



A novel cloud-assisted framework for consumer internet of things based on lanner swarm optimization algorithm in smart healthcare systems

V. Arulkumar¹ · M. Aruna² · D. Prakash³ · M. Amanullah⁴ · K. Somasundaram⁵ · Rajendran Thavasimuthu⁶

Received: 30 September 2023 / Revised: 21 January 2024 / Accepted: 28 February 2024 /
Published online: 12 March 2024

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract

Instantaneous data processing has the potential to enhance scalability, lessen power usage, and permit and improve data presentation in Consumer Internet of Things (CIoT) devices. In simple terms, cloud-based solutions cannot handle many IoT applications. According to Industrialized IoT (IIoT) technologies, an automated resource allocation system can improve service delivery and minimize healthcare costs. To maximize resource usage and response time for end users, there needs to be an effective method to efficiently distribute workload between Fog Layer and Cloud Connection and enhance cloud network capital allocation. Data analytics of complex and vital healthcare data requires timely responses, making it complicated. This paper proposes a design based on the Lanner Swarm Optimization (LSO) algorithm, which was developed to overcome inefficient heuristic strategies where data is transported to the cloud layer based on traffic type. The LSO algorithm is used to improve resource allocation and workload distribution in cloud-assisted CIoT applications for smart healthcare systems, improving scalability, power consumption, and data processing. The objective function determines if diverse virtual machines (VMs) vary accomplishment time the most, considering this study's updating and pruning restrictions. The experimentation analysis demonstrated that the proposed load balancing and work scheduling method outperforms evolutionary and heuristics algorithms. In experimentation, the research model attains a makespan of 10 s, response time of 5.5 s, resource utilization with a rate of 0.9, execution time of 13 s, latency of 10 ms, throughput of 0.78 s, and delivery rate of 0.74%. At resource scheduling, the LSO model had the best payload routing, latency, packet delivery ratio, and network lifetime.

Keywords Multimedia IoT · Cloud computing · Load balancing · Resource allocation and swarm intelligence

1 Introduction

The IoT, one of the fastest-growing technologies, offers the community, organizations, and consumers many opportunities [1]. It builds smart infrastructure in electricity, mobility, safety and protection, remote medical management, agriculture, smart homes, and smart cities using connecting devices. Figure 1 shows the differences between the applications in terms of Industrial IoT and CIoT [2]. Worldwide IoT may reach 26.4 billion by 2026, according to projections. Cellular technology will be 20% of this. The consumer-business IoT device ratio may be 45:55 [3]. The National Digital Communication Policy (NDCP) 2018 by the Department of Telecommunications calls for an ecosystem of 5 billion networked devices in 2022. Thus, India has 3 billion connected devices in 2022, 60% of the global total of 5 billion [4].

CIoT applications improve personal healthcare by integrating wearable IoT-connected devices that send data to doctors, families, and neighbors [5]. The public IoT is growing rapidly because of the requirement for personal healthcare IoT systems. They are also being utilized increasingly frequently in a range of different sporting events. Clinicians make use of them to maintain a level of awareness regarding the needs and well-being of their patients. Many CIoT devices track sleep, pulse, glucose, and other factors [6]. Although e-healthcare intends to improve patients' quality of life and reduce costs, health-monitoring device data has expanded dramatically, making virtualized central information processing increasingly difficult [7]. Cloud-based infrastructure affects real-time applications by moving local workloads to the cloud. Conventional techniques, which generally entail technology integrating sensing devices and the cloud, are impractical for providing healthcare to a large number of patients due to the lack of consideration for various healthcare issues [8].

Task scheduling and load balancing should fairly distribute loads among VMs to drive them to use resources efficiently, minimizing makespan and improving system efficiency. Even though cloud technology is dynamic, these ways are better because they apply more effective systematic and load-balancing strategies [9]. Delays, security flaws, or other issues detected in the cloud are permitted to achieve these standards. The increased

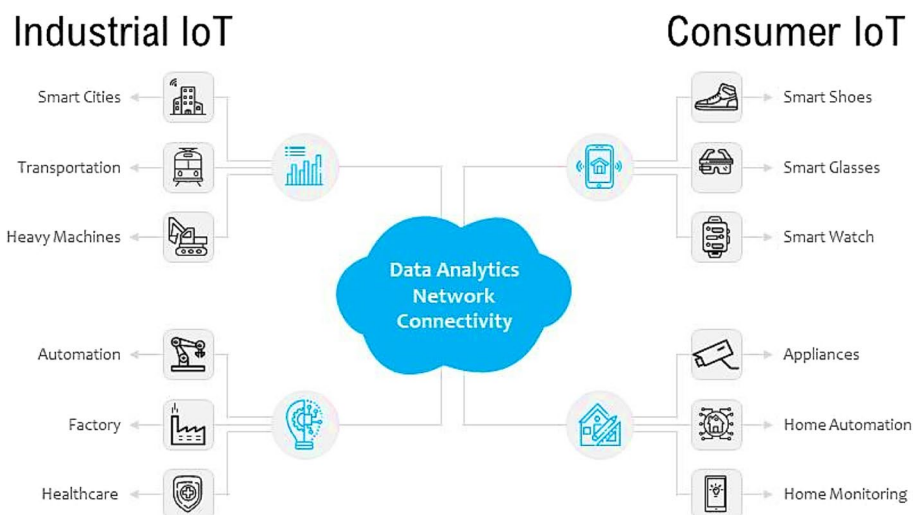


Fig. 1 Application differences between IIoT and CIoT

security, privacy, mobility, and network capacity were made possible and made more accessible so that the service could fulfill the needs of latency-sensitive or real-time applications. To realize the concept of huge computing and vast storage availability, devices included in the fog computing architecture are linked to the cloud [10]. On the other hand, how fog and clouds are distributed is also important. For applications needing low latency and quick response, for instance, healthcare CIoT systems, the fog computing-based architecture shows improved services [11].

In general, healthcare applications generate a lot of data that can be stored and accessed in real-time. Fog computing is excellent for this application since it fits the needs of this architecture. Processing and reaction times vary by device because heterogeneous data is produced differently. Each layer in a heterogeneous public cloud CIoT platform has features crucial for data transportation and processing pipelining. Hence, the research addresses this topic. This design efficiently allocates resources and balances the load to handle diverse data streams. The proposed work uses discrete traffic analysis to improve load balancing, latency, power consumption, and communication costs in e-healthcare. Previous surveys examined cloud utilization in IoT-enabled healthcare [12, 13]. It also highlighted the limitations of fog-based architectures and offered more reliable and secure IoT alternatives. The cloud-enabled IoT method solves this issue.

Since cloud-oriented processing has latency and power consumption constraints, fog-assisted frameworks build gateways between devices and the cloud to improve CIoT framework energy efficiency, dependability, and performance. Some data processing only requires lightweight processors since the input device tasks vary by application. Thus, assigning all functions to the same units is unsuitable. The proposed work uses a fog layer to prevent loss-sensitive, delay-sensitive, and security-sensitive data. It also reduces smartphone-to-cloud data transfer. Fog alone makes additional data accessible to users, not the cloud. In some cases, the fog node processes and prepares data before sending it to the cloud, lowering latency. Thus, rule-based filtering reduces data with the fog layer. Only the cloud processes security-sensitive data, which is then immediately made accessible to the final user. This CIoT architecture ensures optimal resource usage with prioritized load balancing, enabling differentiation and selection of the suitable policy for various data sources.

The novelty of this research is the dynamic adaption of the LSO algorithm, which strikes a unique balance between exploitation and exploration to improve resource allocation and workload distribution in cloud-assisted CIoT applications for smart healthcare systems for the purpose of optimizing resource allocation. The contributions of the work are as follows:

- The proposed load balancing method offers the model suffering latency, throughput, and network congestion metrics of the results generated for allocating the appropriate operating device of the diversified data traffic.
- An optimization problem that considers bandwidth and demand restrictions has been developed to address such problems. Additionally, the General idea has been modified to address the issue such that, at any given time, global particles indicate the advanced machining unit to be allocated to each data stream to get the best results.
- The originality of the proposed work is found in the LSO algorithm's adaption, which iterates with a new change in inertia weight to balance exploitation and exploration.
- The performance comparison between proposed and existing methods is done based on metrics such as makespan, resource consumption, execution time, latency, throughput, and delivery ratio.

