



OPEN Kademia hash snow ablation resource optimized stride scheduling for mobile computing services in healthcare sector

Nithya Rekha Sivakumar

Healthcare plays an essential role in every individual's life. Smart healthcare employs a new generation of information technologies like Internet of Things (IoT), Mobile Computing (MC), and Artificial Intelligence (AI) to transform the conventional medical system into a modernized one. Many researchers have reached better healthcare solutions by ensuring robust and versatile accessibility to the people. Healthcare applications on mobile devices exchange data through communication interfaces between patients and healthcare service providers. However, every patient's data possess a significant amount of computing resources like Central Processing Unit (CPU), network bandwidth, and memory, and hence cannot be processed and validated at the same time. In order to overcome resource limitations and make scheduling efficient, mobile devices contain patient data that is integrated with the scheduling process to provide the resource efficient mobile computing services in the healthcare sector. Kademia HashSnow Ablation Resource Optimized Stride Scheduling (KHSAROSS) is proposed in mobile computing. The novelty of the proposed KHSAROSS method is designed to build mobile phone based remote healthcare monitoring to identify the resource optimized virtual machine and balance the overloaded virtual machine. The key advantage of the proposed KHSAROSS method is to enhance the mobile computing services in the healthcare sector while increasing makespan. The KHSAROSS method involves three distinct processes, namely the patient data collection task, optimization and scheduling. Sensors with a data acquisition unit are used to acquire patients' physiological states like temperature, heart rate, and Electroencephalography (EEG) data to perform mobile phone based remote healthcare monitoring. The Kademia Hash Function here generates a hash value for each patient's collected data. Following this, Snow Ablation Optimization is carried out to identify resource optimized (i.e., CPU, network bandwidth, and memory) virtual machines (i.e., healthcare service providers) for performing scheduling of stored patient data. Finally, Stride Scheduling is used to balance overloaded virtual machines with less loaded virtual machines. This, in turn, ensures efficient resource allocation and scheduling in mobile computing. The effectiveness of the proposed and existing methods is assessed using metrics for scheduling accuracy, scheduling time, throughput and makespan.

Keywords Internet of things, Mobile computing, Artificial intelligence, Kademia hash function, Snow ablation optimization, Stride scheduling

Over the past few years, with the full amalgamation and application of state-of-the-art technologies, to name a few, artificial intelligence and cloud computing, the Internet of Healthcare Things (IoHT), a paramount application area of IoT technology in the healthcare industry, has progressively shown strong influence and vitality. Nevertheless, there are numerous resource concerns, to name a few being battery life, bandwidth, and storage space issues unique to mobile devices. As a result, the service quality is said to be heavily influenced by these issues.

A computational task offloading method called Online Offloading based on the Deep Reinforcement Learning (OO-DRL) in smart healthcare networks with non-divisible features and a tradeoff between latency and energy consumption to converge the necessities of high-reliability services was proposed in¹. The OO-DRL

Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University (PNU), P.O. Box 84428, Riyadh 11671, Saudi Arabia. email: NRRaveendiran@pnu.edu.sa

offloading algorithm via parallel deep neural networks and a weight-sensitive experience replay mechanism was designed to minimize real-time task processing latency and energy consumption extensively. However, the accuracy and time involved in the process of scheduling were not focused. To concentrate on this aspect, a Snow Ablation-based Resource Optimization and Stride Scheduling model is designed to improve the overall accuracy and reduce time.

Owing to constrained resources, mobile devices frequently make use of cloud computing offloading characteristics to perform more complicated tasks. In², a non-dominated multi-objective strategy taking into consideration the Harris Hawks Optimization (HHO), referred to as the Hybrid Multi-objective Harris Hawks Optimization (HMHHO), was proposed with the intent of handling issues concerning constrained resources. With this type of design, the HMHHO method not only accomplished jobs quickly but also minimized energy significantly. In spite of improvements in energy, the latency and makespan involved for mobile computing services in the healthcare sector have not been addressed. To concentrate on these issues, the Kademia Hash Function is employed, which generates a hash value for each patient's collected data, therefore considering overall latency and makespan.

Aging populations, the mutation of critical illnesses to chronic diseases, and economic coherences related to the transfer of patient management to the home, instead of specialized institutions, are steering the application of internet-enabled home care mechanisms and specifically remote health monitoring along with mobile health solutions. A holistic exploration of the integration of Artificial Intelligence (AI) into healthcare, concentrating on its transformative inferences and challenges, was investigated in³. The employment of machine learning techniques in health care is swiftly improving. Nevertheless, the procedures that aid these tools in deployment employing machine learning operations are still in the infant stage. In⁴, not only holistic fusions of prevailing in the field of healthcare but also the gaps were identified, and insights for adoption in clinical practice were presented.

A transient overview of AI along with its contributions and real-world applications in the healthcare domain was investigated in⁵. Also, its influence on medical education, along with the ethical issues of its integration into the medical field was discussed in depth. However, issues concerning user capacity, network speed, price, and resources involved were not focused. To concentrate on these aspects, several research issues for 5G-based smart healthcare were discussed in⁶. Nevertheless, any downtime, in the case of critical health services, could result in patient health issues and also cause mortality. In⁷, an interdisciplinary mechanism integrating stochastic methods with optimization techniques was presented with the intent of analyzing how failures influence e-health monitoring system accessibility.

With the evolution of IoT, E-healthcare applications are making an appearance and reaping popularity. These E-healthcare applications necessitate low latency and rigorous computation or storage resources. But, IoT terminals possess constrained resources that are found to be inadequate to accomplish these applications. A novel framework for an Intelligent and Interactive Healthcare System combining cloud computing techniques and concentrating on Speech Recognition was proposed in⁸. Here, the main emphasized on predicting physical or psychological discomfort via a Hidden Markov model with improved accuracy. However, with high user mobility precision, one of the important factors in resource-constrained sensors has said to be compromised extensively. Here Markov decision⁹ process was employed in acquiring optimal service migration policies extensively.

A review of machine learning techniques for healthcare IoT was investigated in¹⁰. Yet another systematic review on optimization techniques to focus on the scheduling aspects employing machine learning and data-driven approaches was designed in¹¹. A deep dive into healthcare consisting of seamless connectivity, network scalability, and data privacy was analyzed alongside methods presented in¹². Nevertheless, traditional cloud-based methods pose issues owing to the increase in volume and time-sensitive data. In¹³, a Regional Computing (RC) framework for the management of healthcare bid data was presented in detail.

One of the main advancements used in recent years is the Internet of Medical Things (IoMT) in the cloud, which permits users to access cloud services over the Internet in a remote fashion. A scheduling mechanism that addresses the Quality of Service parameters with minimal energy consumption is said to be the need of the hour. An overview of the integration of IoMT and cloud computing frameworks employing Extended Water Wave Optimization (EWWO) task scheduling in the IoMT for healthcare applications was presented in¹⁴. Employing this extended algorithm resulted in the improvement of execution time with improved resource utilization. Yet another method employing the augmented moth flame optimization technique to focus on the task offloading with minimal energy consumption was proposed in¹⁵.

Problem definition

Many scheduling process is developed in the smart healthcare sector. Here, two existing methods are used in the proposed method performance evaluation in smart healthcare networks. At first, Deep Reinforcement Learning based computational task offloading failed to consider the involved rate of accuracy and time during the scheduling process. Another one is the Hybrid Multi-objective Optimization technique that only handles constrained resources problems. But, it failed to consider the latency and makespan involved in the healthcare sector via mobile computing services.

To concern these issues, with the assist of separate optimization and scheduling technique, the proposed KHSAROSS method design Snow Ablation-based Resource Optimization and Stride Scheduling model focused. Followed by, Kademia Hash Function is applied to produce hash value for each patient collected data to focus the latency and makespan of overall performance.

A key objective is to construct a mobile phone based remote healthcare monitoring to identify the resource optimized virtual machine and balance the overloaded virtual machine to the less loaded virtual machine. In order to achieve this, a systematic analysis of KHSAROSS mobile computing services is performed in the CC

environment. Moreover, we analyze the requirements and consequences of utilizing quality of service in terms of scheduling accuracy, scheduling time, makespan, throughput, and latency in the design of the proposed KHSAROSS method for mobile computing services in healthcare.

Contributions of the work

As presented in the above section, there are several works in the literature in regards to mobile computing healthcare services in the CC environment. This research proposes the following novel contributions.

- To overcome resource limitations and make scheduling efficient, a method using mobile devices that integrates patient data that integrated with the scheduling process called KHSAROSS is developed based on three major processes, namely data collection task, optimization, and scheduling.
- First, the Kademia Hash Function-based data collection model is proposed in the KHSAROSS method for generating node ID, initialization, and fine-tuning of the routing table, network joining, and finally storing the data in a cloud server for the corresponding cloud user's requested tasks from the MMASH Datasets. The generation of node ID for each participant employing Division Hashing function and network joining process via distributed hash table is performed in a greedy fashion to minimize makespan and latency.
- To improve scheduling accuracy, scheduling time along with throughput improvement, the Snow Ablation-based Resource Optimization and Stride Scheduling model. Here, optimization and scheduling is performed separately via a fitness function (minimization of load and maximization of resource utilization). The proposed P Snow Ablation-based Resource Optimization and Stride Scheduling model using the QoS factors aids in improving the overall throughput in an efficient manner.
- Finally, a comprehensive experimental assessment is carried out with different types of performance parameters to analyze and validate the performance improvement of the proposed KHSAROSS method over conventional methods. The proposed KHSAROSS method is implemented in Python high-level general-purpose programming language, using hardware requirements such as 8 GB RAM, a 2 TB hard drive, and a CORE i7 CPU.

Organization of the work

The rest of this work is structured as follows. Section "[Related works](#)" discusses different mobile computing services in the healthcare domain with an overview of related work and narrates their indispensable features. Section "[Methodology](#)" describes the proposed KHSAROSS method and presents strategies for algorithms. Section "[Results and discussion](#)" provides a detailed experimental setup, followed by simulation results and comparative comparisons with the state-of-the-art methods. Finally, Sect. "[Conclusion](#)" provides the conclusions of this research.

Related works

Healthcare systems are swiftly progressing owing to the application of sophisticated techniques like, ML, IoT, a portion of AI concentrated on enabling machines to learn from data and enhance their performance to enable faster processing. These evolutions are said to be fine tuning healthcare services being managed and delivered by patients and providers alike.

