Opportunities of IoT in Fog Computing for High Fault Tolerance and Sustainable Energy Optimization

A. Reyana

Sandeep Kautish

Khalid Abdulaziz Alnowibet

See next page for additional authors

Funder: The research is funded by the Researchers Supporting Program at King Saud University, (RSP2023R305).

Authors
A. Reyana, Sandeep Kautish, Khalid Abdulaziz Alnowibet, Hossam Zawbaa, and Ali Wagdy Mohamed

sustainability

Article

Opportunities of IoT in Fog Computing for High Fault Tolerance and Sustainable Energy Optimization

A. Reyana [1], Sandeep Kautish [2], Khalid Abdulaziz Alnowibet [3], Hossam M. Zawbaa [4] and Ali Wagdy Mohamed [5,6,*]

1 Department of Computer Science and Engineering, Karunya Institute of Technology and Sciences, Coimbatore 641114, Tamilnadu, India; reyareshmy@gmail.com
2 Department of Computer Science and Engineering, Lord Buddha Education Foundation, Kathmandu 44600, Nepal; dr.skautish@gmail.com
3 Statistics and Operations Research Department, College of Science, King Saud University, P.O. Box 2455, Riyadh 11451, Saudi Arabia; knowibet@ksu.edu.sa
4 CeADAR Ireland's Center for Applied AI, Technological University Dublin, D7 EWV4 Dublin, Ireland; hossam.zawbaa@gmail.com
5 Operations Research Department, Faculty of Graduate Studies for Statistical Research, Cairo University, Giza 12613, Egypt
6 Applied Science Research Center, Applied Science Private University, Amman 11937, Jordan
* Correspondence: aliwagdy@staff.cu.edu.eg

Abstract: Today, the importance of enhanced quality of service and energy optimization has promoted research into sensor applications such as pervasive health monitoring, distributed computing, etc. In general, the resulting sensor data are stored on the cloud server for future processing. For this purpose, recently, the use of fog computing from a real-world perspective has emerged, utilizing end-user nodes and neighboring edge devices to perform computation and communication. This paper aims to develop a quality-of-service-based energy optimization (QoS-EO) scheme for the wireless sensor environments deployed in fog computing. The fog nodes deployed in specific geographical areas cover the sensor activity performed in those areas. The logical situation of the entire system is informed by the fog nodes, as portrayed. The implemented techniques enable services in a fog-collaborated WSN environment. Thus, the proposed scheme performs quality-of-service placement and optimizes the network energy. The results show a maximum turnaround time of 8 ms, a minimum turnaround time of 1 ms, and an average turnaround time of 3 ms. The costs that were calculated indicate that as the number of iterations increases, the path cost value decreases, demonstrating the efficacy of the proposed technique. The CPU execution delay was reduced to a minimum of 0.06 s. In comparison, the proposed QoS-EO scheme has a lower network usage of 611,643.3 and a lower execution cost of 83,142.2. Thus, the results show the best cost estimation, reliability, and performance of data transfer in a short time, showing a high level of network availability, throughput, and performance guarantee.

Keywords: sustainable energy optimization; environment; wireless sensor networks; data processing; Internet of Things; fog computing; ant bee colony; particle swarm optimization

check for updates

Citation: Reyana, A.; Kautish, S.; Alnowibet, K.A.; Zawbaa, H.M.; Wagdy Mohamed, A. Opportunities of IoT in Fog Computing for High Fault Tolerance and Sustainable Energy Optimization. *Sustainability* **2023**, *15*, 8702. https://doi.org/10.3390/su15118702

Academic Editors: Irfan Ali, Umar Muhammad Modibbo, Syed Mithun Ali, Andrea Appolloni and Guilherme F. Frederico

Received: 2 March 2023
Revised: 14 May 2023
Accepted: 18 May 2023
Published: 27 May 2023