The remaining sections of this research are organized as follows: Section 2 of the paper discusses the related works based on resource allotment and routing protocols for IoT in healthcare. Section 3 briefly discusses the implementation of the proposed technique. Section 4 explains the experimental findings and discussion. Section 5 discusses the conclusion and the work that will be done in the future.

2 Related work

In an IoT-enabled healthcare system, Li et al. [14] build an architecture for SDN-based Edge computing. The edge servers of this architecture used a simple authentication technique to verify the IoT devices' authenticity. These devices captured patient data after authentication and transferred it to edge devices for archiving, processing, and analysis. An SDN controller that managed the healthcare system's routing protocols, network optimization, and practical resource usage was connected to the edge servers. This framework was assessed through simulations that were performed on computers. The outcomes showed that the framework offered superior solutions for IoT-enabled medical systems. However, these underpowered devices and the patient data they were connected to were vulnerable to several security risks.

To reduce latency and network use, Asghar et al. [15] suggested a health monitoring system. In addition, because when the health monitoring framework was developed on a broad scale, a Load Balancing Scheme (LBS) was implemented for balancing the load across fog nodes. This work performed in-depth simulations using the sufficient form tools to evaluate the viability of the model and correlated the findings with the cloud implementation's Fog Node Placement Approach, denoted as FNPA, and Load Balancing, denoted as LAB scheme, in relation to network and latency use. In correlation to secure access, FNPA, and LAB scheme, the LBS model of health tracking system dramatically lowers latency and network use. However, the substantial deployment of proxy healthcare applications was constrained due to the significant delay produced by cloud-based framework when processing huge data.

Through considering the resources in instances format of VMs, Dubey et al. [16] suggested a flexible cloud-fog integrated architecture for processing whole IoT applications and dramatically enhanced the latency, compute cost, packet forwarding, and usage of energy. The Cuckoo Search Optimizer (CSO) and Particle Swarm Optimization (PSO) approaches were utilized in this work. This work has created a safe framework to handle IoT service allocation in a cloud infrastructure while reducing the key factors. The effectiveness of this work was thoroughly assessed using synthetic datasets and resource heterogeneity in a fog and cloud simulation scenario. According to the simulation findings, the hybrid metaheuristic algorithm performed better than alternative baseline policies and enhanced several key metrics. With this setup, users may access a subset of cloud services with minimal latency and power consumption at the edge node. However, they will be unable to run highly computational IoT applications.

An efficient resource allocation and prediction method for Fog environments that was appropriate for healthcare applications was introduced by Talaat in [17]. Resource allocation was an interesting task since it calls for a variety of resources and fog nodes to do the calculations necessary for the IoT systems. Through real-time resource allocation and prediction algorithms, this model aimed to manage resources effectively in a fog environment. The Data Processing Module (DPM), Resource Allocation Module (RAM), and

Effective Prediction Module (EPM) make up the three primary components of EPRAM. A target field was predicted by the EPM utilizing more predictions and the PNN. An ensemble classifier was developed using the given data to determine the likelihood of a heart attack. To determine the likelihood of a heart attack and choose the best course of action, PNN was then evaluated utilizing the patient's sensory data from the IoT layers. The system's primary objective was to reduce latency while enhancing Quality of Services (QoS) parameter, including bandwidth effectiveness, allocation response time, and consumption of energy. In contrast to earlier RA methods, the model used a novel model-free reinforcement Learning (RL) algorithm. Additionally, the PNN was utilized in the prediction process. The use of deep RL, as well as PNN, has enabled it to perform in such an acceptable manner. This model was efficient at tracking and immediately forecasting the patient's condition. However, if the servers' processing power varies, one of them can experience overload and crash.

A Structured Literature Survey (SLR) was presented by Ghanbari et al. [18], and the algorithms that were employed and resource allocation techniques in the IoT were examined. To examine the resource allocation strategies, several classifications, such as outlay, situation, efficiency-aware, huge pile, power-aware, QoS-aware, SLA-based, and appropriation capital allocation mechanisms, were organized. In each category, the work list discussed several parameters. Additionally, the parameters utilized in various papers were assessed, the key accomplishments in each area were reviewed, and the new difficulties were described. The structure of different technological keys around energy allocation as in IoT and its platforms was presented in this paper, along with the critical areas for future method improvements and the unresolved problems with resource distribution in the IoT that need to be addressed to maximize the technology's potential. This study demonstrates that there was no stand-alone method that could handle all the problems and difficulties associated with allocating resources for IoT.

Abdulhammed [19] developed an IoT-based healthcare system that consisted of two parts: establishing and resolving the issue of cloud computing load balancing by employing an intelligence algorithm called the sparrow search algorithm (SSA). Through using SSA, the optimal VM was selected from a group of VMs. The SSA was also used to schedule numerous and diverse tasks with primary consideration and allocate them to the best VMs based on their instruction millions (IM), only with the task with the greatest IM delegated to the best VM with the most significant fitness value. The results demonstrated that this method focused on minimizing network congestion and delay while producing or delivering in the medical system; likewise, the optimization model has evidenced its reliability, performance, and accomplishment when correlated to various models in the decrease of scheduling time, overall manufacturing time, and offers scheduling policy between many VMs, in which its value of resource utilization time, time consumption, and level of discrepancy has decreased. The focus of this strategy was on the lag in wireless connections, which was a drawback.

To reduce scheduling latency, Kanbar and Faraj [20] proposed the RADISH (Region Aware DynamIc SCheduling) model, which consisted of five successive processes. The first process used a task nature-based bi-class neural network to classify incoming tasks as sensitive or non-sensitive by considering the user's login information, email address, password, the services they use, and QoS parameters. The second step used a multi-criteria-based improved moth flame optimization (QoS aware AMFO) to schedule the categorized tasks while taking non-sensitive, sensitive, energy, priority, completion time, and workload into account. Due to this algorithm's strong convergence, the scheduling lag was decreased. The work conducted load balancing in the third step by recommending SAC

with a prospective field clustering approach. The VM categorization was considered while calculating local potential, energy, and density. To balance the workload and increase the effectiveness of the procedure, three repositories were created. To minimize allocation delay and enhance QoS, the work introduced the VM state-aware Hopcroft-Karp algorithm in the task allocation proposal. The work accomplished good SLA but also QoS in an IoT fog number of co-systems in this way. The simulation was carried out using the CloudSim simulation tool, which assessed effectiveness in terms of latency, connectivity, deadline, throughput, energy demand, CPU and memory use, SLA violations, and overhead. Additionally, the platform's delayed and operation latency may both increase because of this architecture.

Jangra and Mangla [21] suggested a congestion control model for resource allocation implementation in cloud-based healthcare settings. In this work, the RL approaches like SARSA, GA, and Q-learning were utilized. These approaches were applied in cloud-based medical facilities to predict the optimal technique for controlling load. This model reduced latency, had a fast production cycle, and was energy-saving. Utilizing MATLAB, the recommended procedure was implemented. Utilizing performance metrics like delay, makespan, and bandwidth, the performance of the model was evaluated. The model has a shorter make time than the current approach but a more significant throughput.

Abdelmoneem et al. [22] expanded the subject by proposing effective IoT architecture, resource allocation, and scheduling approaches for healthcare. To assist the patient's mobility, this model employed a delivery mechanism based on the adaptive Reference Signal Strengths (RSS). In this model, a mobility-aware heuristic-based scheduling and allocation approach (MobMBAR) enabled the dynamic spread of healthcare operations across computing nodes, whether cloud or fog devices. Patient activities and the temporal and territorial leftover of their visual information dynamically balance the distribution of task performance. Using task features such as threshold and optimum response time during the sorting and redeployment stages, the model aimed to minimize the total schedule time. The findings indicated that the frequency of tasks missed was less, that the system consumes 92% less energy, and that it has a Makespan that was 88% shorter than that of reducing systems.