The role and influence of new techniques being applied to healthcare systems were investigated in¹⁶. Constrained probabilities of mobile devices, in terms of processing, time, energy consumption, and storage, cause issues concerning energy optimization and time management during the processing of tasks on mobile phones. These issues increase the need for offloading to mobile devices. A novel task scheduling method to minimize energy consumption and time using an energy-efficient dynamic decision-based method was proposed in¹⁷. With this type of design, not only was the task scheduling time but also the energy consumption was reduced significantly. Yet another smart and secure IoT framework mobile healthcare sector was investigated in¹⁸.

The consolidation of healthcare and technological innovation has seen mushroom improvement inspired by the ceaseless pursuit for effectiveness, accuracy and treatment personally. As this development evolves, the drastic issue of controlling enormous amount of data and computational necessities becomes more acute.

The method¹⁹ aspires to focus on the real-time healthcare demands employing dynamic offloading techniques using Digital Twins (DT) and social health determinants to improve healthcare interferences. Yet another method employing AI for Mobile Cloud Computing (MCC) in healthcare was presented in²⁰. As the evolution and application of mobile cloud computing techniques in healthcare, novel optimization mechanisms have been analyzed and validated to assist mobile cloud healthcare services in being deployed in a more efficient fashion.

In²¹, a survey of mechanisms demonstrated in various healthcare applications was investigated in detail. Yet another qualitative examination was performed in²² for a community health worker to analyze contextual factors involved in India. Mobile health interferences could have advantageous influences on health care delivery processes. The IoT is indispensable in innovative applications, to name a few, being smart homes, healthcare, smart cities, and so on. IoT applications are specifically advantageous for imparting healthcare owing to the reason that they enable real-time remote patient monitoring to boost quality of life.

A systematic review of the efficiency of mobile health technologies for the corresponding service delivery process was investigated in²³. Yet another investigation of improving the quality of life concerning the healthcare monitoring system was designed in²⁴. A case analysis of implementing survival among Indians associated with mobile health technology was investigated in²⁵. A method to focus on the important health metrics on real-time cloud computing employing a web-based interface was designed in²⁶.

A review of the CC environment in healthcare systems and its influence on bioinformatics research was investigated in²⁷. Yet another mobile computing services framework to focus on the accuracy and precision aspects involved in an enhanced healthcare system was designed in²⁸. A holistic review of data storage

mechanisms for cloud computing in healthcare was presented in²⁹. A decentralized architecture was designed in³⁰ for effective content delivery in use crowd-sourced video streaming using Kademia information-sharing and exchange method. But, it failed to evaluate the real-world events' performance effectiveness when using a larger number of nodes. Also, decentralized multi-view stream editing implementation was challenging. To address this, Kadabra, a decentralized protocol was designed in³¹ to learn the routing tables in Kademia for increasing the speed of lookups. Here, an integration of a fully decentralized and non-cooperative algorithm was termed as Kadabra that adapts to a heterogeneity network.

Except for the above works, most other research only failed to focus on either minimizing the makespan or minimizing the latency. Different from the research aforementioned, optimal KHSAROSS aims at minimizing the makespan latency along with the overall throughput improvement. The elaborate description of the KHSAROSS method is provided in the following sections.

Methodology

As far as healthcare is concerned, mobile computing is reconstructing the process of medical services, allowing real-time access to patient data, therefore strengthening communication between healthcare providers and smoothing the path of telemedicine. However, strategic management of technology resources like data, CPU, network bandwidth, and memory to ensure effective patient care delivery still remains in infancy. In this section, a method called KHSAROSS in mobile computing is designed to focus on the resource optimization strategy therefore resulting in the improvement of efficient resource allocation and scheduling in mobile computing. Figure 1 shows the block diagram of the KHSAROSS method.

Figure 1 represents the Structure of the KHSAROSS method for mobile computing services in the healthcare sector. The KHSAROSS method is split into three processes. Initially, the patient data collection task is

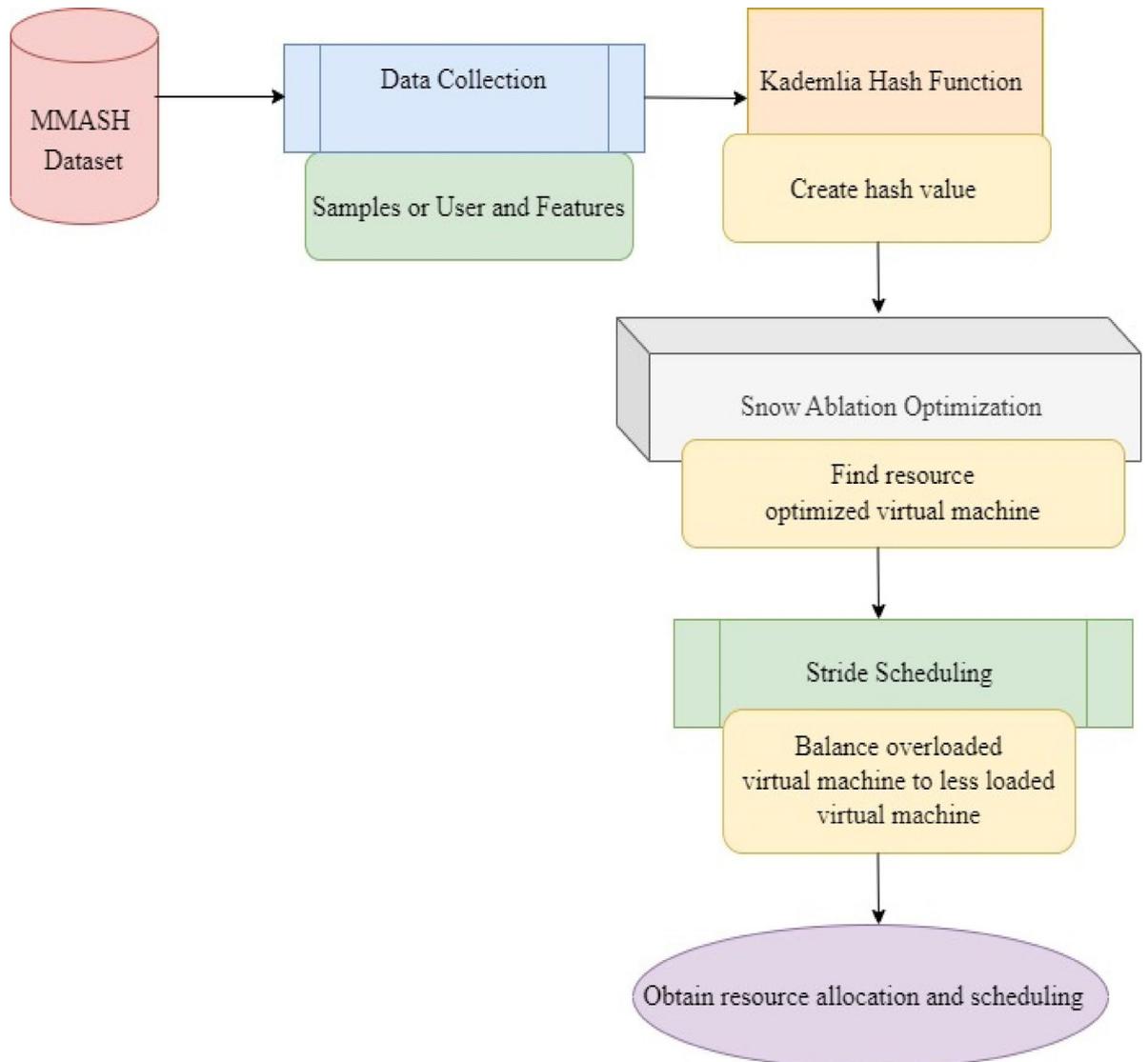


Fig. 1. Structure of KHSAROSS for mobile computing services in healthcare sector.

performed employing the MMASH dataset. The proposed method input parameters such as CPU, network bandwidth, and memory, are virtual machines or healthcare service providers. The objective function value of the proposed method is minimum load and maximum resource utilization. The Kademia Hash Function-based data collection task is performed to protect and keep patient data privacy with the assistance of node ID generation, routing table initialization, network joining, and routing data storage in a cloud server. Here, the Kademia Hash Function is applied to generate a hash value for each patient collected data and stored on cloud servers to scale resource allocation based on demand. Second, Snow Ablation Optimization is carried out with the intent of identifying the resource optimized virtual machine for performing scheduling of stored patient data. Third and finally, Stride Scheduling is used to balance overloaded virtual machines with less loaded virtual machines with the objective of making certain efficient resource allocation and scheduling in mobile computing.

Kademia hash function-based data collection

Multilevel Monitoring of Activity and Sleep in Healthy People (MMASH) dataset from Physionet (<https://physionet.org/content/mrash/1.0.0/>) imparts 24 h of continuous sleep quality, beat-to-beat heart data, physical activity, triaxial accelerometer data, and psychological features, to name a few, including anxiety status, emotions, and stress events for 22 healthy participants. In addition, saliva bio-markers like cortisol and melatonin, along with an activity log are also included in the dataset, hence correlating between physical activity, sleep quality, and psychological characteristics. The data were collected and provided by BioBeats in association with researchers from the University of Pisa, who generate IoT wearable devices striving to detect people's psycho physiological stress. At the outset of the data recording process, certain anthropomorphic features like age, height, and weight of the participants were recorded. In addition, a set of inceptive questionnaires that dispense information about participants' psychological status was also recorded. During the test, participants wore two devices for 24 h, namely, a heart rate monitor to record heartbeats and beat-to-beat interval and an actigraph to record actigraphy information like accelerometer data, sleep quality, and physical activity. Both Positive and Negative Affect Schedules were also recorded at different times of the day. Twice a day, the participants acquired saliva samples at home in pertinent vials. Saliva samples were used to extract RNA and to assess specific hormones. In the proposed KHSAROSS method, the number of patient data is collected using the MMASH dataset. The size of the MMASH dataset is 23.5 MB. Patient data is being collected from the MMASH dataset in Table 1.

Collected data is used to perform two processes, namely data normalization and feature selection. In the data obtaining step, sensors are used to gather the patient's physiological conditions like temperature, heart rate, and EEG data to carry out mobile phone-based remote healthcare monitoring. After data collection, the Canberra Match data normalization based Pre-processing technique is applied to process the raw dataset into a structured format. It helps to increase the quality of the data, provides more accurate analytics and decision-making for achieving improved patient care, and more effective healthcare functions. Once data normalization is completed, relevant feature selection is performed to select only the most pertinent features from the input data using the Elasticnet Regressive Attribute Selection technique. Thus, it achieves higher scheduling accuracy with minimal time consumption. With the above set of selected features, the Kademia Hash Function is applied to each patient's collected data to generate a corresponding hash value. Figure 2 shows the structure of the Kademia Hash Function-based data collection model.

