taipei, Taiwan, 2016.
- [45] C. De Capua, G. Lipari, M. Lugara, and R. Morello, "A smart energy meter for power grids," in *Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, pp. 878–883, Montevideo, Uruguay, 2014.
- [46] J. Zheng, D. W. Gao, and L. Lin, "Smart meters in smart grid: an overview," in *2013 IEEE Green Technologies Conference (GreenTech)*, pp. 57–64, Denver, CO, USA, 2013.
- [47] F. Qayyum, M. Naeem, A. S. Khwaja, A. Anpalagan, L. Guan, and B. Venkatesh, "Appliance scheduling optimization in smart home networks," *IEEE Access*, vol. 3, pp. 2176–2190, 2015.
- [48] Z. Ahmed, A. Farooqi, and R. M. Navid-ur Rehman, "Implementation of smart system based on smart grid smart meter and smart appliances," in *Iranian Conference on Smart Grids*, pp. 1–4, Tehran, Iran, 2012.
- [49] A. M. Carreiro, C. H. Antunes, and H. M. Jorge, "Energy smart house architecture for a smart grid," in *2012 IEEE International Symposium on Sustainable Systems and Technology (ISSST)*, p. 1, May 2012.
- [50] G. S. Aleena, P. Sivraj, and K. K. Sasi, "Resource management on smart micro grid by embedded networking," *Procedia Technology*, vol. 21, pp. 468–473, 2015.
- [51] G. Suciuc, V. A. Poenaru, C. G. Cernat, G. Todoran, and T. L. Militaru, "ERP and e-business application deployment in open source distributed cloud systems," in *The Eleventh International Conference on Informatics in Economy IE*, pp. 12–17, 2012.
- [52] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 97–107, 2014.
- [53] R. Buyya, R. N. Calheiros, and A. V. Dastjerdi, *Big Data: Principles and Paradigms*, Morgan Kaufmann, 2016.
- [54] F. Pop, J. K. Lodziej, and B. Di Martino, *Resource Management for Big Data Platforms*, Springer, 2016.
- [55] C. Yao, C. Yang, and Z. Xiong, "Energy-saving predictive resource planning and allocation," *IEEE Transactions on Communications*, vol. 64, no. 12, pp. 5078–5095, 2016.
- [56] H. Won, M. C. Nguyen, M.-S. Gil, and Y.-S. Moon, "Advanced resource management with access control for multitenant Hadoop," *Journal of Communications and Networks*, vol. 17, no. 6, pp. 592–601, 2015.
- [57] A. Braganza, L. Brooks, D. Nepelski, M. Ali, and R. Moro, "Resource management in big data initiatives: processes and dynamic capabilities," *Journal of Business Research*, vol. 70, pp. 328–337, 2017.
- [58] D. Cao, P. Liu, W. Cui, Y. Zhong, and B. An, "Cluster as a service: a resource sharing approach for private cloud," *Tsinghua Science and Technology*, vol. 21, no. 6, pp. 610–619, 2016.
- [59] D. Wu, L. Zhu, X. Xu, S. Sakr, D. Sun, and Q. Lu, "Building pipelines for heterogeneous execution environments for big data processing," *IEEE Software*, vol. 33, no. 2, pp. 60–67, 2016.
- [60] M. V. Moreno, F. Terroso-Saenz, A. Gonzalez-Vidal et al., "Applicability of big data techniques to smart cities deployments," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 800–809, 2017.
- [61] R. B. Hagmann, "A crash recovery scheme for a memory-resident database system," *IEEE Transactions on Computers*, vol. 35, no. 9, pp. 839–843, 1986.
- [62] M. Zaharia, M. Chowdhury, T. Das et al., "Resilient distributed datasets: a fault-tolerant abstraction for in memory cluster computing," in *9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI})*, pp. 15–28, 2012.
- [63] D. Loghin, B. M. Tudor, H. Zhang, B. C. Ooi, and Y. M. Teo, "A performance study of big data on small nodes," *Proceedings of the VLDB Endowment*, vol. 8, no. 7, pp. 762–773, 2015.
- [64] N. Davies, A. Friday, S. Clinch et al., "Security and privacy implications of pervasive memory augmentation," *IEEE Pervasive Computing*, vol. 14, no. 1, pp. 44–53, 2015.

- [65] C. Song, Z. Qu, N. Blumm, and A.-L. Barabasi, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.
- [66] H. Abou-Zeid and H. S. Hassanein, "Predictive green wireless access: exploiting mobility and application information," *IEEE Wireless Communications*, vol. 20, no. 5, pp. 92–99, 2013.
- [67] A. Nadembega, A. Hafid, and T. Taleb, "Mobility-prediction-aware bandwidth reservation scheme for mobile networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 6, pp. 2561–2576, 2015.
- [68] B. Mokhtar and M. Eltoweissy, "Big data and semantics management system for computer networks," *Ad Hoc Networks*, vol. 57, pp. 32–51, 2017.
- [69] K. Yang, R. Liu, Y. Sun, J. Yang, and X. Chen, "Deep network analyzer (DNA): a big data analytics platform for cellular networks," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2019–2027, 2017.