In the study [23], a strategy for dynamic resource allocation based on optimization techniques and the evolutionary algorithm was proposed. This model employed a dynamic resources allocation strategy to manage requests incoming and distribute those fairly across the servers available while monitoring and controlling network traffic and collecting statistics on each server's load. As a result, performance was enhanced even during peak hours. Consequently, this model was efficient in genuine fog computing systems, like those utilized in the healthcare industry. The model focused on the design of an IoT-based healthcare framework. The IoT-Fog model consisted of an IoT, a fog, and an upper layer. The testing has finally been concluded, and the results reveal that the model improved the quality of service in the cloud/fog computing system by decreasing allocation costs and reducing response time. Consequently, this model was an effective way of measuring resource use and guaranteeing service continuity. This concept has consequences for privacy and security.

Meng et al. [24] reviewed the current state of the art of possible responses to the security and privacy issues raised by the IoT platforms. This work provided a detailed analysis of the number of new attacks on the speech interface of home automation platforms. These attacks were aimed at gaining illegal access and acting in ways that were too powerful to protect the privacy of the user. To counter these threats, a new voice liveness detection system was presented, which first analyzed the radio signals emitted by IoT devices and

then used the voice samples it received to verify the identity of the user. This work was executed on a real-world experimentation with Samsung's SmartThings platform to evaluate the effectiveness of the system and demonstrate its efficacy.

Baho et al. [25] helped to identify cybersecurity risks and control IoT vulnerabilities by providing insight into current methods for assessing the vulnerabilities. Readers from a wide variety of backgrounds were drawn to it, from experts in vulnerability management and cybersecurity risk research to academics specializing in the IoT. This study increased IoT security awareness and supported research into IoT risk assessment methodology by providing the most up-to-date perspective on current IoT vulnerability assessment approaches. Future scholars with an interest in IoT security challenges and solutions will benefit from the information offered by this work. Those attempting to develop new techniques to identify IoT vulnerabilities may find this work helpful because it clarifies the research direction in existing vulnerability assessment procedures.

Harkin et al. [26] discussed the results of interviews with 32 influential Australians in the disciplines of data security, regulation and policy, consumer and data privacy laws, and the IoT industry and academics. It described a wide range of problems and challenges, including those related to the effects on populations at risk, ecology, and the norms of IoT production, which extend beyond the well-known issue of privacy and the scientific requirements of data safety. However, the respondents did not identify any clear regulatory priorities or strategies, despite the consensus among key stakeholders that Australia needs stricter regulation. Future regulatory tactics and the consequences of these findings for the legalization of consumer IoT were discussed.

Verhoef et al. [27] provided a quick summary of the current state of knowledge on the connections between people, objects, and the natural environment, described the POP framework, and described how these linkages lead to an explosion in the volume of related data. This study also mapped out areas for further investigation into the ways in which the IoT and smart products may alter consumer habits and business practices.

Olga & Sarmah [28] conducted a study on the cyber security standards and ratings of CIoT requirements to determine their adequacy. Comparisons were made between Cyber Security for CIoT (CSCIoT) and other relevant projects, such as the Secure by Design study by the UK Department for Digital, Cultural Background, Journalism, and Sport and the worldwide professional IoT standard IEC 62443. Implications for consumer accountability in security were also discussed. The purpose of this analysis was to increase the specificity and breadth of criteria for consumer IoT devices to reduce the likelihood of cyberattacks.

Poyner et al. [29] considered the necessity of healthcare solution security and privacy frameworks. This work evaluated the limitations of these strategies within the context of an IoT system. Ngwenya and Ngoepe [30] used a Delphi method in conjunction with narrative inquiry to investigate South Africans' faith in CIoT data. This study primarily relied on semi-structured conversations, surveys, and unstructured interviews to obtain its data. Five experts were selected for the Delphi method based on their experience with IoT, either as sellers or buyers of IoT services or as providers of support services for IoT ecosystems. Six participants for the narrative inquiry were chosen using the snowball method based on their familiarity with consumer IoT solutions, their capability to provide comprehensive descriptions of their experiences, and their readiness to describe the lessons learned.

Wood et al. [31] proposed a means for automatically identifying cleartext data that may reveal private medical states and behaviors in internet traffic from medical IoT devices. The study was done in three steps: collecting traffic, finding clear text, and analyzing meta-data. Four widely used consumer medical IoT devices were examined, one of which was found to leak sensitive medical data in cleartext. An easy-to-use system for capturing and

analyzing network traffic was also provided, making it possible for users to keep tabs on the information flowing in and out of their homes via IoT devices. Alladi et al. [32] provided a detailed account of the threats that consumer IoT devices face and offered advice on how to defend against them. The research presented here should prove helpful in shaping how future IoT devices are developed.

Summary The literature review emphasized the threats associated with the widespread utilization of such (unsafe) devices in areas like smart cities, smart homes, and critical infrastructure. As the world becomes more reliant on IoT devices, security and privacy should not be an afterthought in their design. Attacks like those discussed in this section could have disastrous implications. Standards, prevention strategies (such as secure-by-design), continual testing and maintenance, and cross-sector partnerships are, therefore, urgently required to handle both current and future security threats.

3 Proposed methodology

To monitor patients, the CIoT equipment in the information gathering layer captures near the actual healthcare data as well as nonreal-time data. The IoT-focused access points serve as gateways to receive the collected data. This material can be handled at the cloud and fog layers, as detailed in the preceding sections, depending on the transportation class and processing needs. The proposed CIoT framework's throughput, energy efficiency, end-to-end (E2E) latency, and packet loss must all be improved through the effective processing of this data. The efficient handling of this diverse data while upholding QoS necessitates dynamic resource allocation. The CIoT framework of this research work, which is installed on top of the frameworks as a decentralized control layer for networking, resource allocation, scheduling, and flow control with the aid of LSO, depends on software-defined networking to accomplish this purpose. With grid virtualization, which separates the data plane from the control flat, LSO satisfies the demands of many applications and workloads (Fig. 2).

3.1 Network framework

This research assumes the presence of an external intruder, referring to an unauthorized user who lacks the necessary permissions to manage the sensor network. The invader intends to compromise the network's availability, but is unable to directly target the controller. To carry out a successful attack, the intruder has to acquire knowledge about the network's architecture and detect the prominent nodes that have significant involvement in network communication. These nodes include sink nodes, intermediary nodes, and shared nodes that handle both data and control traffic. Most communications are sent along routes that consist of nodes with significant visibility. This results in distinct traffic patterns that disclose information about the traffic paths, direction, and therefore the details of these nodes. To get unauthorized access, the intruder must initiate either a traffic analysis attack, a remote software-based, or a physical attack on the network. The intruder could seize control of sensor nodes, allowing them to access the flow table. This enables the intruder to passively monitor and intercept communications within the node's range, therefore disclosing some statistics about the surrounding area. The intruder can covertly acquire information about the network without rising suspicion, as the sensor node will maintain its usual behavior without engaging in any nefarious activities. The attacker is only able to