1. Introduction

Although the cloud can be used to perform computation and store sensor data, it is limited in terms of processing time. The cloud platform is centralized, but this causes performance limitations. Cloud computing uses a high-end infrastructure to provide high-speed services. Managing a pool of resources requires high energy efficiency; thus, the utilization of adequate resources is of major concern in cloud computing. The energy efficiency and sustainability of these resources remains a challenge. This requires service delivery at low power, as well as energy-aware resource scheduling and sustainability. A decentralized computing infrastructure where data, storage, computation, and applications are located elsewhere is considered fog computing. The fog system brings additional intelligence capability and improves efficiency. The fog is closer to the cloud, enabling short-term analysis, while fog computing provides bandwidth conservation, improves the response time, and is also network-agnostic. Despite these advantages, fog systems are tied to a specific physical location that comes into play with cloud computing. However, it is distributed and is located close to the client's device. The cloud takes a long time to communicate and is slow to respond. Today, sensor devices receive much research attention due to their varied applications. As a result of their data, a control center can make more informed decisions in the context of a forest monitoring application, which necessitates a large number of resources and computing paradigms that work in tandem to function efficiently. Fog computing technology in the IoT environment has attracted the interest of many researchers [1]. It forms one of the key elements of the Industrial Revolution 4.0. This is due to device-to-device connectivity, real-time data access, proactive fault tolerance, and failure management. As a result, data must be processed in real time for services to be delivered on schedule. The use of data centers for data processing is not feasible as they can result in propagation delays, congestion, high execution costs, and slow delivery times. Fog computing is one of the models used to overcome these limitations. The sensor nodes can be distributed in a heterogeneous manner, in terms of networking standards and processing speeds. Resource management, energy optimization, and responsiveness must be efficient for a real-time catastrophic application [2]. Sensor devices collaborate and interact with one another to achieve a common goal. In this regard, fog computing has had a significant impact [3]. Hence, sensor devices, with their capacity to execute multiple tasks concurrently, have grown in popularity over the years. This has significantly aided in the adoption of WSNs in applications that support human operations. Because of latency, the quality of service suffers as a result of unstable network connectivity [4]. In fog computing, the fog serves as an interface between sensor devices and the cloud, reducing request-response times. It can support a large geographical area, increasing location awareness, streaming real-time applications, and providing the primary functions of wireless sensor networks [5]. The characteristics of fog computing have encouraged its utilization in pervasive health monitoring, distributed computing, etc. In most cases, sensor data are saved on the cloud server for later processing. In practice, fog computing makes use of end-user clients and neighboring edge devices to perform computation and communication [6]. This enables fog computing to support WSN applications in the context of time-sensitive transmission, identifying closer heterogeneous devices such as PCs, gateways, data centers, and servers [7]. Previously, data processing relied on the cloud. However, issues such as device heterogeneity, privacy, security, and bandwidth continue to pose a challenge when conquering new frontiers [8]. This paper aims to develop a quality-of-service-based energy optimization (QoS-EO) scheme for wireless sensor environments, deployed via fog computing. The original schemes of ant and bee colonies [9] and particle swarm optimization [10] are hybridized to derive two novel techniques, (i) checkpointing (CP), and (ii) particle-bee optimization. The implemented techniques offer services such as (i) energy optimization, making the system highly fault-tolerant, (ii) good handling quality of the service placement of multiple data centers positioned at different locations, (iii) reducing the packet transmission time using the CP technique, and (iv) identifying the best cost path to perform network energy optimization. In the rest of this article, Section 2 reviews the various IoT-fog-related techniques, and Section 3 describes in detail the proposed technique, the QoS-EO scheme. Section 4 discusses the results gained from using the proposed technique. Finally, our conclusions are presented in Section 5.

2. Related Work

Sustainable energy optimization has attracted many researchers from different research areas to accomplish this goal. Many researchers have presented various insights into fog computing, only a few of which are discussed in this section. Varghese et al. [11]