Figure 2 illustrates that the Kademia Hash Function-based data storage on a cloud server involves around four steps, namely, generation of node ID, routing table initialization, network joining, and routing data storage in the cloud server. Let us consider the MMASH dataset 'DS' with 'N' samples or users corresponding to seven different files (i.e., user information, sleep, RR, questionnaire, activity, actigraph, and saliva) for each participant with an overall of '69' features. The information in the dataset is mathematically formulated as given below.

$$DS \rightarrow \{user, sleep, RR, questionnaire, activity, actigraph, saliva\} \quad (1)$$

The above seven files present in the dataset are then split in the form of an input vector matrix representation as given below.

$$IM = \begin{bmatrix} S_1 F_1 & S_1 F_2 & \dots & S_1 F_M \\ S_2 F_1 & S_2 F_2 & \dots & S_2 F_M \\ \dots & \dots & \dots & \dots \\ S_N F_1 & S_N F_2 & \dots & S_N F_M \end{bmatrix} \quad (2)$$

From the above formulation according to Eq. (2), the input vector matrix 'IM' is generated with respect to 'N' samples 'S_N' and 'M' features 'F_M'. Let us consider a Kademia network of 'N' samples with IDS 'S₁, S₂, ... S_N' each of which denotes a string length 's' corresponding to '1' and '0'. Kademia network of 'N' samples in the proposed method is modeled as a trie or prefix tree, wherein each lead denotes a node, whereas the labeled path between the root and a lead denotes its ID. Then, the generation of node ID for each participant employing the Division Hashing function is formulated as given below.

$$Key \rightarrow H(IM) = IM(modTS) \rightarrow Hashvalue \quad (3)$$

From the above formula, according to Eq. (3), the Division Hashing function 'H' results for the corresponding input matrix 'IM' are arrived at based on the modulo function 'mod' by selecting a divisor 'TS', a prime number close to the table size. Then, for a sample of input vector matrix 'IM ∈ {IM₁, IM₂, ... , IM_n}'; let 's_i(IM)' denote the set of samples of input vector matrix (IDs) that share a prefix with 'IM' of length 's - i'. Nodes or the samples initiate a routing table employing 'k' structure as a table with every bucket containing contact information for other nodes or samples with their IP address and node ID, respectively. Then, filling the

S. No	Files	Description along with features
1	User_info [anthropocentric characteristic]	<ol style="list-style-type: none"> 1. Gender - M and F refer to Male and Female 2. Height- It denoted in centimetre (cm) 3. Weight- It measured expressed in kilograms (kg) 4. Age -It expressed in years
2	Sleep [participant sleep duration and sleep quality]	<ol style="list-style-type: none"> 1. In bed date- 1 and 2 define to the first and second day of data recording. 2. In bed time-time of day (hours, minutes) when the user go to the bed. 3. Out bed date-1 and 2 define to the first and second day of data recording 4. Out bed time-time of day (hours, minutes) when the user go out of the bed. 5. Onset date- 1 and 2 define to the first and second day of data recording 6. Onset time-time of day (hours, minutes) when the user falls asleep 7. Latency efficiency-ratio of sleep time on whole sleep in bed 8. Total minutes in bed-Minutes exhausted in the bed per night 9. Total sleep time-Duration of the sleep per night expressed in minutes. 10. Wake after sleep onset-time spent awake after falling asleep the first time. 11. Awakenings during night 12. Average awakening length-time in seconds spent awakening through the night. 13. Movement index-number of minutes without movement denoted as a percentage of the movement phase 14. Fragmentation index-number of minutes with movement represented as a percentage of the immobile phase 15. Sleep fragmentation index-proportion among the Movement and Fragmentation indices.
3	RR [beat-to-beat interval]	<ol style="list-style-type: none"> 1. ibi_s - time in seconds among two consecutive beats 2. day - 1 and 2 define to the first and second day of data recording 3. time - day time when the heartbeat happened
4	Questionnaire [scores for questionnaire]	<ol style="list-style-type: none"> 1. MEQ - Morningness - Eveningness Questionnaire value 2. STAI1-State Anxiety value achieved from State-Trait Anxiety Inventory 3. STAI2-Trait Anxiety value attained from the State-Trait Anxiety Inventory 4. PSQI-Pittsburgh Sleep Quality Questionnaire Index 5. BIS/BAS-Behavioural avoidance/inhibition index 6. Daily stress- Daily Stress Inventory value (DSI) 7. PANAS-Positive and Negative Affect Schedule
5	Activity [activity categories throughout day]	<ol style="list-style-type: none"> 1. Sleeping 2. Laying down 3. Sitting 4. Light movement 5. Medium 6. Heavy 7. Eating 8. Small screen usage 9. Large screen usage 10. Drink consumption 11. Smoking 12. Alcohol consumption
6	Actigraph [accelerometer and inclinometer data]	<ol style="list-style-type: none"> 1. Axis 1-Raw Acceleration data of the X-axis denoted in Newton-meter. 2. Axis 2-Raw Acceleration data of the Y-axis represented in Newton-meter. 3. Axis 3-Raw Acceleration data of the Z-axis indicated in Newton-meter. 4. Steps-Number of steps per second. 5. HR-Beats per minutes (bpm). 6. Inclinometer off-Values equal to 1 define to no activation of the inclinometer. The values are reported per second. 7. Inclinometer sitting-values equal to 1 denoted to the standing position of the user and 0 indicated to other user positions. Values are reported per second. 8. Inclinometer lying-values equal to 1 represented to the sitting position of the user and 0 indicated to other user positions. Values are reported per second. 9. Vector magnitude-Vector movement derived from raw acceleration data denoted in Newton-meter. 10. Day- 1 and 2 define to the first and second day of data recording 11. Time- day time when the heartbeat happened
7	Saliva	Clock genes, hormone concentrations [two samples]/ person are included

Table 1. MMASH dataset.

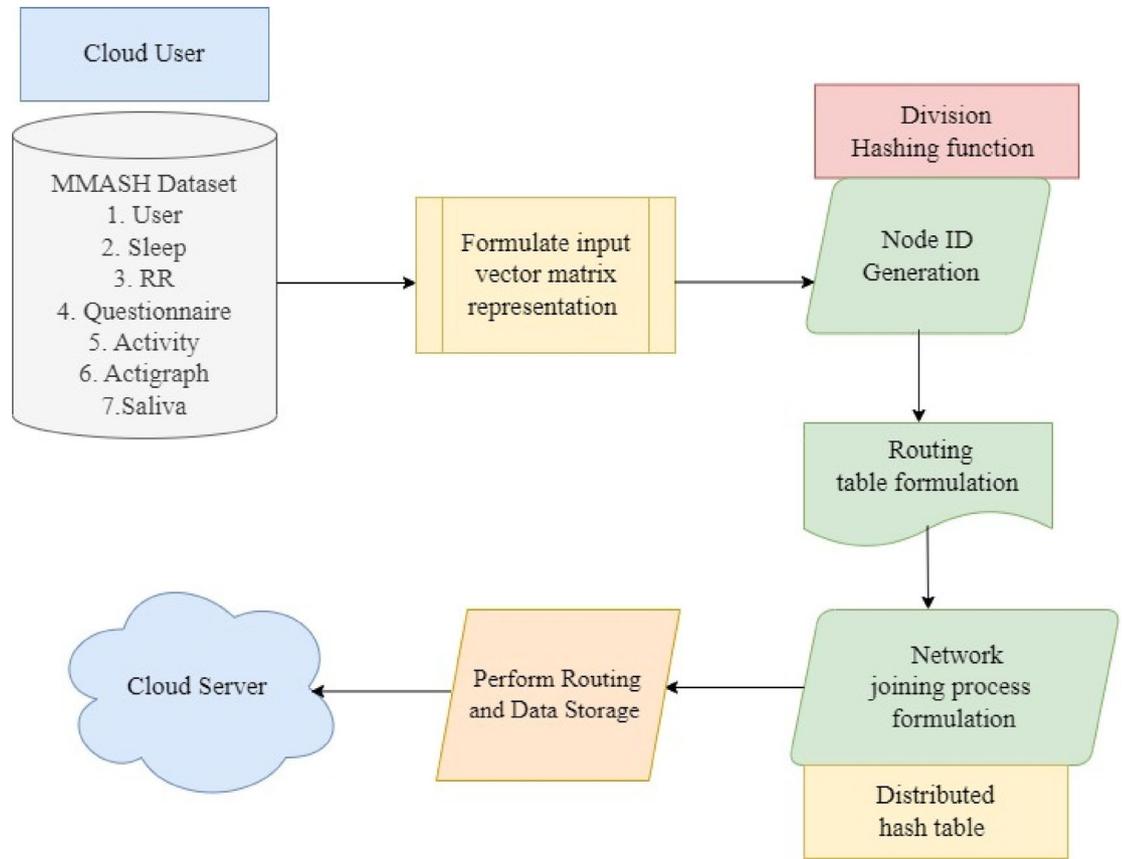


Fig. 2. Structure of Kademia Hash Function-based data storage.

‘ $i - th$ ’ bucket is said to be modeled as inserting pointers from the leaf ‘ IM ’ to ‘ k ’ leaves arbitrarily. Hence, routing is said to be jumping among leaves along these pointers in such a manner so as to go towards the target ID to the fullest extent in a greedy fashion, as given below.

$$RT [IM] \rightarrow AddingIM_i [H (IM)] toIM_k [H (IM)] \tag{4}$$

With the above routing table formulated according to Eq. (4), when a node or sample becomes a member, contact with other samples is said to be initiated, so that network joining is said to be ensured robustly. Let ‘ J_{xy} ’ represent the number of jumps required to traverse from leaf ‘ x ’ to target ‘ y ’. Then, the network joining process via a distributed hash table is mathematically represented as given below.

$$\sup_{IM_1, IM_2, \dots, IM_n} \sup_{IM \in \{IM_1, IM_2, \dots, IM_n\}} \sup_{y \in \{0,1\}^s} E [J_{xy}] \leq (1 + o(1)) \frac{\log n}{H_k} \tag{5}$$

From the above Eq. (5) ‘ H_k ’ denotes the harmonic number where only ‘ $O(\log(n))$ ’ samples are contacted in searching for a target node to be stored in the routing table. Finally, the routing and data storage in the cloud server are performed as given below.

$$CS \rightarrow J_{xy} [IM_i] = \frac{\log n}{C_k} [IM_i] \tag{6}$$

From the above Eq. (6), the routing and data storage information for each sample or cloud user ‘ $J_{xy} [IM_i]$ ’ via input matrix is stored in the cloud server ‘ CS ’ for further processing. The pseudo code representation of Kademia Hash Function-based data collection is as given below.