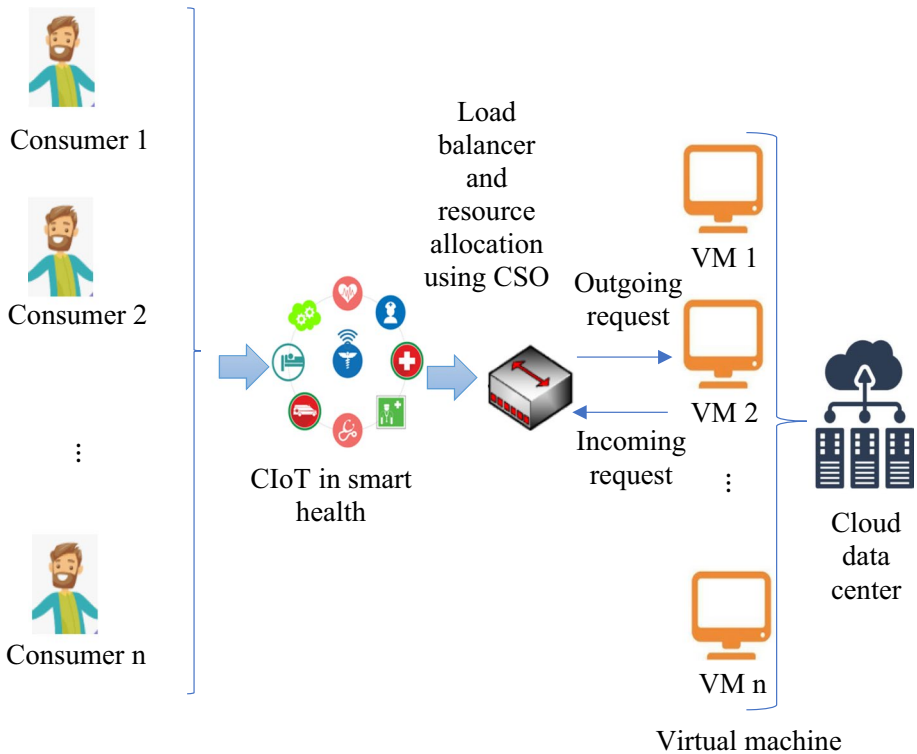


Fig. 2 Architecture for load balancing and resource allocation using LSO

compromise a limited number of nodes within an acceptable timeframe. This research assumes, without any loss of generality, that the intruder has the ability to compromise just one node within a control period Δt . As can be seen in Fig. 3, this architecture is divided into four separate layers: the data collection, the fog layer, the cloud layer, and the network protocol.

Data Collection Layer: This is the foundational layer in the proposed architecture. The data capture layer is responsible for the identification of actual objects and the collection of environmental and medical data from several devices that supply both actual and background information. The link between the devices and the communications system is established. In addition to the immediate data generated by this equipment, big data in health-care demands specialized management [33], as it necessitates extensive data analyses. This large amount of data includes medical files, standardized EHR, detailed medical records, and information from medical imaging. Depending on the information kind and treatment needs, the appropriate layer, whether fog or cloud, will receive health information.

Fog Layer details: The mobile cloud layer has been added to the model to facilitate the analysis of time-sensitive information. One of the primary motivators for the growth of the IoT is the capacity to review data "on the fly," provide real-time alerts, detect anomalies, and initiate the required actions immediately. The additional fog layer relocates computing power towards the edge, hence accelerating reaction times. In addition, it does filter, fusion, compression, consolidation, and intermediary data

3.2 LSO-based load balancing and resource allocation

Getting the schedule right requires an effective optimization technique. LSO’s usefulness comes from the fact that it can be used to optimize issues that occur in real time [33]. In addition, it is simple to implement due to the reduced number of parameters.

LSO The metaheuristic algorithm described by [34] was motivated by the hunting activities of a Lanner. The LSO, which includes a three-stage procedure that necessitates modifications to a variety of parameters, is a reliable approach for solving randomized population-based issues. The hunting strategy adopted by Lanners while they were flying in quest of their prey served as the model for the proposed method. Retired Lanners modify their hunting techniques to meet their unique dietary requirements. As a result, novel tactics develop, and unique models maintain assumptions about flying. Tucker asserts that lanners are the bird species with the best flying abilities. The appropriate objectives are evaluated at various levels of enhanced hunting to see if they go beyond what is possible in the air [35]. One of the fastest creatures on the earth, stoops have been seen to attain speeds of more than 200 mph (320 km per hour).

Lanners have several tiny tubercules in their beaks that allow them to breathe easily. These control how the air is directed through fast stoops. The majority of hunting occurs daily (including morning and night). Smaller and medium-sized birds make up the majority of their prey, although they will also consume insects, including grasshoppers, worms, swarms, and crickets [35]. To reach its prey, the lanner takes a variety of flight paths. Each route has two parts: a straight section where the lanner keeps flying and aims when it is in its line of sight and a logarithmic spiral where the lanner keeps its head straight and its eyes fixed on the prey with exceptional accuracy. This makes it possible to divide the method by which a Lanner accomplishes movement into three steps. A graphic representation of a Lanner’s flight route during a hunt is shown in Fig. 4.

During transmission, minimizing both the long delay (\mathbf{td}) and the packet loss rate (\mathbf{pdr}) will improve resource allocation. It involves load balancing and the best resource allocation. Suppose the processing device’s capacity is \mathfrak{D} . In that case, the j th user’s resource allocation needs are described as $\langle \mathbf{td}_j, \mathbf{pdr}_j \rangle$, and the corresponding resource demand is $\langle \mathfrak{B}_j, \mathfrak{s}l_j \rangle$, where \mathfrak{B}_j and $\mathfrak{s}l_j$ indicate the j th CIoT user node’s bandwidth need and safeguard length demand, respectively. When the average buffering length AQ_j is smaller than the required safeguard length $D\mathfrak{s}l_j = (\mathfrak{s}l_j/\mathfrak{P}_{size})$, where \mathfrak{P}_{size} is packet size, and $D\mathfrak{s}l_j$ is, then \mathbf{pdr}_j may be computed as

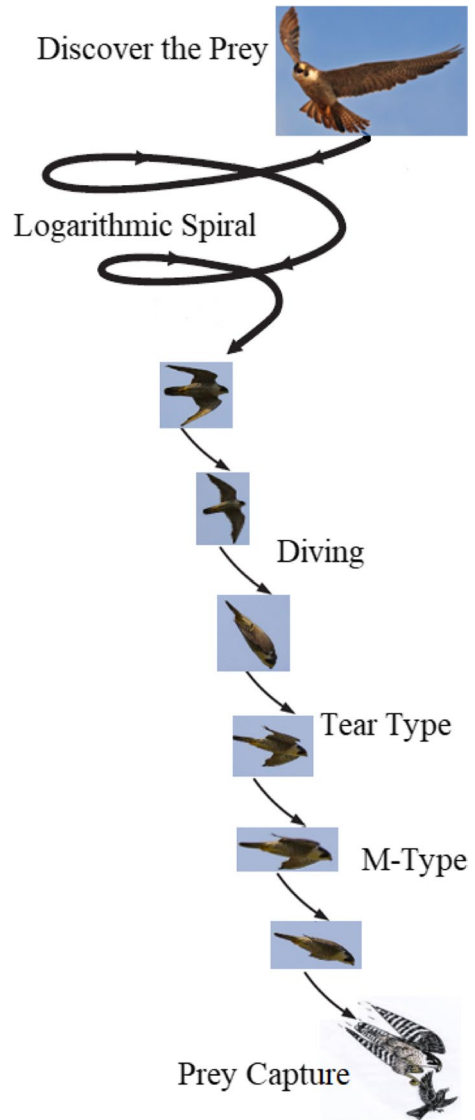
$$\mathbf{pdr}_j = \frac{AQ_j - D\mathfrak{s}l_j}{AQ_j} \tag{1}$$

The scheduler chooses several links to distribute the traffic, and the load distribution problem may be stated as,

$$\max \left(\mathbf{tp}, \frac{AQ_j}{AQ_j - D\mathfrak{s}l_j} \right) \cdot \sum_{k=1}^K \beta_k \tag{2}$$

where \mathbf{tp} represents throughput, β_k is the percentage of the k th link’s allotment based on the kind of traffic. In the provided scheduling issue, Lanners are produced at random for

Fig. 4 The schematic representation of a Lanner flight route during a prey hunt



allocating classified traffic to random links, and the fitness is assessed by considering Eq. (2). Below is the short approach for implementing the LSO, which consists of five steps [36].

Step 1: Starting with the optimization problem's parameters, such as the number of Lanners (N_L), maximum speed (\mathfrak{B}_{max}), cerebral pace ($C\tau$), socializing tenacious ($S\mathfrak{p}$), ensuing consistent ($\mathfrak{C}\mathfrak{p}$), diving rate ($\mathbb{D}\mathfrak{p}$), and awareness probability, the method is then initialized ($\mathbb{A}\mathfrak{p}$).