examined the feasibility of fog computing. User data in large volumes increase the frequency of communication issues and communication latency, which vary depending on the geographical distances between devices. This has an impact on the quality of the user experience. The authors emphasized the viability of computing in the context of an online game for location awareness. Their model improved the average response time by 20%, while reducing data traffic between the remote server and the user device by 90%, thereby improving the QoS. Mohammed et al. [12] discussed fault tolerance and reliability issues in the context of IoT applications, which is similar to the research currently under consideration. Emerging smart-city projects are centered on using the IoT with fog computing. The work herein presented takes into account the failures of fault tolerance due to connectivity, hardware malfunctions, power outages, and cyber-attacks. To avoid operational disruptions and traffic problems, the aforementioned failures must be addressed for the fog nodes to maintain adequate operations. A combination of various fog services was identified in the work presented by Gill and Singh [13], who proposed an ant colony optimization-based optimal container placement method. It is well known that virtual machines are an essential component of data centers; however, containers are a new class of virtualization that represents a novel operating system. Containers are portable and lightweight, but their placement is a challenge, facilitating resource isolation. Dynamic optimization can thus benefit from ant colony optimization techniques. The presented work explores optimal make-span tasks in the context of virtualization, demonstrating better resource utilization. A major consideration in this article is the mentioned future scope of improving resource utilization and energy efficiency in fog devices using particle swarm optimization. Gong and Zu [14] compared the flaws of different resource allocation algorithms. Min–min, max–min, genetic, and ant colony algorithms are used for the methods under consideration. For initial population calculation in a fog environment, the algorithms were simulated in the 3.0 cloud simulation platform. The simulation results demonstrated that the ant colony and genetic algorithms perform well in fog calculations. Wang et al. [15] proposed the MAC-GAC algorithm for matching spatial tasks and online assignment tasks. Using MQC-GAC, a combination of genetic and ant colony algorithms, the author created a mechanism for credibility and punishment. The task assignment calculates the quality of the candidates for the workspace and assigns the worker credibility upon completion. High-quality workers are motivated by rewards; otherwise, their remuneration is reduced. As a result, MQC-GAC optimizes the assigned task. Martinez et al. [16] developed the design and dimensioning of fog infrastructures to provide services to the IoT traffic network. Despite the design, it was not fault-tolerant. To overcome the above issue, reliable and fault-tolerant standby fog nodes were activated with any fog node failure. For this purpose, a mixed-integer linear program (MILP) was applied. Thus, a column-generation approach was used to increase the scalability, with little loss to the optimal design. However, a fault tolerance mechanism with a reduced cost remains a challenge. Mohammad et al. [17] adopted an intelligent strategy to eradicate the interference of defective nodes. Their work extended the fog competence into the SIoT to empower resource-poor objects for handling intensive tasks. A Markov model was employed for fault diagnosis in the fog node. Based on the type of faults and threshold, reliability, availability, and average time to failure were measured. Although the scheme increases the number of live nodes, improves detection accuracy, and reduces invalid requests, network reliability remains a challenge. Ahmed et al. [18] focused on improving communication in IoT-based vehicle networks for increasing the overall system performance. A software-defined fault tolerance and QoS-aware IoT-based vehicular network using edge computing secured by blockchain was utilized to reduce the overall communication delay and message failure fault tolerance and to secure service provisioning for VANET ad hoc networks. The model received the vehicle messages through software-defined network nodes to provide secure services to the vehicles. Once the message was delivered to its destination, a fault tolerance mechanism checked the acknowledgments. Upon delivery failure, it sent the failed message again. The result shows a performance improvement, with a communication delay of only 55%. Therefore, improving execution time, security risk, and message failure ratios remains a challenge.

3. Research Methodology

Obtaining accurate real-time environmental monitoring data is a complex task. The development of IoT and fog systems has improved computing services. The challenges of environmental monitoring include the spatial distribution of devices, communication cost limitations, real-time monitoring, and the analysis of massive amounts of data. As a result, it is difficult to develop an accurate environmental monitoring system using data collected from various regions. This article describes a WSN-based collaborative mechanism employed in conjunction with fog computing. The QoS-EO scheme is made up of three layers: sensing, network, and service. The messages and environmental alerts are provided by the service layer [19]. In the network layer, fault tolerance necessitates handling communication failure. Fault tolerance in WSN is a challenging task. From the different links, the cumulative fault tolerance is analyzed, as expressed in Equation (1) [20]:

$$F_{i,j} = \left(1 - \sum_{t=0}^{M.e} \left(er_{\ i,j}\right)^t \left(1 - er_{i,j}\right)\right) + d_{i,j} \tag{1}$$

where M_{re} is the successful transmission-based count of retransmission, $er_{i,j}$ is the error rate, $d_{i,j}$ is the links degree estimation and $_{i,j}$ are the nodes.