- [70] T. Ban, M. Eto, S. Guo, D. Inoue, K. Nakao, and R. Huang, "A study on association rule mining of darknet big data," in *2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, Killarney, Ireland, 2015.
- [71] A. Mohamed, M. Hamdan, S. Khan et al., "Software-defined networks for resource allocation in cloud computing: a survey," *Computer Networks*, vol. 195, article 108151, 2021.
- [72] C. Huang and R. Lu, "EFPA: efficient and flexible privacy-preserving mining of association rule in cloud," in *2015 IEEE/CIC International Conference on Communications in China (ICCC)*, pp. 1–6, Shenzhen, China, 2015.
- [73] M. Dai and Y.-L. Huang, "Organizing the discovered association rules based on general-specific (GS) hierarchical patterns," in *2005 International Conference on Machine Learning and Cybernetics*, pp. 2206–2211, Guangzhou, China, 2005.
- [74] W. S. Seol, H. W. Jeong, B. Lee, and H. Y. Youn, "Reduction of association rules for big data sets in socially-aware computing," in *2013 IEEE 16th International Conference on Computational Science and Engineering*, pp. 949–956, Sydney, NSW, Australia, 2013.
- [75] X. Zhou and Y. Huang, "An improved parallel association rules algorithm based on MapReduce framework for big data," in *2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pp. 284–288, Xiamen, China, 2014.
- [76] P. Ducange, F. Marcelloni, and A. Segatori, "A MapReduce-based fuzzy associative classifier for big data," in *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1–8, Istanbul, Turkey, 2015.
- [77] H.-Y. Chang, Z.-H. Hong, T.-L. Lin, W.-K. Chang, and Y.-Y. Lin, "IPARBC: an improved parallel association rule based on MapReduce framework," in *2016 International Conference on Networking and Network Applications (NaNA)*, pp. 370–374, Hakodate, Japan, 2016.
- [78] Z. He, Y. He, and L. Wang, "Root causes identification approach based on association rule mining for product infant failure," in *2015 8th International Symposium on Computational Intelligence and Design (ISCID)*, pp. 624–628, Hangzhou, China, 2015.
- [79] C. Zhou, H. Jiang, Y. Chen, L. Wu, and S. Yi, "User interest acquisition by adding home and work related contexts on mobile big data analysis," in *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 201–206, San Francisco, CA, USA, 2016.
- [80] G. Sheng, H. Hou, X. Jiang, and Y. Chen, "A novel association rule mining method of big data for power transformers state parameters based on probabilistic graph model," *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 695–702, 2018.
- [81] K. K. Zame, C. A. Brehm, A. T. Nitica, C. L. Richard, and G. D. Schweitzer III, "Smart grid and energy storage: policy recommendations," *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 1646–1654, 2018.
- [82] M. Molina-Solana, M. Ros, M. D. Ruiz, J. Gomez-Romero, and M. J. Martin-Bautista, "Data science for building energy management: a review," *Renewable and Sustainable Energy Reviews*, vol. 70, pp. 598–609, 2017.
- [83] X. Li, W. Yu, and S. Villegas, "Structural health monitoring of building structures with online data mining methods," *IEEE Systems Journal*, vol. 10, no. 3, pp. 1291–1300, 2016.
- [84] J. Wan, P. Nan, Q. Guo, and Q. Wang, "Multi-mode radar signal sorting by means of spatial data mining," *Journal of Communications and Networks*, vol. 18, no. 5, pp. 725–734, 2016.
- [85] L. Dong, K. Wu, and G. Tang, "A data-centric approach to quality estimation of role mining results," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2678–2692, 2016.
- [86] R. Rawassizadeh, E. Momeni, C. Dobbins, J. Gharibshah, and M. Pazzani, "Scalable daily human behavioral pattern mining from multivariate temporal data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 11, pp. 3098–3112, 2016.
- [87] J.-D. Zhang and C.-Y. Chow, "CRATS: an LDA-based model for jointly mining latent communities, regions, activities, topics, and sentiments from geosocial network data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 11, pp. 2895–2909, 2016.
- [88] F. Tian, T. Lan, K.-M. Chao et al., "Mining suspicious tax evasion groups in big data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 10, pp. 2651–2664, 2016.
- [89] S. Y. Han, J. No, J.-H. Shin, and Y. Joo, "Conditional abnormality detection based on AMI data mining," *IET Generation, Transmission & Distribution*, vol. 10, no. 12, pp. 3010–3016, 2016.
- [90] Y. Zhang and Y.-M. Cheung, "Discretizing numerical attributes in decision tree for big data analysis," in *2014 IEEE International Conference on Data Mining Workshop*, pp. 1150–1157, Shenzhen, China, 2014.
- [91] C. Liu, R. Ranjan, C. Yang, X. Zhang, L. Wang, and J. Chen, "MUR-DPA: top-down levelled multi-replica Merkle hash tree based secure public auditing for dynamic big data storage on cloud," *IEEE Transactions on Computers*, vol. 64, no. 9, pp. 2609–2622, 2015.