Step 2: Based on the boundary constraints, randomly determine the speed and location of the Lanners in a D-dimensional space, with each lanner's position taking the total number of NP applicants into account throughout all of the space's D dimensions. The

\mathfrak{V}_{min} and \mathfrak{V}_{max} limitations, which were each defined in the following equations, were used arbitrarily to obtain the speeds:

$$\mathfrak{V}_{max} = 0.1 \times \mathfrak{ub} \tag{3}$$

$$\mathfrak{V}_{min} = -\mathfrak{V}_{max} \tag{4}$$

where \mathfrak{ub} stands for the upper bound, which is the region around all dimensions boundary, creating the pairs of integers $(\mathcal{G}\mathbb{A}\mathbb{p}, \mathcal{G}\mathbb{D}\mathbb{p})$ for all lanners connection between the dive and awareness probabilities at random first.

Step 3: Choose the global (g_{best}) and best (x_{best}) sites after computing the value of fitness. New locations will be generated from the selected ones using the logic that governs the action of diving and the probability of being aware of it.

Step 4: Along with creating new sites, the lanner’s location has also been updated. Afterward, compare $\mathcal{G}\mathbb{A}\mathbb{p}$ with the likelihood of cognizance $\mathbb{A}\mathbb{p}$; if $\mathbb{A}\mathbb{p} > \mathcal{G}\mathbb{A}\mathbb{p}$, the Advice or assistance switches from hunting for prey relying on its activities and some other memories from other Lanners:

$$X_{it+1} = X_{it} + \mathfrak{V}_{it} + Cr(X_{best}, X_{it} + Sp(g_{best}, X_{it})) \tag{5}$$

where \mathfrak{V}_{it} represents the current speed, and X_{it} represents the lanner’s current location ($Cr = Sp = 1.5$). If $\mathcal{G}\mathbb{A}\mathbb{p}$ is greater than $\mathbb{A}\mathbb{p}$, then first compare $\mathcal{G}\mathbb{D}\mathbb{p}$ to the dive likelihood $\mathbb{D}\mathbb{p}$. If $\mathbb{D}\mathbb{p} < \mathcal{G}\mathbb{D}\mathbb{p}$, the Lanner (X_{ζ}) completes its first phase in the hunting process by selecting one of the objectives as its ζ victim. A polynomial spiral is offered.

$$X_{it+1} = X_{it} + |X_{\zeta} - X_{it}|^2 \cdot \exp^{f\alpha} \cos(2\Pi\alpha) \tag{6}$$

where f is a constant that causes the spiral logarithm’s state to be 1, and α is an arbitrary value between (-1,1) that specifies the lanner’s following location about its precise destination. In the event when $\mathbb{A}\mathbb{p} > \mathcal{G}\mathbb{A}\mathbb{p}$, then particular of the chosen prey will first be compared to the system equation of the lanner. When it comes to a diving step, the lanner will follow through with it everywhere the prey is most suitable, and this includes:

$$X_{it+1} = \begin{cases} X_{it+1} = X_{it} + \mathfrak{V}_{it+1} + r(\mathcal{G}\mathbb{P}(X_{\zeta}, X_{it})) & \mathbb{A}\mathbb{p} > \mathcal{G}\mathbb{A}\mathbb{p} \\ X_{it+1} = X_{it} + \mathfrak{V}_{it+1} + r(\mathbb{D}\mathbb{P}(X_{best}, X_{it})) & \text{otherwise} \end{cases} \tag{7}$$

In terms of the velocities and geographical boundaries, the newly discovered place will be examined in the future. Following this, its new scoring function is constructed, and the multiple features of X_{best} and g_{best} are figured out.

Step 5: In the last step, additional assessments of Step 4 are kept going until the maximum number of possible iterations (itermax) is achieved.

Figure 5 and Table 1 exhibit, respectively, the flowchart and pseudocode of LSO-based wealth distribution and load balancing.

4 Experimentation results and discussion

The advantages and viability of the proposed fog-cloud CIIoT architecture are shown through an example model. It considers 100 CIIoT nodes to gather loss- and delay-sensitive medical data from diverse residences or hospitals. Five edge devices, 1 server, and a 54 Mb/s

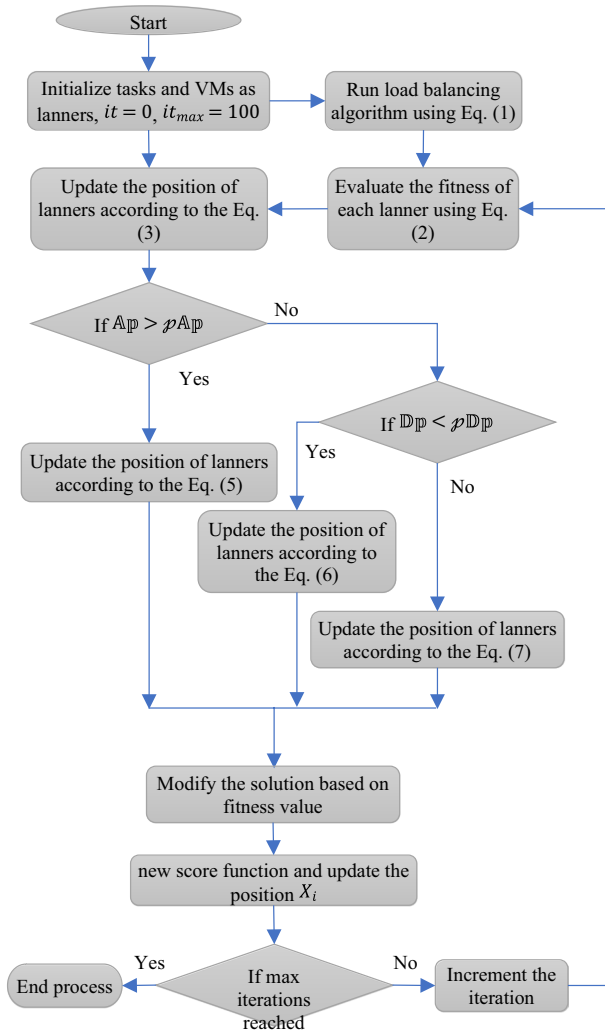
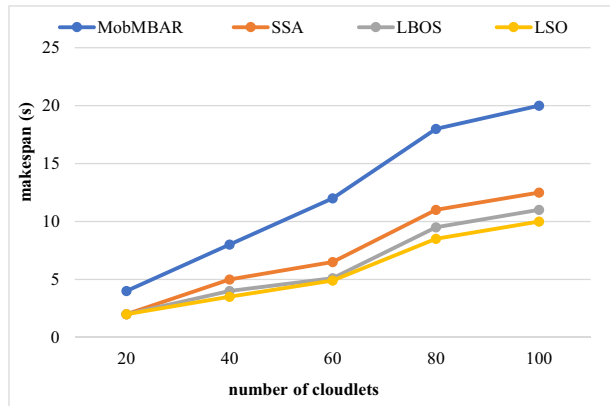


Fig. 5 Flowchart of LSO-based resource allocation and load balancing

connection data rate are present. Depending on the kind of traffic, created data is sent to the cloud for preprocessing before being offloaded to the cloud. Additionally, it is assumed that the gateways' and router's connection bandwidths are allocated randomly. The LSO methodology is used to choose the fog nodes based on buffer demand, throughput, and processing latency. Results are compared to current techniques such as MobMBAR [22], SSA [19], and LBOS [23] algorithms for the metrics makespan, response time, resource utilization, execution time, latency, throughput, and delivery rate.

Table 1 Algorithm of lanner optimization algorithm based resource allocation and load balancing

1. INPUT: datasets Population size, parameters of LSO. OUTPUT: Resource allocation and load balancing of CIoT
2. Initialized empty structure and initialized parameters of LSO algorithm.
3. For each lanner, create an N-dimensional vector $VM = \{VM_1, VM_2, \dots, VM_n\}$, where VM_j ($j \in \{1, 2, \dots, n\}$) indicates the total VMs on the task T_i ($i \in \{1, 2, \dots, n\}$) which is to be processed.
4. Compute PDR using Equation (1)
5. Establish position and velocity for each Lanner's arbitrarily, compare all lanners by score functions, and discover the best in the present location.
6. For loop to maximum iterations number.
7. For loop to population size.
8. Produce random values pAp , pDp . Choose the new best positions by correlating the score functions of every lanner.
9. if $Ap > pAp$, update lanner velocity with Equation (5); else
10. if $Dp < pDp$, update lanner velocity with Equation (6). Else, equivalence is the score functions of the previous and current one. If this one is healthier, update Lanner velocity using Equation (7); if not, use Equation (8).
11. Update position X_i .
12. Lanner optimization algorithm for load balancing with fitness function using eq. (2)
13. Evaluate the groove function of the new position, and the best score values and solutions are stored for resource allocation.
14. If max iteration is met, then end the iteration process, and the system is balanced, results are obtainable. If not, go to Step 5.
15. Return resource allocation and load balancing.