The degree estimation for the nodes is expressed in Equations (2) and (3):

$$d_{i,j} = \left\{1, d_i = d_j = s_{n-1}\right\} \tag{2}$$

$$d_{\ i,j=\{} \begin{matrix} 1 - a^{di} di = dj < s_n - 1 \\ 1 - a^{\frac{(di-dj)^2}{di+aj}}, |di - dj| > 0 \end{matrix} \tag{3}$$

The degrees of i and j are d_i and d_j, and the α decision variable ranges between zero and one.

The delay is determined by the link quality expressed in Equation (4) [20]:

$$D_{i,j} = \left(1 - \sum_{t=0}^{Mre} \left(er_{i,j}\right)^t \left(1 - er_{i,j}\right)\right) + \frac{d_{i,j}}{P_s} + \frac{s_p}{T_s}. \tag{4}$$

The devices compromise the QoS regarding high latency, leading to unstable network connectivity. QoS affects time-sensitive functions as the data must be backed by cloud computing. This technical gap needs to be managed in the growing wireless sensor network. The paper thus focuses on enhancing the QoS elements from the perspective of wireless sensor networks, in collaboration with fog computing [21]. The physical layer in the wireless sensor networks consists of multiple sensors that gather real-time environmental data, as shown in Figure 1.

The sensed data are dispatched to the various meteorological stations using the internet through the gateway layer. Real-time activities are managed and communicated via the service layer. This layer sends alerts to the real world via mobile applications, web apps, etc. [22]. Thus, it improves the quality of life, enhancing urbanization, and handling natural and man-made disasters. The fog layer enables the use of computing services in the context of emphasizing proximity to end-users, thus performing real-time processing, intelligence awareness with end-to-end communication, network efficiency, scalability, and agility for faster and less expensive computation [23]. The fog nodes are deployed in specific geographical areas that cover the sensor activity performed in those areas. The logical situation of the entire system is informed by the fog nodes, as portrayed.

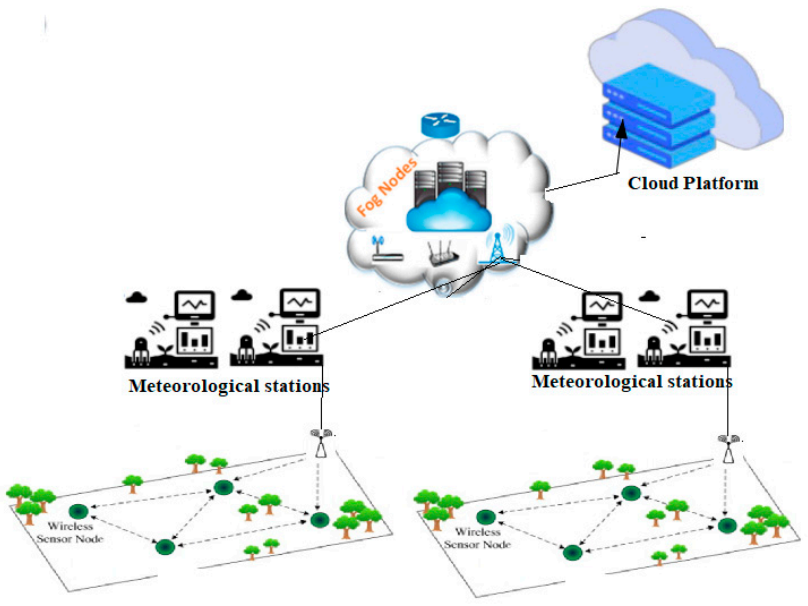

Figure 1. The system model of the proposed QoS scheme.