Input: Dataset ‘ DS ’, Samples or Cloud Users ‘ $S = \{S_1, S_2, \dots, S_N\}$ ’, Features ‘ $F = \{F_1, F_2, \dots, F_M\}$ ’
Output: makespan and latency-efficient data collection
Step 1: Initialize ‘ $N = 22$ ’, ‘ $M = 69$ ’, table size ‘ TS ’ Step 2: Begin Step 3: For each ‘ DS ’ with ‘ S ’ and ‘ F ’ Step 4: Formulate input vector matrix representation according to (1) and (2) // Generation of Node ID Step 5: Generate Node ID using Division Hashing function according to (3) // Initialization of routing table Step 6: Formulate the routing table according to (4) // Network joining Step 7: Formulate the network joining process according to (5) // Routing and data storage in cloud server Step 8: Perform routing and data storage in the cloud server according to (6) Step 9: End for Step 10: End

Algorithm 1 Kademia Hash Function-based data collection.

As given in the above algorithm, the data collection processing using the Kademia Hash Function is split into four steps. Initially, with the corresponding input vector matrix representation for respective samples or cloud users, by employing the Division Hashing function, undergoes a process of generation of node ID, therefore, with fast lookup time ensures minimum makespan time. Next, according to the results of the Division Hashing function, the results routing table is generated. Followed by which network joining process for each sample or cloud users is generated via a distributed hash table. Finally, the data stored in the cloud server not only serves as the identification process but also employs the node ID in locating the corresponding information of each cloud user. This in turn, aids in minimizing the overall latency involved extensively.

Snow ablation-based resource optimization and Stride scheduling model

With the data collected and stored in a cloud server, the next step in the proposed method is to design a resource optimization model. In this section Snow Ablation-based Resource Optimization and Stride Scheduling model is designed. Snow Ablation Optimization is a meta-heuristic process inspired by two processes that turn snow into liquid water via melting and liquid water into steam via evaporation, permitting a balance between exploration (searching a wide range of solutions) and exploitation (concentrating on fine-tuning optimal solutions). The Snow Ablation Optimization is carried out to identify resource optimized virtual machines for performing scheduling of stored patient data. Figure 3 shows the structure of the Snow Ablation-based Resource Optimization and Stride Scheduling model.

Figure 3 illustrates the process of Snow Ablation-based Resource Optimization and Stride Scheduling model is split into two parts. In the first part, optimization is performed using Snow Ablation-based Resource Optimization, and scheduling is done employing the Stride Scheduling function. The Snow Ablation-based Resource Optimization and Stride Scheduling model is designed to increase scheduling accuracy with the help of optimization and scheduling techniques. Stride Scheduling is used in the proposed method with the purpose of balancing overloaded virtual machines with less loaded virtual machines, which makes certain efficient resource allocation and scheduling in mobile computing. This scheduling process results in precise control over proportional resource allocation between competing processes. For example, a different disease dataset is given as input to Stride Scheduling in the healthcare system. Here, the stride scheduling function estimation and precise VM balancing steps are performed. The Stride Scheduling function is estimated by the difference between the number of loads of process by a virtual machine to the large number (virtual machine). This stride scheduling process is used with more samples receiving a larger share of the resource over time, therefore achieving better throughput for that process.

Snow Ablation-based resource optimization model

Snow Ablation-based Resource Optimization model is constructed with three components to identify the optimal solution, i.e., the initialization phase, the exploration phase, and the exploitation phase. The iterative process of the Snow Ablation-based Resource Optimization model initiates with arbitrarily randomly generated populations. The entire arbitrarily randomly generated populations are denoted by a matrix with an organization of ‘ R ’ rows and ‘ C ’ columns as provided in Eq. (7). Here ‘ R ’ represents to total number of agents (i.e., virtual machines or healthcare service providers) and ‘ C ’ represents the dimension of solution space, with upper and lower boundaries of search domain being ‘ UB ’ and ‘ LB ’ respectively.

$$P = LB + rand * (UB - LB) = \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1C} \\ p_{21} & p_{22} & \dots & p_{2C} \\ \dots & \dots & \dots & \dots \\ p_{R1} & p_{R2} & \dots & p_{RC} \end{bmatrix}, \text{ where } p \in P \text{ [i.e.data stored in } CS] \quad (7)$$

Then the ‘ $i - th$ ’ search agent position (i.e., samples or cloud users) is then described as ‘ $P_i = (p_{i1}, p_{i2}, \dots, p_{iC})$ ’. Distinct candidate solutions equivalent to discrete function values, and these function values are mathematically formulated as given below.

$$F = [F(P_1), F(P_2), \dots, F(P_R)] \quad (8)$$

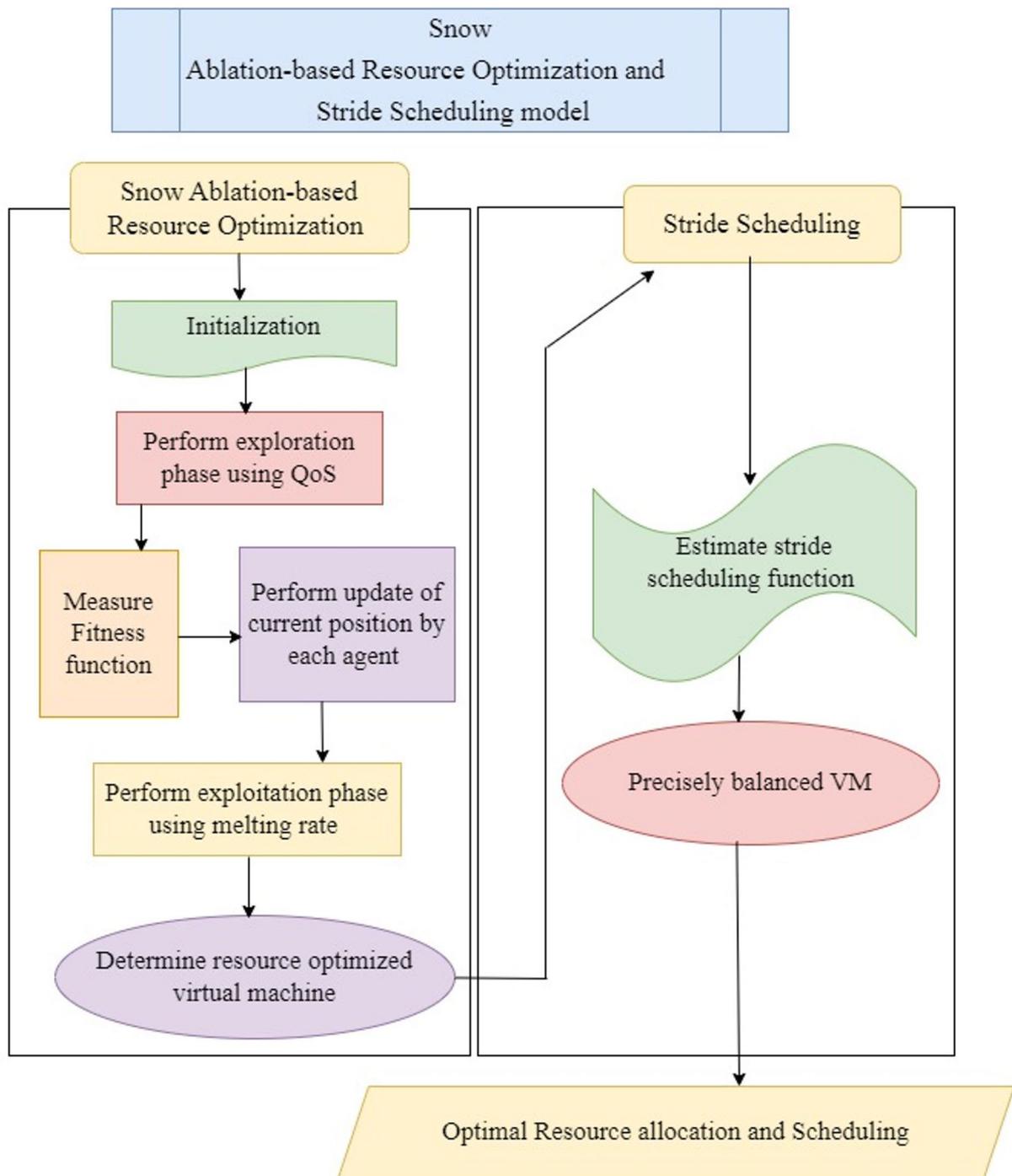


Fig. 3. Structure of Snow Ablation-based Resource Optimization and Stride Scheduling model.

In Snow Ablation Optimization, a meta-heuristic, the candidate solution quality is measured by the function value of the problem (CPU, network bandwidth, and memory). Therefore, the agent (i.e., virtual machines or healthcare service providers) analogous to the best objective function value (i.e., minimum load and maximum resource utilization) is considered the best agent. In this paper, it is assumed that the minimization (i.e., load) and maximization (i.e., resource utilization) problem is solved, and hence, the agent (i.e., virtual machines or healthcare service providers) with the minimum and maximum value of ' F ' is the best agent. Then, the load of a VM on a physical node is resources of CPU, memory, network to consuming and how that usage impacts the overall performance of the physical server. It is mathematically formulated as,

$$L_{VM_j} = \frac{1}{1 - CPU_{UR_{j,i}}} * \frac{1}{1 - Net_{UR_{j,i}}} * \frac{1}{1 - MEM_{UR_{j,i}}} \quad (9)$$

From the above Eq. (9), ' L_{VM_j} ' denotes the load of ' $j - th$ ' virtual machine ' VM_j ' corresponding to ' $i - th$ ' physical node ' S_i ' (i.e., cloud user) and ' $CPU_{UR_{j,i}}$ ', ' $Net_{UR_{j,i}}$ ', ' $MEM_{UR_{j,i}}$ ' representing the utilization rate of its CPU, network bandwidth, and memory, respectively. After that, the resource utilization is measured to utilize this rate, it's essential to calculate it accurately, track its running time, and then use the insights to optimize resource allocation and enhance the overall efficiency and profitability. It is formulated as,

$$RUR = \sum_{i=1}^N \sum_{j=1}^J \frac{UR_{j,i}}{f_i} \quad (10)$$

Also, according to Eq. (10), ' RUR ' denotes the resource utilization rate or the entire population in the sample space, and also ' f ' represents the flag with ' $f_i = 1$ ' denoting certain cloud user requested tasks are running on virtual machine ' VM ' on ' $i - th$ ' physical node ' S_i ' and on the contrary the flag with ' $f_i = 0$ ' denoting certain cloud user requested tasks not running on virtual machine ' VM ' on ' $i - th$ ' physical node ' S_i '. Finally, the ' RUR ' is improve the efficiency, lesser costs. Hence, the QoS has to meet the following conditions as given below.

$$F = \text{Min} (L_{VM_j}) \text{ and } \text{Max} (RUR) \quad (11)$$

From the above equation results Eq. (11) on the basis of minimal load and maximal resource utilization rate, the optimal virtual machine (i.e., healthcare service providers) is selected for performing scheduling of stored patient data. Eventually, when the load or the resource utilization rate is violated, reallocation or right VMs are performed via the exploration and exploitation phase.