Fig. 6 Makespan comparison results**Table 2** Makespan comparison results

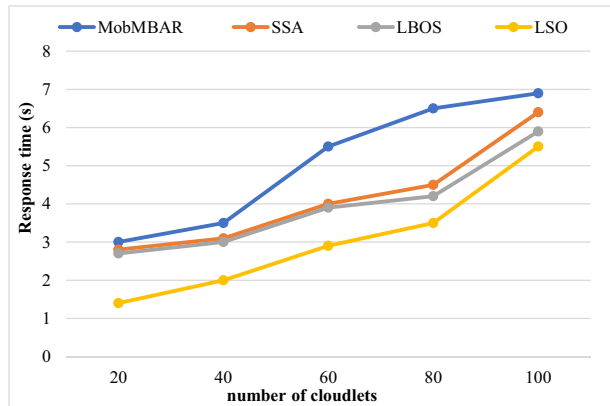
Number of cloudlets	Makespan (s)			
	MobMBAR	SSA	LBOS	LSO
20	4	2	2	2
40	8	5	4	3.5
60	12	6.5	5.1	4.9
80	18	11	9.5	8.5
100	20	12.5	11	10

4.1 Makespan comparison results

In terms of makespan, the research model is contrasted with the other current methods mentioned above. The research model exhibits a superior result for equally distributing the load across nodes, as shown in Fig. 6 and Table 2, which also illustrates the acquired values of the resource usage for several cloudlets for the various methods with the research model. The outcomes demonstrate that the research model responds faster than alternative algorithms. The simulation findings show that when the number of cloudlets increases, the effectiveness of other comparison methods on makespan declines. However, the research model outperforms the comparison well. The proposed research model, the LSO technique, performs better because it can use resources effectively by distributing loads across the appropriate VMs.

4.2 Response time comparison results

The response times of the MobMBAR, SSA, LBOS, and LSO techniques in various cloudlets are shown in Fig. 7 and Table 3. For the sake of the simulation, cloudlets are considered autonomous and non-pre-emptive. The research model is used to schedule separate jobs dynamically. The size of the cloudlets has an impact on the response time. According to the data, LSO has a substantially faster reaction time than MobMBAR, SSA, and LBOS, which consume energy at rates of 6.9 s, 6.4 s, and 5.9 s, respectively. Additionally, Table 5

Fig. 7 Response time comparison results**Table 3** Response time comparison results

Number of cloudlets	Response time (s)			
	MobMBAR	SSA	LBOS	LSO
20	3	2.8	2.7	1.4
40	3.5	3.1	3	2
60	5.5	4	3.9	2.9
80	6.5	4.5	4.2	3.5
100	6.9	6.4	5.9	5.5

shows that SSA, LBOS, and MobMBAR all have longer reaction times. Based on the LSO's outstanding load balancing and quicker reaction times, Table 3 compares resource allocation efficiency. However, as the quantity of cloudlets increases, LSO is shown to be more efficient than MobMBAR, SSA, and LBOS. The average percentage of reaction time is maximum during the cloudlets due to a slightly increased PDR, and the truth is that response time employed in the early phases of LSO with a unique data transmission stage is regarded as overhead energy. The research model makes use of a suitable load-balancing method to distribute cloudlets onto virtual machines.

4.3 Resource utilization comparison results

The resource requirements of the algorithms mentioned above are compared to those of the research model. The research model exhibits a more significant benefit for distributing the load across nodes equally, as shown in Fig. 8 and Table 4, which also depicts the acquired estimates of the resource usage for a quantity of cloud data centers for the various methods with the research model. The outcomes demonstrate that, in comparison to other algorithms, the research model has superior resource consumption. The simulation findings show that when the number of cloudlets grows, other comparison methods perform better in terms of resource consumption. The research model can effectively use the resources by distributing the loads within the appropriate VMs using the LSO; the LSO algorithm performs better. Due to the decrease in resource use and the heuristic information employed

Fig. 8 Resource utilization comparison results

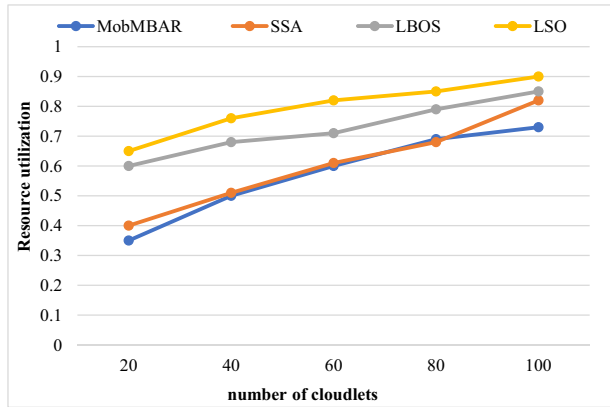
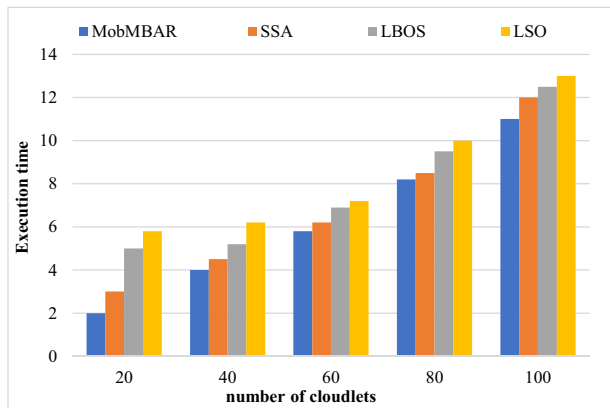


Table 4 Resource utilization comparison

Number of cloudlets	Resource utilization			
	MobMBAR	SSA	LBOS	LSO
20	0.35	0.4	0.6	0.65
40	0.5	0.51	0.68	0.76
60	0.6	0.61	0.71	0.82
80	0.69	0.68	0.79	0.85
100	0.73	0.82	0.85	0.9

Fig. 9 Execution time comparison results



in the algorithm, the productivity of resources is considerably exploited in the condition of resource utilization.

4.4 Execution time comparison results

Comparisons are made between the research model and the existing methods in terms of runtime. Thus, according to Fig. 9 and Table 5, the research model demonstrates a

Table 5 Execution time comparison

Number of cloudlets	Execution time (s)			
	MobMBAR	SSA	LBOS	LSO
20	2	3	5	5.8
40	4	4.5	5.2	6.2
60	5.8	6.2	6.9	7.2
80	8.2	8.5	9.5	10
100	11	12	12.5	13

superior result for uniformly distributing the load over all the nodes. This figure and table also indicate the acquired estimates of the completion time for a handful of virtual machines for the various methods using the research model. The outcomes demonstrate that the research model executes more quickly than alternative methods. The simulation findings show that as the number of cloudlets increases, other comparison algorithms perform better in terms of execution time. It demonstrates how well the research model distributes the loads within the VMs and how much the level of instability was diminished.

4.5 Latency comparison results

The research model achieves optimum resource usage while reducing delay, as seen in Fig. 10. The current approach can use resources to their fullest extent and dramatically increases delay. As a result, LSO thought about using a task-resource suitability test to assess how well the assignments for the overcrowded VMs and the underused VMs were compatible, given their respective resource availability. According to data, LSO has a substantially lower latency than MobMBAR, SSA, and LBOS, which have energy consumption values of 20 ms, 12.5 ms, and 11 ms, respectively. Table 6 analyses latency efficiencies depending on how well the LSO balances loads while requiring less response time.