The QoS-EO framework has 3 layers of sensor nodes; each layer has sensor devices deployed in the physical environment that record the observations and perform sampling of the measurement data to send it to the sink node. Furthermore, data are transmitted to the fog computing node. Here, there is a combination of various fog nodes that are fixed in specific regions. The alert services sent to the nearby regions are created in the fog layer by establishing connections with neighboring nodes. The observations obtained from the sensor nodes are time-correlated. Hence, the fog nodes analyze the time correlation of the received data from each region and establish a prediction for environmental awareness. If it is within the tolerance, it is considered to be longer time, resulting in unnecessary energy consumption upon transmission [24]. While exceeding the threshold range, the actual measurements are sent to the fog node [25]. Thus, the fog nodes forward the predicted value to the cloud. With the use of homologous-type sensors, redundancy is avoided. These sensors are usually deployed in the same region [4]. The time-series data are collected by the cloud from the fog [26]. The QoS-EO factors analyzed are, namely, throughput, response time, resource utilization, cost, execution time, energy consumption, reliability, availability, and scalability [27]. In the considered application it performs a checkpointing (CP) technique for evaluating the network performance; the best cost estimation is performed using particle bee optimization. This is followed by an analysis of QoS-EO-based service placement delay [28]. The mandatory aspect of fog computing is fault tolerance. This can be either proactive or reactive. Anticipating failure and scheduling before the service request is considered proactive behavior. Here, the status of the cloud is continuously monitored. Once a failure occurs, the reaction is reactive. The proposed study follows reactive fault tolerance using the checkpointing technique, wherein the level of fault tolerance is set to its resistance to sensor defects. The checkpointing technique saves the state of execution, limiting the computation loss due to failure. In the case of the global checkpoint, the application entity synchronizes with the global checkpoint [29]. On recovery, the logged messages and recent checkpoints reach a pre-failure state [30]. Here, the fault-tolerant parameter is the region coverage factor, c, as expressed in Equation (5):

$$c = c_k - c_c \tag{5}$$

where c_k represents fault detection accuracy and $\mathrm{c_c}$ represents the probability of an unsuccessful replacement of the identified faulty sensor with a good spare sensor. The calculation of c_k is shown in Equation (6):

$$c_k = \frac{(k(1-p))}{k^{\left({}^{k/}_{M(p)}\right)} 1/^{M(p)} + (1 - \frac{k}{M(p)})k} \tag{6}$$

where $c_k \leq 1$ and $M(p)$ is a function of p representing an adjustment parameter [31], while p is the cumulative probability of sensor failure expressed in Equation (7).

$$F_s(t_s; \gamma_s) = p = 1 - exp^{(-\gamma ts)} \tag{7}$$

Solving Equation (7) for λ_s, the failure rate over the period t_s is given in Equation (8):

$$\gamma_{s=-(\frac{1}{t_s})\ln(1-p)} \tag{8}$$

Fault tolerance in fog computing can support several services, such as (i) the monitoring of one fog node by another, (ii) identifying the list of neighboring nodes, and (iii) providing transmission checkpointing [32]. The original scheme of the ant and bee colony and particle swarm optimization is hybridized in deriving the two proposed novel techniques of (i) the checkpointing technique and (ii) particle bee optimization. The implemented techniques offer various services in a fog-collaborated WSN environment, such as

- Fault tolerance (reducing energy consumption);
- Fog service placement (handling mobile clients from multiple data centers positioned at different locations);
- Performing transmission checkpointing (reducing the packet transmission time by choosing the nearest neighbor) [33];
- Network optimization (identifying the best cost path).

The offered services are implemented utilizing the two techniques of (i) checkpointing and (ii) particle bee optimization. To build fault tolerance in fog computing when utilizing the IoT, the following assumptions are considered:

(i) Fog sensor devices are configured in such a way that the devices are capable of communicating either directly or indirectly, with more than one neighboring node.
(ii) Fog nodes withstand node failure and perform the necessary communication.
(iii) Fog node coverage in multiple distribution areas increases the fault tolerance level.
(iv) Input population localization will enable fault discovery.

Thus, fog nodes provide services by considering: (i) the resource availability for service placements, (ii) accessibility maintenance among multiple services, (iii) real-time service constraints with periodic updates, and (iv) timing; resource constraints will not overlap the proxy replications [17]. The pseudocode represents the various steps carried out in the presented tasks.