Second, in the exploration phase, the Snow Ablation-based Resource Optimization model simulates the transition of snow and liquid water to steam with the intent of arriving at achieve an exploration of the problem space. Then, each agent (i.e., virtual machines or healthcare service providers) updates its current position as given below.

$$P_i^{t+1} = P_{elite}^t + BM_i * (Cons_1 * (P_{best}^t - P_i^t) + (1 - Cons_1) * (P_{mean}^t - P_i^t)) \quad (12)$$

From the above Eq. (12) ' P_i^{t+1} ' denotes the position of the ' $i - th$ ' agent or the virtual machine in ' $t + 1$ ' iteration, with ' BM_i ' denoting the Brownian random numbers, ' $Cons_1$ ' denoting the arbitrarily generated value, ' P_{best}^t ' representing the best solution with minimum fitness obtained so far for the current iteration ' t ' respectively. Moreover, ' P_{mean}^t ' and ' P_{elite}^t ' denotes the current mean location and agents or virtual machine selected from the elite pool, mathematically formulated as given below.

$$P_{mean}^t = \frac{\sum_{i=1}^R P_i^t}{R} \quad (13)$$

$$P_{elite}^t = [P_{best}^t, P_{second}^t, P_{third}^t, \dots] \quad (14)$$

From the above Eqs. (13) and (14), ' P_{second}^t ' and ' P_{third}^t ' represents the second best agent or virtual machine and the third best agent or virtual machine.

Third in the exploitation stage, when the snow melts and becomes liquid water, search agents (i.e., virtual machines or healthcare service providers) are prompted to employ high-quality solutions adjacent to the current best solution. The exploitation stage of the Snow Ablation-based Resource Optimization model motivates searching agents to concentrate on exploitation encompassing the current optimal agent.

$$MR^t = \left(0.35 + 0.25 * \frac{e^{\frac{t}{t_{max}}} - 1}{e - 1} \right) * e^{-\frac{t}{t_{max}}} \quad (15)$$

From the above Eq. (15), ' MR^t ' represents the melting rate at ' $t - th$ ' iteration and accordingly the position update to identify resource optimized virtual machine for performing scheduling of stored patient data is mathematically formulated as,

$$P_i^t = MR^t * P_{best}^t + BM_i * (Rand_2 * (P_{best}^t - P_i^t) + (1 - Rand_2) * (P_{mean}^t - P_i^t)) \quad (16)$$

From above Eq. (16), the ' P_i^t ' is position update to ' P_{best}^t ' denotes the best solution, ' P_{mean}^t ' denotes the current mean location and agents or virtual machine selected from stored patient data. It utilized to schedule and optimize the storage and retrieval of patient data in a resource-efficient manner. By using Kademia hash function, patient data can be ensuring high availability and fault tolerance. The hash function within Kademia data to specific nodes, enabling efficient data location and retrieval based on the hashed identifiers. The proposed KHSAROSS method carried out normalization after data collection. It increases the accuracy of the designed process. Then, the hash value for each patient collected data using Kademia hash function to optimize resources

and then virtual machine to achieve better performance of patient data scheduling process. It improves the resource utilization by distributing the storage and processing load and can also enhance data security and privacy through decentralized storage.

Stride scheduling

Finally, Stride Scheduling is utilized in the proposed method with the intent of balancing overloaded virtual machines to less loaded virtual machines. This in turn, makes certain efficient resource allocation and scheduling in mobile computing. The Stride Scheduling is mathematically formulated as given below.

$$SSf = \frac{\text{a large Equation Number [virtual machine]}}{\text{the Equation Number of loads of process by virtual machine}} = \frac{VM_{LN}}{VM_L} \quad (17)$$

From the above Eq. (16), the Stride Scheduling function ‘ SSf ’ results are arrived at based on the large number of virtual machines in the simulation setup ‘ VM_{LN} ’ and the number of loads by virtual machine ‘ VM_L ’. In this manner, Stride Scheduling function results aids in balancing overloaded virtual machines to less loaded virtual machines precisely and accurately. The pseudo-code representation of Snow Ablation-based Resource Optimization and Stride Scheduling is given below.

Input: Dataset ‘ DS ’, Samples or Cloud Users ‘ $S = \{S_1, S_2, \dots, S_N\}$ ’, Features ‘ $F = \{F_1, F_2, \dots, F_M\}$ ’
Output: computationally-efficient and throughput improved scheduling
Step 1: Initialize ‘ $N = 22$ ’, ‘ $M = 69$ ’ Step 2: Initialize ‘rand between 0 and 1’, ‘ $Cons_1$ between 0 and 1’, ‘ $Rand_2$ between – 1 and 1’ Step 3: Begin Step 4: For ‘ DS ’ with ‘ S ’ and ‘ F ’ //Fetch details of routing and data storage in cloud server //Initialization phase Step 5: Perform the initialization process according to (7) and (8) //Exploration phase Step 6: Formulate fitness function according to (9) and (10) Step 7: Formulate QoS ‘ $F = \text{Min} (L_{VM_j}) \text{ and Max} (RUR)'$ Step 8: Perform an update of the current position by each agent according to (12), (13), and (14) //Exploitation phase Step 9: Formulate exploitation encompassing the current optimal agent using the melting rate according to (15) and (16) Step 10: Return resource optimized virtual machine Step 11: End for Step 12: For each ‘ DS ’ with ‘ S ’ and ‘ F ’ with resource optimized virtual machine //Scheduling Step 13: Evaluate ‘ SSf ’ according to (17) Step 14: Return balanced scheduled VM results Step 15: End for Step 16: End

Algorithm 2 Snow Ablation-based Resource Optimization and Stride Scheduling.

According to the above algorithm, the overall process is split into two parts, namely optimization and scheduling. Initially, the process of optimization is split into three portions, namely, initialization, exploration, and exploitation. Here, initially, details of routing and data stored in the cloud server are obtained. Following which, the initialization process is triggered according to the upper and lower boundaries of the search domain. Following which, the fitness function is generated according to the QoS factors (i.e., minimum load and maximum resource utilization rate) in the exploration stage. Based on the fitness function results position updated is made. Finally, the exploitation function is performed employing the melting rate, therefore returning resource optimized virtual machine both accurately and minimum time. On the other hand scheduling process is carried out using the Stride Scheduling function. By applying these functions result in achieve precise control over the proportional allocation of resources between competing processes, ensuring deterministic fairness. This in turn, aims to allocate resources proportionally to the cloud user requests assigned to each process, prompting a direct association between cloud user requests and the process’s throughput. In other words, a scheduling process with more cloud user requests or samples receives a larger share of the resource over time, hence achieving higher throughput for that process.

Results and discussion

This method introduces a resource-efficient mobile computing service in healthcare sector. The study used the MMASH dataset³². The proposed method is executed in Python high-level general purpose programming language. The hardware requirements of the model implementation include 8 GB RAM, a 2 TB hard drive, and a CORE i7 CPU. The results were assessed with regard to scheduling accuracy, scheduling time, latency, throughput, and makespan.

Training and testing performance

The KHSAROSS’s performance is evaluated by splitting the input MMASH dataset into a 80:20 ratio for training and testing. Table 2 lists the implementation parameters and their specifications.

S. No	Parameters	Specifications
1	Dataset name	MMASH dataset
2	Dataset link	https://physionet.org/content/mmash/1.0.0/
3	Dataset size	99.2 MB
4	Data packets	
5	Number of virtual machines	10, 20, 30, 40, 50, 60, 70, 80, 90, 100
6	Training ratio	80
7	Testing ratio	20
8	Implementation platform	Python
9	Version	3.10.11

Table 2. Parameter specifications.

Performance comparison with current methods

In this section, the results obtained using the proposed KHSAROSS method were compared to those achieved with existing methods, including OO-DRL¹ and HMHHO². The outcomes metrics used for comparative evaluation include scheduling accuracy, scheduling time, makespan, latency, and throughput. The performance of these methods is arrived at by modeling and implementing them on the Python version 3.10.11 (same implementation tool used for evaluating the proposed method), for the common input MMASH dataset. Similar to implementing the proposed method, these existing methods¹ and ² will be trained and tested using the input based on Python 3.10.11. The results are determined by incrementing the sample size from 6000 to 60,000 (i.e., the actigraph of each user). For each sample size, analyze the above-stated outcome metrics that enable us to determine how the methods respond to different samples for mobile computing services.

Performance analysis of Makespan and latency

As far as cloud computing is concerned, makespan refers to the total time it consumes in accomplishing all the cloud users' requested tasks within a given workload. To be more specific, makespan refers to the time taken from the start of the first cloud user's requested task to the completion of the last cloud user's requested task. On the other hand, latency denotes the delay encountered when sending healthcare information between the cloud user and the cloud server. To be more specific, the latency rate refers to the time it takes for a single cloud user's requested data packet to travel from one point to another on the cloud computing network.

$$MS = \sum_{i=1}^N S_i * Time (St - Ft) \quad (18)$$

From the above Eq. (18), Makespan 'MS' essentially refers to the time from when the first task starts 'St' to when the last task finishes 'Ft'. The lower the makespan more efficient the method is said to be and vice versa. Makespan is measured in terms of seconds (sec).

$$Lat = \sum_{i=1}^N S_i * Time [S - CS] \quad (19)$$

From the above Eq. (19), latency is 'Lat' refers to the delay between a cloud user's request (samples) 'Time (S)' and the response from the cloud server 'Time (CS)'. It is measured in terms of seconds (sec).