- [92] N.-Q. Doan, M. Ghesmoune, H. Azzag, and M. Lebbah, "Growing hierarchical trees for data stream clustering and visualization," in *2015 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, Killarney, Ireland, 2015.
- [93] F. Yuan, F. Lian, X. Xu, and Z. Ji, "Decision tree algorithm optimization research based on MapReduce," in *2015 6th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pp. 1010–1013, Beijing, China, 2015.
- [94] X. Huang, H. Zhou, and W. Wu, "Hadoop job scheduling based on mixed ant-genetic algorithm," in *2015 International*

- Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, pp. 226–229, Xi'an, China, 2015.
- [95] Q. Lu, S. Li, and W. Zhang, "Genetic algorithm based job scheduling for big data analytics," in *2015 International Conference on Identification, Information, and Knowledge in the Internet of Things (IIKI)*, pp. 33–38, 2015.
- [96] M. De Sanctis, I. Bisio, and G. Araniti, "Data mining algorithms for communication networks control: concepts, survey and guidelines," *IEEE Network*, vol. 30, no. 1, pp. 24–29, 2016.
- [97] R. Kune, P. K. Konugurthi, A. Agarwal, R. R. Chillarige, and R. Buyya, "Genetic algorithm based data-aware group scheduling for big data clouds," in *2014 IEEE/ACM International Symposium on Big Data Computing*, pp. 96–104, London, UK, 2014.
- [98] Y. Zhang, Z. Jing, and Y. Zhang, "MR-IDPSO: a novel algorithm for large-scale dynamic service composition," *Tsinghua Science and Technology*, vol. 20, no. 6, pp. 602–612, 2015.
- [99] M. O. Ulfarsson, F. Pálsson, J. Sigurdsson, and J. R. Sveinsson, "Classification of big data with application to imaging genetics," *Proceedings of the IEEE*, vol. 104, no. 11, pp. 2137–2154, 2016.
- [100] P. Agarwal, R. Ahmed, and T. Ahmad, "Identification and ranking of key persons in a social networking website using Hadoop & big data analytics," in *Proceedings of the International Conference on Advances in Information Communication Technology & Computing*, p. 65, 2016.
- [101] R. Vatrappu, R. R. Mukkamala, A. Hussain, and B. Flesch, "Social set analysis: a set theoretical approach to big data analytics," *IEEE Access*, vol. 4, pp. 2542–2571, 2016.
- [102] G. Ghosh, S. Banerjee, and N. Y. Yen, "State transition in communication under social network: an analysis using fuzzy logic and density based clustering towards big data paradigm," *Future Generation Computer Systems*, vol. 65, pp. 207–220, 2016.
- [103] M. Narayanan and A. K. Cherukuri, "A study and analysis of recommendation systems for location-based social network (LBSN) with big data," *IIMB Management Review*, vol. 28, no. 1, pp. 25–30, 2016.
- [104] M. Zaharieva, M. Del Fabro, and M. Zeppelzauer, "Cross-platform social event detection," *IEEE MultiMedia*, vol. 22, no. 3, pp. 14–25, 2015.
- [105] O. Liu, K. Man, W. Chong, and C. Chan, "Social network analysis using big data," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, pp. 6-7, 2016.
- [106] S. Qian, T. Zhang, C. Xu, and J. Shao, "Multi-modal event topic model for social event analysis," *IEEE Transactions on Multimedia*, vol. 18, no. 2, pp. 233–246, 2016.
- [107] E. Cambria, N. Howard, Y. Xia, and T.-S. Chua, "Computational intelligence for big social data analysis [guest editorial]," *IEEE Computational Intelligence Magazine*, vol. 11, no. 3, pp. 8-9, 2016.
- [108] J. Zhang, F. Xia, Z. Ning et al., "A hybrid mechanism for innovation diffusion in social networks," *IEEE Access*, vol. 4, pp. 5408–5416, 2016.
- [109] J. Liu, F. Liu, and N. Ansari, "Monitoring and analyzing big traffic data of a large-scale cellular network with Hadoop," *IEEE Network*, vol. 28, no. 4, pp. 32–39, 2014.
- [110] A. Imran, A. Zoha, and A. Abu-Dayya, "Challenges in 5G: how to empower son with big data for enabling 5G," *IEEE Network*, vol. 28, no. 6, pp. 27–33, 2014.
- [111] J. Wu, S. Guo, J. Li, and D. Zeng, "Big data meet green challenges: big data toward green applications," *IEEE Systems Journal*, vol. 10, no. 3, pp. 888–900, 2016.
- [112] K. Yu, M. Arifuzzaman, Z. Wen, D. Zhang, and T. Sato, "A key management scheme for secure communications of information centric advanced metering infrastructure in smart grid," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 8, pp. 2072–2085, 2015.
- [113] C. Hewitt, "Orgs for scalable, robust, privacy-friendly client cloud computing," *IEEE internet computing*, vol. 12, no. 5, pp. 96–99, 2008.
- [114] Z. Han, M. Bennis, D. Wang, T. Kwon, and S. Cui, "Special issue on big data networking-challenges and applications," *Journal of Communications and Networks*, vol. 17, no. 6, pp. 545–548, 2015.