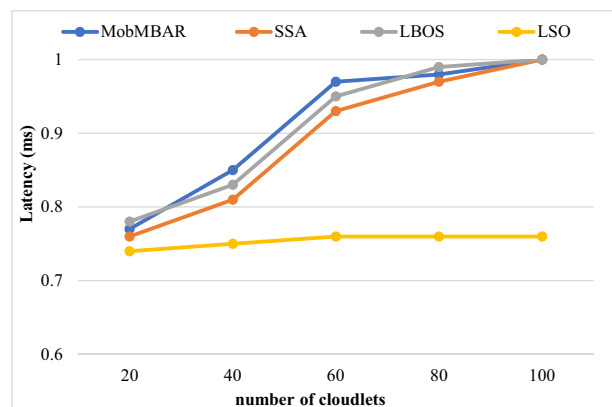
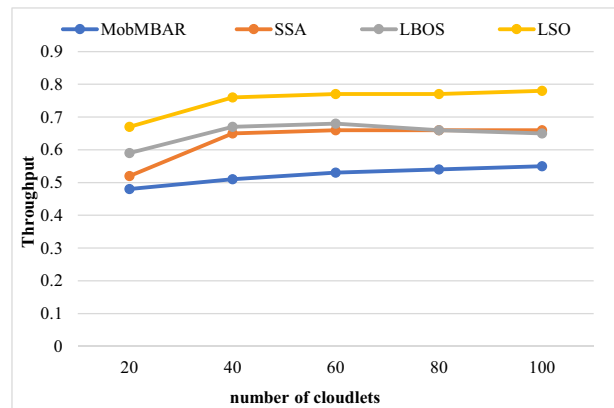
Fig. 10 Latency comparison results

Table 6 Results of latency comparison

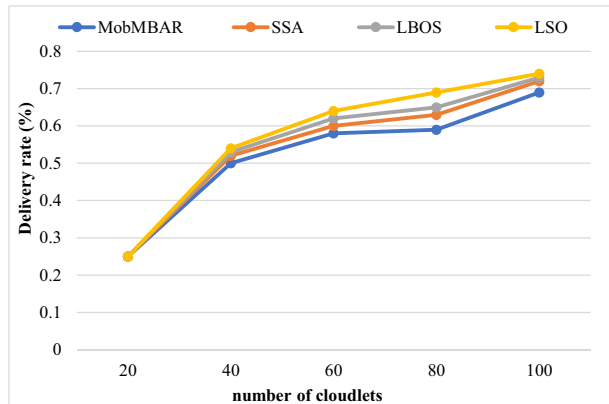
Number of cloudlets	Latency (ms)			
	MobMBAR	SSA	LBOS	LSO
20	4	2	2	2
40	8	5	4	3.5
60	12	6.5	5.1	4.9
80	18	11	9.5	8.5
100	20	12.5	11	10

Fig. 11 Throughput comparison results**Table 7** Results of throughput comparison

Number of cloudlets	Throughput (s)			
	MobMBAR	SSA	LBOS	LSO
20	0.48	0.52	0.59	0.67
40	0.51	0.65	0.67	0.76
60	0.53	0.66	0.68	0.77
80	0.54	0.66	0.66	0.77
100	0.55	0.66	0.65	0.78

4.6 Throughput comparison results

Figure 11 contrasts and compares the performance of the existing models and the proposed research model in a standardized environment. The research model has a high throughput for a clustering approach, whereas the existing systems, MobMBAR, SSA, and LBOS, have durations of 0.55, 0.66, and 0.65, respectively, when the size of the cloudlets is 100. The results show that the performance in LSO, even during the initial phases of cloud data centers, is relatively low. In Table 7, throughput efficiencies are compared depending on how well the LSO balances the load while requiring less reaction time.

Fig. 12 Delivery rate comparison results**Table 8** Delivery rate comparison

Number of cloudlets	Delivery rate (%)			
	MobMBAR	SSA	LBOS	LSO
20	0.25	0.25	0.25	0.25
40	0.50	0.52	0.53	0.54
60	0.58	0.60	0.62	0.64
80	0.59	0.63	0.65	0.69
100	0.69	0.72	0.73	0.74

4.7 Delivery rate comparison results

The research model is contrasted with the other current algorithms mentioned above regarding the delivery ratio. As shown in Fig. 12 and Table 8, the research model achieves a superior result for uniformly distributing the load across nodes and depicts the achieved values of the download speed for several cloud servers for the various methods. The outcomes demonstrate that the research model outperforms existing algorithms in terms of delivery rate. The simulation findings show that when the number of cloudlets rises, other comparable algorithms perform better in terms of delivery rate. The buffer size contributes to a reduction in packet drops, which improves the performance of the proposed LSO method. The fog layer's packet delivery rates decrease when the storage size is large. Based on the LSO's outstanding load balancing and shorter reaction times, Table 8 examines delivery ratio efficiency.

The dynamic LSO algorithm gives the proposed research better results. It balances exploitation and exploration with dynamic inertia weight modifications to distribute effort efficiently. The adaptable algorithm optimizes resource allocation for system efficiency by considering real-time situations and varied traffic types. VM accomplishment time variation is minimized by the LSO algorithm, improving load balancing delay, throughput, and network congestion measures. The algorithm's responsiveness and optimization help explain its experimental outperformance.

4.8 Advantages & limitations

The proposed research has several advantages. First, the LSO-driven cloud-assisted framework improves resource allocation and workload distribution in CIoT applications for smart healthcare systems. Scalability, power consumption, and data processing increase with this modification, addressing crucial field concerns. The innovative LSO algorithm balances exploitation and exploration with dynamic adaptation, outperforming evolutionary and heuristic methods. The load balancing strategy developed in this research improves latency, throughput, and network congestion measures, improving system dependability and responsiveness. The framework's extensive performance comparison against existing methodologies shows its efficacy and originality, demonstrating its potential influence on healthcare-oriented IoT.

Despite substantial advantages, the research has few limitations. The healthcare setting may affect the framework's adaptation and efficacy. Network changes and hardware limits may affect generalizability. The framework's real-world deployment and scalability need more study to determine its practicality in various healthcare contexts and workloads. With every algorithmic approach, computational cost and overhead must be addressed, especially when using LSO in resource-constrained contexts. Technological advances and changing healthcare needs may demand ongoing updates or revisions to the suggested framework to be relevant and successful in dynamic CIoT environments.

5 Conclusion and future work

This work developed the LSO method, which examines the learner's positions at the beginning and updates them continuously during each iteration. Utilizing a suitable fitness function may reduce the makespan and maximize resource consumption. A task similarity test has been conducted to evaluate how well tasks on overloaded VMs work with resources on underloaded VMs. It has been evaluated and contrasted with other current algorithms. Additionally, in any cloud context, the research model can handle independent, pre-emptive, and non-pre-emptive activities. Future solutions to the load-balancing issue with diverse resources could be based on meta-heuristics. Applying a growing number of tasks and VMs in a diverse environment might solve this issue. To verify the success of the method, additional QoS performance measures may also be taken into account. Results are compared to current techniques such as MobMBAR, SSA, and LBOS algorithms, where the research model attains a makespan of 10 s, response time of 5.5 s, resource utilization with a rate of 0.9, execution time of 13 s, latency of 10 ms, throughput with 0.78 s, and delivery rate with 0.74%. However, it is important to recognize its limitations, such as the impact of unique healthcare contexts, external influences, computational complexity, and the need for constant changes to changing technology landscapes. The inherent variability in healthcare environments, external factors affecting experimental outcomes, real-world deployment challenges, the computational complexity of the LSO algorithm, and evolving healthcare technologies can all threaten validity. To assure the framework's stability and applicability in varied CIoT applications in smart healthcare systems, diversified testing, continual improvement, and rigorous validation are needed.

Despite these limitations, this research provides a potential path for improving CIoT application efficiency in smart healthcare systems, providing the platform for future study.

In future, the DL and Reinforcement Learning (RL) together can be developed as an effective strategy for long-term success with maximum efficiency. Because DL takes a smart approach, RL can figure out the optimal reward strategy for carrying out a task. Complex patterns in the dataset contribute to the misclassification issue, and RL will learn this pattern without the need for feature engineering and will automatically solve it, setting up an effective classification for a DL method.