Figures 4 and 5 given above show the makespan and latency evaluation using the proposed KHSAROSS and two existing methods, OO-DRL¹ and HMHHO². The horizontal axis in the above two figurative representations denotes the virtual machines used for simulation. From both of the above graphical representations, increasing the number of virtual machines results in a decrease of makespan and latency because the higher the number of virtual machines, the greater the probability of allocating resources in an efficient manner and vice versa. Also, simulation results show betterment of makespan and latency improvements using the proposed KHSAROSS method upon comparison to¹ and ². The reason behind the improvement was owing to the application of the Kademia Hash Function-based data collection algorithm. By applying this algorithm with the use of the Division Hashing function, the corresponding input vector matrix representations for respective samples or cloud users were obtained. This in turn, ensured fast lookup time, therefore reducing the makespan time using the proposed KHSAROSS method by 29% compared to¹ and 27% compared to². In addition, based on the Division Hashing function results routing table was generated. Finally network joining process for each sample or cloud user was generated using a distributed hash table. This in turn, aids in reducing the overall latency of the proposed KHSAROSS method by 31% compared to¹ and 33% compared to².

Performance analysis of scheduling accuracy

Accurate scheduling is pivotal for effective mobile computing cloud operations, making certain cloud user requested tasks are completed on time, employing resources efficiently, and delivering optimal performance to cloud users. To be more specific, scheduling accuracy in mobile computing cloud operations refers to the degree

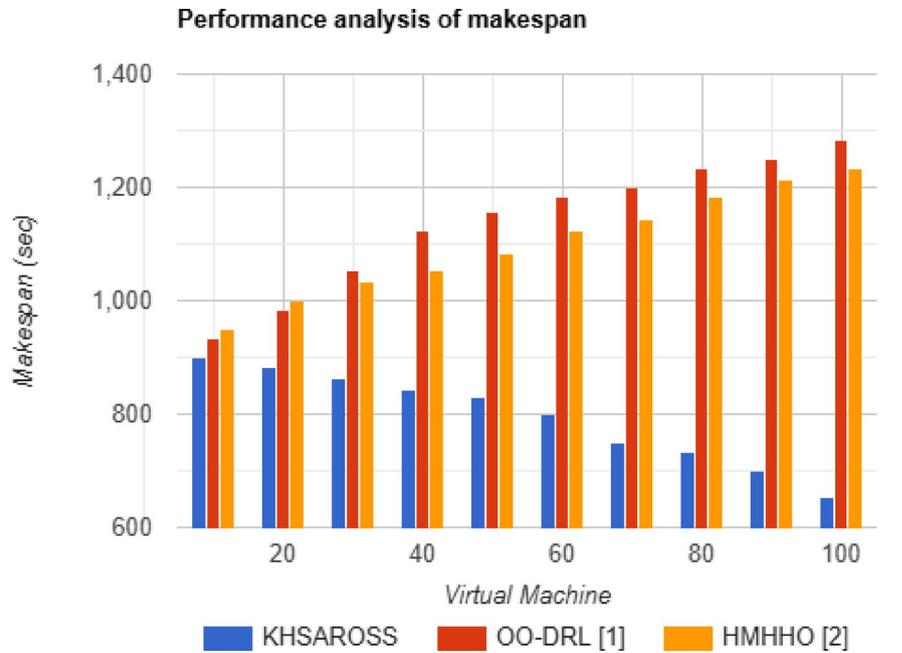


Fig. 4. Makespan evaluations of existing methods with the proposed method.

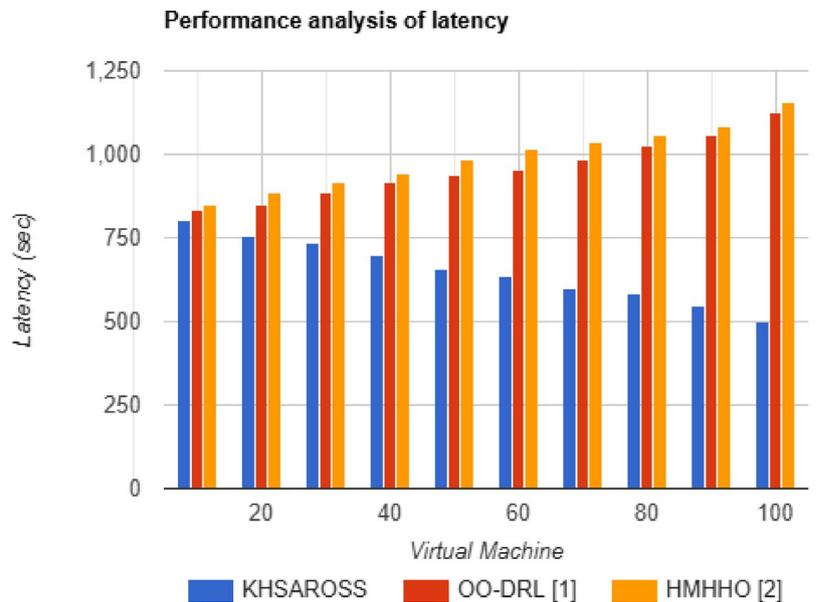


Fig. 5. Latency evaluations of existing methods with the proposed method.

to which the scheduling algorithm precisely allocates cloud user-requested tasks to the most pertinent available resources within the CC environment. Scheduling Accuracy ‘SA’ is mathematically represented as given below.

$$SA = \sum_{i=1}^N \frac{S_{SA}}{S_i} * 100 \tag{20}$$

From the above Eq. (20), ‘SA’ is measured based on the cloud user’s requested tasks or samples in the queue. S_i and the samples are scheduled accurately according to the requirements ‘ S_{SA} ’. Scheduling accuracy more efficient the method is said to be higher, and vice versa. It is measured in terms of percentage (%).

Figure 6, given above, illustrates the scheduling accuracy involved in the mobile computing services in healthcare. The horizontal axis in the above figure represents the samples or the cloud user requests placed in the cloud computing environment for performing mobile computing services involving healthcare data. From

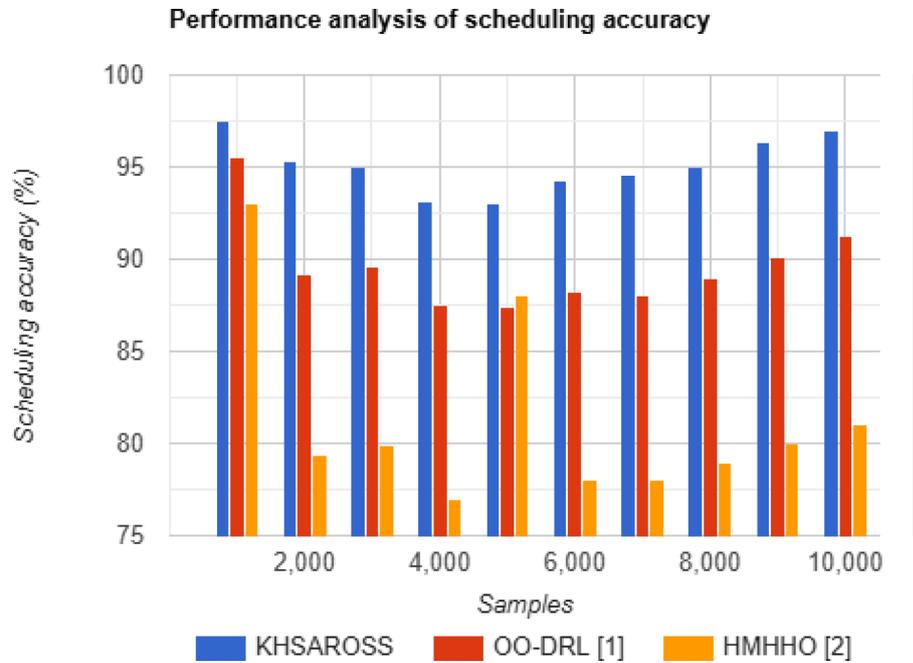


Fig. 6. Scheduling accuracy evaluations of existing methods with the proposed method.

the above figure, the scheduling accuracy was found to be neither increasing proportionally nor decreasing proportionally to the samples provided as input. This corroborates the objective that increasing or decreasing the sample size will not affect the output scheduling accuracy. Moreover, the scheduling accuracy using the proposed KHSAROSS method was found to be comparatively better than OO-DRL¹ and HMMHO². The scheduling accuracy improvement using the proposed KHSAROSS method would contribute to the application of Snow Ablation-based Resource Optimization and Stride Scheduling algorithm. By applying this algorithm first resource optimization was performed using candidate solution quality measured based on the QoS. This, in turn, improved the scheduling accuracy of the proposed KHSAROSS method by 6% compared to¹ and 17% compared to².

Performance analysis of scheduling time

The core function for mobile computing services in the healthcare sector is to map incoming cloud user-requested tasks to the appropriate virtual machines on the basis of factors like processing power, memory needs, and available capacity. To be more specific, scheduling time refers to the procedure of allocating computing resources like memory, bandwidth, and CPU time to different tasks with the objective of optimizing resource utilization by mapping tasks to available virtual machines. Scheduling Time ‘ ST ’ is mathematically represented as given below.

$$ST = \sum_{i=1}^N S_i * Time [S_{SA}] \quad (21)$$

From the above Eq. (21), ‘ ST ’ is measured based on the samples or the cloud user’s requested tasks in the queue. ‘ S_i ’ and the time consumed in scheduling accurately ‘ $Time [S_{SA}]$ ’. It is measured in terms of seconds (sec).

Figure 7 given above illustrates the scheduling time measured in terms of seconds (sec) with respect to sample size ranging between 1000 and 10,000. The horizontal axis in the above figure represents the samples and the vertical axis represents the scheduling time involved in performing mobile computing services of resource optimization and scheduling in healthcare sector. From the above figure, the scheduling time using all the three methods are neither said to be increasing or decreasing with the increase in the sample size, therefore corroborating the objective of efficient scheduling in an extensive manner. Moreover, the scheduling time of the KHSAROSS method was found to be comparatively lesser than¹ and ². The reason was due to the application of Snow Ablation-based Resource Optimization and Stride Scheduling algorithm. By applying this algorithm only after the optimization process of resources being allocated the scheduling process was performed employing Stride Scheduling. By applying this scheduling technique overloaded virtual machine were balanced with that of less loaded virtual machine. This in turn aided in improving the overall scheduling process, therefore reducing the scheduling time using KHSAROSS method by 18% compared to¹ and 22% compared to².

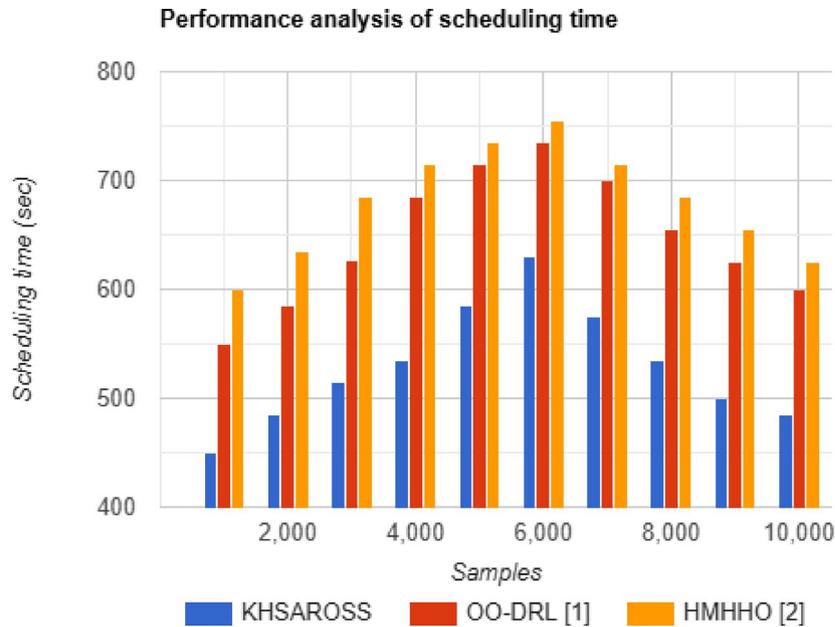


Fig. 7. Scheduling time evaluations of existing methods with the proposed method.

S. No	Methods	Throughput (bps)
1	KHSAROSS	1350
2	OO-DRL ¹	1235
3	HMMHO ²	1015

Table 3. Throughput evaluation.

Performance analysis of throughput

In mobile computing services performed via cloud computing, throughput is the amount of data that is transferred across a storage system within a given time period. High throughput means that more data or cloud user requests are said to be transferred in less time and vice versa. It is measured in terms of bits per second (bps). Table 3 tabulates the through rate of three different methods, KHSAROSS, OO-DRL¹ and HMMHO².

From the above table results the throughput analysis using the proposed KHSAROSS method was observed to be comparatively better than OO-DRL¹ and HMMHO². The throughput improvement was owing to the application of QoS fitness function in the exploration phase of the Snow Ablation-based Resource Optimization and Stride Scheduling algorithm and in exploitation stage, virtual machines or healthcare service providers are prompted to employ high-quality solutions adjacent the current best solution. This in turn aids in improving the overall throughput of proposed KHSAROSS method by 9% upon comparison to¹ and 22% upon comparison to².

Computation overhead (CO)

It is measured as an amount of time consumed to generate hash value using Kademia Hash function for data storage on cloud server.

$$CO = D * time(GH_{Value}) \quad (22)$$

where, D indicates a data, $time(GH_{Value})$ represents generating hash value time on cloud server. It is measured in terms of milliseconds (ms).

Figure 8 given above illustrates the Computational overhead computed in terms of milliseconds (ms) with respect to sample size ranging between 1000 and 10,000. The horizontal axis in the above figure represents the samples and the vertical axis represents the computational overhead involved in performing mobile computing services of resource optimization and scheduling in healthcare sector. Moreover, the computational overhead of the KHSAROSS method is comparatively minimum than¹ and ². The reason was due to the application of using Kademia Hash function. This in turn to minimize the computational overhead using KHSAROSS method by 5% compared to¹ and 19% compared to².

Storage complexity (SC)

It is defined as the amount of memory space utilized by data storing on cloud through the hash function and it formulated as follows.

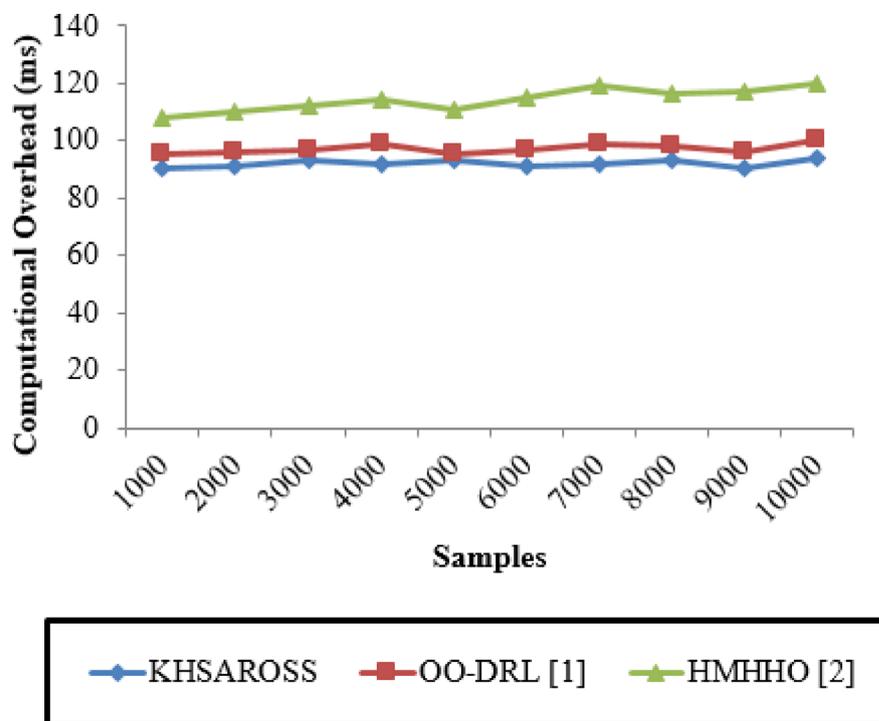


Fig. 8. Computational overhead evaluations of existing methods with the proposed method.

$$SC = D * \text{memory consumption} (DS) \quad (23)$$

where, 'D' represents a patient data, 'memory consumption (DS)' denotes memory space consumed by data storing on cloud server. The storage complexity is measured in terms of Kilobytes (KB).

Figure 9 given above illustrates the Storage complexity measured computed in terms of kilobytes (KB) with respect to sample size ranging between 1000 and 10,000. In the above figure, horizontal axis represents the samples and the vertical axis represents the memory consumed during mobile computing services of resource optimization and scheduling in healthcare sector. Moreover, the computational overhead of the KHSAROSS method is comparatively lesser than¹ and ². The reason was due to the application of using Kademia Hash function. This in turn to decreases the storage complexity using KHSAROSS method by 4% compared to¹ and 7% compared to².

Statistical test

F-test or ANOVA test is a statistical test for computing of two nominal predictor variables on a nonstop result variable. An output ANOVA test of has two independent variables on a dependent variable. ANOVA test outcomes were utilized in an F-test.

Table 4 shows a comparison of the ANOVA test for proposed and existing methods. The overall obtained result of the proposed KHSAROSS method offers better performance than other state-of-the-art methods.

Conclusion

Real-time data sharing between health-care providers ensures robust coordination of care plans that in turn aids in healthcare delivery to even underserved populations. A plethora of optimization techniques have been deployed to solve the issue of optimal resource usage and scheduling in mobile computing services in healthcare sector. In this work, scheduling process with resource efficient mobile computing services called, KHSAROSS is proposed. To begin, Kademia Hash Function-based data collection model was applied to the cloud user request schedules to perform efficient data collection with minimal latency and makespan. This eventually leads to a decrease in the makespan and improves resource efficient mobile computing services in healthcare sector. The Snow Ablation-based Resource Optimization and Stride Scheduling algorithm therefore returning resource optimized virtual machine accurately with minimum time. The results demonstrate that the proposed SQEGB method achieves throughput by 15% and scheduling accuracy by 12% than the existing methods. The results of the proposed KHSAROSS method is provided to minimize the makespan by 28%, latency by 32% and scheduling time by 20% compared to the conventional methods.

The limitation of proposed method is ineffective noise reduction during preprocessing and failed to address the patient data security with attack scenario. In future, different preprocessing method will be performed to reduce the noise. Also, the secure and efficient data storing methods will be designed for enhancing the security and enabling the attack detection in healthcare. In addition, the key generation, encryption, decryption and

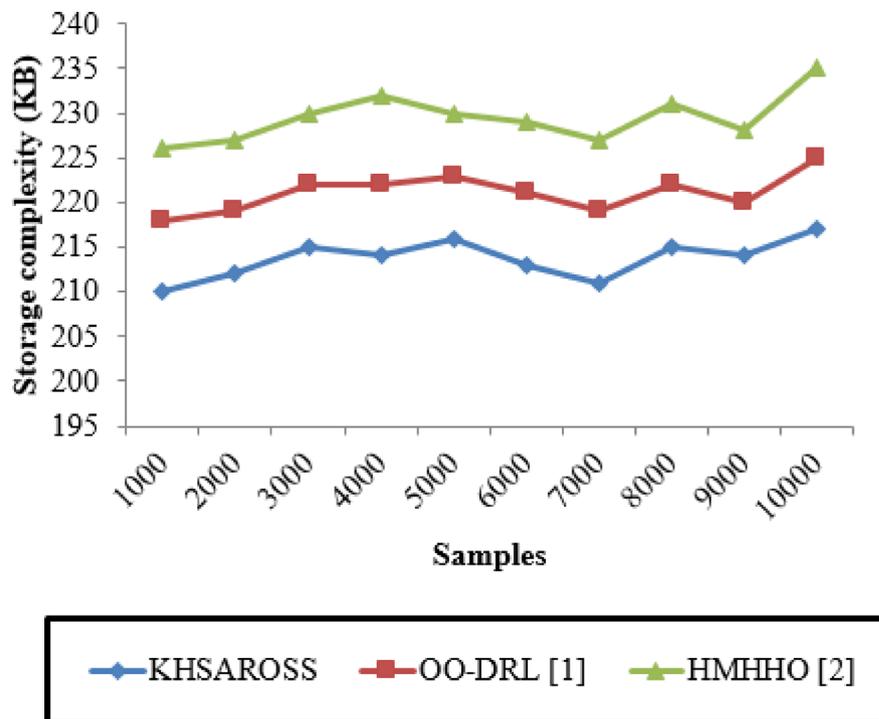


Fig. 9. Storage complexity evaluations of existing methods with the proposed method.