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

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References


1. Rose K, Eldridge S, Chapin L (2015) The internet of things: an overview. *Internet Soc (ISOC)* 80:1–50
2. Shah SH, Yaqoob I (2016) A survey: Internet of Things (IoT) technologies, application and challenges. 2016 IEEE Smart Energy Grids Engineering (SEGE). 381–385
3. Sharma N, Shamkumar M, Singh I (2019) The history, present and futures with IoT. In: *IoT and big data analytic for smart generations*. Springer, Cham, pp 27–51
4. Gulati AG (2021) Moving up global value chains to unleash the potential of an empowered digital India. Available at SSRN 3924041
5. Wood D, Apthorpe N, Feamster N (2017) Cleartexts data transmission in consumer IoT medical devices. In: *Proceedings of the 2017 workshops on IoT security and privacy (IoTS&P '17)*. Association for Computing Machinery, New York, pp 7–12. <https://doi.org/10.1145/3139937.3139939>
6. Garge GK, Balakrishna C, Datta SK (2017) Consumer healthcare: current trend in consumer health monitoring. *IEEE Consum Electron Mag* 7(1):38–46
7. Boric-Lubecke O, Gao X, Yavari E, Baboli M, Singh A, Lubecke VM (2014) E-healthcare: remote monitoring, privacy, and security. In: *2014 IEEE MTT-S international microwave symposiums (IMS2014)*. IEEE, pp 1–3
8. Gu D, Yang X, Deng S, Liang C, Wang X, Wu J, Guo J (2020) Tracking knowledge evolution in cloud health care research: knowledge map and common word analysis. *J Med Internet Res* 22(2):e15142. <https://doi.org/10.2196/15142>
9. Sivan R, Zukarnain ZA (2021) Security and privacy in cloud-based e-health systems. *Symmetry* 13(5):742
10. Isa ISBM, El-Gorashi TE, Musa MO, Elmighani JM (2020) Energy efficient fog-based healthcare monitoring infrastructures. *IEEE Access* 8:197828–197852
11. Opara AC (2022) Representing IoT, cloud and edge computing security and privacy policy and detecting potential problem. (Doctoral dissertation)
12. de Moura Costa HJ, da Costa CA, da Rosa Righi R, Antunes RS (2020) Fog computing in health: a systematic literature review. *Heal Technol* 10(5):1025–1044
13. Kraemer FA, Braten AE, Tamkittikhun N, Palma D (2017) Fog computing in healthcare—a review and discussions. *IEEE Access* 5:9206–9222
14. Li J, Cai J, Khan F, Rehman AU, Balasubramaniam V, Sun J, Venu P (2020) A secured framework for sdn-based edge computing in IOT-enabled healthcare systems. *IEEE Access* 8:135479–135490
15. Asghar A, Abbas A, Khattak HA, Khan SU (2021) Fog based architectures and load balancing methodology for health monitoring system. *IEEE Access* 9:96189–96200
16. Dubey K, Sharma SC, Kumar M (2022) A secure IoT application allocations framework for integrated fog-cloud environments. *J Grid Comput* 20(1):1–23
17. Talaat FM (2022) Effective predictions and resources allocation method (EPRAM) in fog computing environments for smart healthcare systems. *Multimed Tools Appl* 81(6):8235–8258
18. Ghanbari Z, JafariNavimipour N, Hosseinzadeh M, Darwesh A (2019) Resources allocation mechanism and approaches on the IoT. *Clust Comput* 22(4):1253–1282

19. Abdulhammed OY (2022) Load balancing of IoT task in the cloud computing by using sparrow search algorithm. *J Supercomput* 78(3):3266–3287
20. Kanbar AB, Faraj K (2022) Region aware dynamic tasks scheduling and resource virtualization for load balancing in IoT-fog multi-cloud environment. *Futur Gener Comput Syst* 137:70–86
21. Leontiou N, Dechouniotis D, Denazis S, Papavassiliou S (2018) A hierarchical control framework of load balancing and resource allocation of cloud computing services. *Comput Electr Eng* 67:235–251
22. Abdelmoneem RM, Benslimane A, Shaaban E (2020) Mobility-aware tasks scheduling in cloud-Fog IoT-based healthcare architecture. *Comput Netw* 179:107348
23. Talaat FM, Saraya MS, Saleh AI, Ali HA, Ali SH (2020) A load balancing and optimization strategy (LBOS) using reinforcement learning in fog computing environments. *J Ambient Intell Human Comput* 11(11):4951–4966
24. Meng Y, Zhang W, Zhu H, Shen XS (2018) Securing consumer IoT in the smart home: architectures, challenge, and countermeasure. *IEEE Wirel Commun* 25(6):53–59
25. Baho SA, Abawajy J (2023) Analysis of consumer IoT devices vulnerability quantifications framework. *Electronics* 12(5):1176
26. Harkin D, Mann M, Warren I (2022) Consumer IoT and its under-regulations: finding from an Australian study. *Policy Internet* 14(1):96–113
27. Verhoef PC, Stephen AT, Kannan PK, Luo X, Abhishek V, Andrews M, ..., Zhang Y (2017) Consumer connectivity in a complex, technology-enabled, and mobile-oriented world with smart product. *J Int Mark* 40(1):1–8
28. Olga GK, Sarmah DK (2022) The baseline of global consumer cyber security standard for IoT: quality evaluations. *J Cyber Secur Technol* 6(4):175–200
29. Poyner IK, Sherratt RS (2018) Privacy and security of consumer IoT device for the pervasive monitoring of vulnerable people. In: *Living in the internet of things: cybersecurity of the IoT-2018*, pp 1–5. <https://doi.org/10.1049/cp.2018.0043>
30. Ngwenya M, Ngoepe M (2022) Data trust in Consumer Internet of Things assemblages in the mobile and fixed telecommunications operators in South Africa. *S Afr J Inf Manag* 24(1):1426
31. Alladi T, Chamola V, Sikdar B, Choo KKR (2020) Consumer IoT: security vulnerability case studies and solution. *IEEE Consum Electron Mag* 9(2):17–25
32. Lee J, Ardakani HD, Yang S, Bagheri B (2015) Industrial big data analytics and cyber-physical system for future maintenances & services innovation. *Procedia CIRP* 38:3–7
33. Mishra K, Majhi SK (2021) A binary bird swarm optimization based load balancing algorithms for cloud computing environments. *Open Comput Sci* 11(1):146–160
34. Rini DP, Shamsuddin SM, Yuhaniz SS (2011) Particles swarm optimization: techniques, systems and challenges. *Int J Comput Appl* 14(1):19–26
35. Gowree ER, Jagadeesh C, Talboys E, Lagemann C, Brücker C (2018) Vortices enable the complex aerobatics of peregrine falcons. *Commun Biol* 1(1):1–7
36. de Vasconcelos Segundo EH, Mariani VC, dos Santos Coelho L (2019) Design of heat exchanger using Falcon Optimization Algorithm. *Appl Therm Eng* 156:119–144

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

V. Arulkumar¹  · M. Aruna² · D. Prakash³ · M. Amanullah⁴ · K. Somasundaram⁵ · Rajendran Thavasimuthu⁶

✉ V. Arulkumar
arulkumaran.ckpc@gmail.com

M. Aruna
arunam@srmist.edu.in

D. Prakash
prakash.dhanagopalsamy@gmail.com

M. Amanullah
amanhaniya12@gmail.com

K. Somasundaram
soms72@yahoo.com

Rajendran Thavasimuthu
rajendran.thavasimuthusamy@gmail.com

¹ School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamilnadu, India

² Department of Computing Technologies, Faculty of Engineering and Technology, College of Engineering and Technology, SRM Institute of Science and Technology, Kattankulathur, Tamilnadu, India

³ Department of Computer Science and Engineering, Muhammed Sathak A J College of Engineering, Chennai, India

⁴ Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, Tamil Nadu, India

⁵ Computer Science and Engineering, Sri Muthukumaran Institute of Technology, Kanchipuram Chennai, India

⁶ Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, Tamilnadu, India