ANOVA test			
Parameters	OO-DRL ¹	HMMHO ²	Proposed KHSAROSS
Makespan (sec)	1141.3	1103	796.5
Latency (sec)	956.5	992.5	651
Scheduling accuracy (%)	89.565	81.32	95.115
Scheduling time (sec)	647.6	680.5	529.5
Throughput (bps)	1235	1015	1350
Computational overhead milliseconds (ms)	97.2	114.2	91.9
Storage complexity Kilobytes (KB).	221.1	229.5	213.7

Table 4. Comparison of ANOVA test for proposed and existing methods.

multi-factor authentication will be used to protect sensitive patient information from unauthorized access for early detection of potential attacks on healthcare systems.

Data availability

The full code and dataset involved in this work can be found at: <https://drive.google.com/drive/folders/1SgY6KrMxhhDBeFQ3ysqLRKBAle5v54je>.

Received: 11 April 2025; Accepted: 11 August 2025

Published online: 28 November 2025

References

1. Su, X. F. X., Cheng, Z., Gong, Z. & Choi, C. Deep reinforcement learning based latency-energy minimization in smart healthcare network. *Digital Communications and Networks*. Elsevier 1–15 (2024). <https://doi.org/10.1016/j.dcan.2024.06.008> [Online Offloading based on the Deep Reinforcement Learning (OO-DRL)].
2. Saemi, B. et al. Solving task scheduling problem in mobile cloud computing using the hybrid Multi-Objective Harris Hawks optimization algorithm. *IEEE Access*. **11**, 125033–125054. <https://doi.org/10.1109/ACCESS.2023.3329069> (2023). [Hybrid Multi-objective Harris Hawks Optimization (HMMHO)].
3. Olawade, D. B. et al. Artificial intelligence in healthcare delivery: prospects and pitfalls. *J. Med. Surg. Public. Health Elsevier*. **3**, 1–9. <https://doi.org/10.1016/j.jlmedi.2024.100108> (2024).
4. Rajagopal, A. et al. Machine Learning Operations in Health Care: A Scoping Review. *Mayo Clinic Proceedings: Digital Health*. **2** (3), 421–437 (2024). <https://doi.org/10.1016/j.mcpdig.2024.06.009>
5. Sektöründe, S. & Kullanım, Y. Z. Use of Artificial Intelligence in the Healthcare Sector. *Lokman Hekim Health Sciences*. **4** (3), 195–202 (2024). <https://doi.org/10.14744/lhhs.2024.27920>

6. Pradhani, B., Das, S., Roy, D. S., Routray, S. & Jhaveri, F. B. R. H., An AI-Assisted smart healthcare system Using 5G communication. *IEEE Access*. **11**, 108339–108355. <https://doi.org/10.1109/ACCESS.2023.3317174> (2023).
7. Santos, G. L. et al. The internet of things for healthcare: optimizing-health system availability in the fog and cloud int. *J. Comput. Sci. Eng.* **21** (4), 1–6. <https://doi.org/10.1504/IJCSE.2020.106873> (2020).
8. Raina, R. & Jha, R. K. Intelligent and interactive healthcare System(I2HS) using machine learning. *IEEE Access*. **10**, 116402–116424. <https://doi.org/10.1109/ACCESS.2022.3197878> (2022).
9. Zhao, X. & Wang, L. J. J. B. L., Service Migration Policy Optimization considering User Mobility for E-Healthcare Applications. *Journal of Healthcare Engineering*. Hindawi. 1–13 (2021). (2021). <https://doi.org/10.1155/2021/9922876>
10. Khatun, M. A., Memon, S. F., Eising, C. & Dhiran, L. L. Machine learning for Healthcare-IoT security: A review and risk mitigation. *IEEE Access*. **11**, 145869–145896. <https://doi.org/10.1109/ACCESS.2023.3346320> (2023).
11. Abdullah, N. et al. Optimization techniques for physicianscheduling problem: A systematic review of recent advancements and future directions. *IEEE Access*. **13**, 5203–5218. <https://doi.org/10.1109/ACCESS.2024.3524599> (2025).
12. Rajak, S. et al. Revolutionizing healthcare with 6G: A deep Dive Into smart, connected systems. *IEEE Access*. **12**, 194150–194170. <https://doi.org/10.1109/ACCESS.2024.3519567> (2024).
13. Alsaifi, T., Badshah, A., Aboulola, O. I. & Daud A., Optimizing healthcare big data performance through regional computing. *Sci. Rep.* **15** (1), 1–19. <https://doi.org/10.1038/s41598-025-87515-5> (2025).
14. Bapuram, B. et al. Extended water wave optimization (EWWO) technique: a proposed approach for task scheduling in IoMT and healthcare applications. *Evolutionary Intell.* Springer. **17**, 3609–3620. <https://doi.org/10.1007/s12065-024-00947-2> (2024).
15. Bhasker, B., Kaliraj, S., Gobinath, C. & Sivakumar, V. Optimizing energy task offloading technique using IoMT cloud in healthcare applications. *J. Cloud Computing: Advances Syst. Appl.* Springer. **14** (9), 1–11. <https://doi.org/10.1186/s13677-025-00733-0> (2025).
16. Sinha, R. The role and impact of new technologies on healthcare systems. *Health Syst. Springer*. **3** (96), 1–14. <https://doi.org/10.1007/s44250-024-00163-w> (2024).
17. Iqbal, A. A. et al. An efficient Dynamic-Decision based task scheduler for task Offloading optimization and energy management in mobile cloud computing. *Sens. MDPI*. **21** (13), 1–20. <https://doi.org/10.3390/s21134527> (2021).
18. Kaushal, R. K., Bhardwaj, R., Kumar, N. & Aljohani, A. Singh, P., Purohit, N., Using Mobile Computing to Provide a Smart and Secure Internet of Things (IoT) Framework for Medical Applications. *Wireless Communications and Mobile Computing*. Wiley. 1–13 (2022). <https://doi.org/10.1155/2022/8741357>
19. Jameil, A. K. & Al-Raweshidy, H. Healthcare and enhanced computational resource management with digital twins into task offloading strategies. *IEEE Access*. **12**, 90353–90370. <https://doi.org/10.1109/ACCESS.2024.3420741> (2024).
20. Eneyew, L. M. & Alamrew, B. D. Empowering mobile cloud computing in healthcare environment: Ethiopia. *Int. J. Innovative Res. Sci. Eng. Technol.* **8**, 1–5. <https://doi.org/10.15680/IJRSET.2019.0812075> (2019).
21. Wang, X. & Jin, Z. An overview of mobile cloud computing for pervasive healthcare. *IEEE Access*. **7**, 66774–66791. <https://doi.org/10.1109/ACCESS.2019.2917701> (2019).
22. Gopalakrishnan, L., Buback, L., Fernald, L., Walker, D. & Diamond-Smith, N. Using mHealth to improve health care delivery in india: A qualitative examination of the perspectives of community healthworkers and beneficiaries. *PLOS ONE*. **15** (1), 1–22. <https://doi.org/10.1371/journal.pone.0227451> (2020).
23. Free, C., Felix, P. G. W. L. G. L., Patel, L. E. P. & Haines, V. The effectiveness of Mobile-Health technologies to improve health care service delivery processes: a systematic review and Meta-Analysis. *PLoS Med.* **10** (1), 1–26. <https://doi.org/10.1371/journal.pmed.1001363> (2013).
24. Abdulmalek, S. et al. IoT based Healthcare-Monitoring system towards improving quality of life: A review. *IoT Healthc. Sens.* **10** (10), 1–32. <https://doi.org/10.3390/healthcare10101993> (2022).
25. Maharani, A., Sujarwoto, P. D., Tampubolon, O. D. & Patel, D. Implementation of mobile-health technology is associated with five-year survival among individuals in rural areas of Indonesia. *PLOS Digit. Health*. **3** (4), 1–11. <https://doi.org/10.1371/journal.pdig.0000476> (2024).
26. Shafi, I. et al. Design and development of patient health tracking, monitoring and big data storage using internet of things and real time cloud computing. *PLOS ONE*. **19** (3), 1–14. <https://doi.org/10.1371/journal.pone.0298582> (2024).
27. Sachdev, S. et al. Unraveling the role of cloud computing in health care system and biomedical sciences. *Heliyon Elsevier*. **10** (7), 1–15. <https://doi.org/10.1016/j.heliyon.2024.e29044> (2024).
28. Ananta, O. Z. & Richard, T. G. Mobile cloud computing services for an enhanced healthcare system. *Int. J. Eng. Technol.* **2** (2), 1–4. <https://doi.org/10.1016/j.ijinfomgt.2018.09.012> (2022).
29. Kedi, W. E., Ejimuda, C. & Ajegbile, M. D. Cloud computing in healthcare: A comprehensive review of data storage and analysis solutions. *World J. Adv. Eng. Technol. Sci.* **12** (02), 290–298. <https://doi.org/10.30574/wjaets.2024.12.2.0291> (2024).
30. Geng, J. Enhancing Crowd-Sourced video sharing through P2P-Assisted. *HTTP Video Streaming Electron.* **13**, 1–18. <https://doi.org/10.3390/electronics13071270> (2024).
31. Zhang, Y. & Venkatakrisnan, S. B. Kadabra: Adapting Kademia for the Decentralized Web. **1**, pp.1–27. <https://doi.org/10.48550/arXiv.2210.12858> (2023).
32. Da Pozo, R. A. et al. D, Multilevel Monitoring of Activity and Sleep in Healthy People. *PhysioNet*. **103** (23), 215–220 (2020). <https://doi.org/10.13026/ceqf-fc86>

Acknowledgements

This research is supported by Princess Nourah bint Abdulrahman University (PNU) Researchers Supporting Project number (PNURSP2025R194), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Author contributions

The corresponding author is the sole author and is the sole contributor to the work reported in this manuscript. The author confirms her contribution to the paper: study conception and design, data collection, analysis and interpretation of results, and draft manuscript preparation by Nithya Rekha Sivakumar. The author reviewed the results and approved the final version of the manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to N.R.S.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2025