The particle bee optimization algorithm initializes the input population and identifies the fog sensor source and its neighbor. The completion time of each transmission is analyzed, based on which of the best solution costs is fixed [19]. This step is repeated until all the services are assigned. The source device reaches the neighboring node for communication. The neighboring node N receives the data packets and acknowledges the source device for every received packet. The acknowledgment includes the number of packets received, along with the minimum, maximum, and average turnaround times. This iteration is repeated until all the devices identify their neighbors [34]. Next is the fog service placement, which offers multiple benefits. Both the fog server and its replication proxy server serve to meet regional service needs. The nature of the ant inspired an optimization technique for constructing the best cost solution. This converges the terms of the best cost solution space via subsequent iterations exploiting more favorable outcomes [35]. The performance evaluation of the novel QoS-EO scheme is compared with the best achievement of the algorithm as shown. The results are presented in detail in Section 4.

```
Pseudocode
// Fog Service Placement//
Begin
Initialization: SDevice_ID;
Fog Sensors==Enabled;
Begin
Run((vr_game_0)&&(vr_game_1))
For(vr_game_0; SDevice_ID++)
Actuator Signal==Enabled;
End for
For(vr_game_1; SDevice_ID++)
Actuator Signal==Enabled;
End for
End
End
// Check Point Testing//
Begin
Set IP, No. of Bytes, Time t, TTL;
Display(){
For (Device_ID; Device Status==Active; Device ++)
No. of Packets Sent;
No. of Packets Received;
Lost==0;
Maximum Round Trip(ms);
Minimum Round Trip(ms);
Average Round Trp(ms);
Calculate()
{Throughput; Response Time; Scalability; Performance; Availability; Usability;
Reliability; Overhead; Cost;}
End For }
End
// Particle Bee Optimization //
Begin
For(X==n; Y==m; Iterate++;){
Iterate()
Until Solution(Best X, Best Y);}
End For
Initialize Input Population;
Food Foraging of Bees()
{ Fog Sensor==Source;
Calculate Source Cost;
Fog Sensor==Neighbour;
Calculate Neighbour Cost;}
Until the Cycle for all sources is completed;
Display Best Solution Cost();
End//Particle Bee Optimization
//Fault Tolerance Service Placement//
Begin
Display()//For each Fog device
{
Execution Time;
Loop Delays;
Tuple CPU Execution Delay;
Player_Game_State; Global_Game_State;
EEG; Concentration; Sensor;
Energy Consumed;}
End
```

4. Results and Discussion

The results of the CP testing for network performance evaluation are summarized in Figure 2. For creating a simulation of the proposed model, CloudSim was used. Five sensor nodes were considered for technical feasibility. Sensor node 1 is represented by SN1 (as shown in Table 1). All devices are assumed to be operational and network performance is measured, using metrics such as performance, throughput, response time, scalability, availability, usability, reliability, overhead, and cost-effectiveness. The number of bytes transmitted is taken to be 32 bytes, divided into four packets, with a transmission time of 1 ms for each packet. The TTL for each packet was 121, which was assigned at random. In this scenario, all packets are received with no packet loss. A maximum turnaround time of 8 ms, a minimum turnaround time of 1 ms, and an average turnaround time of 3 ms were observed.

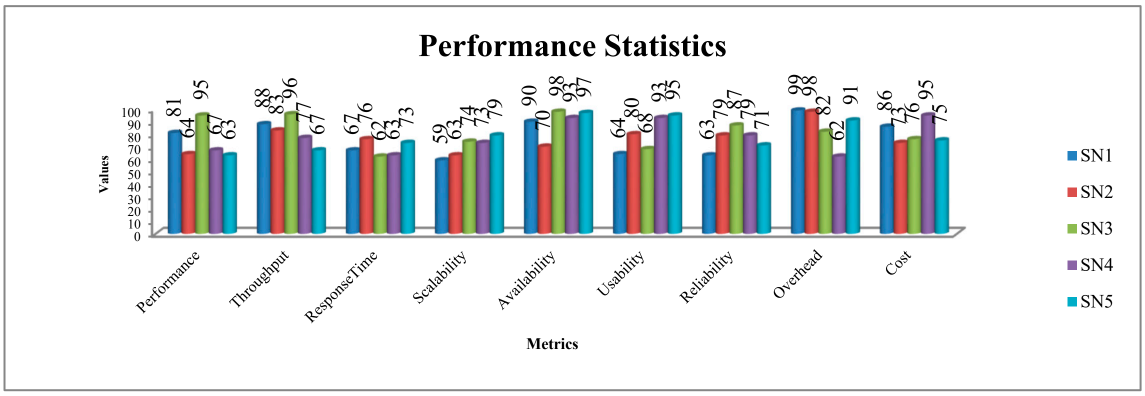

Figure 2. Performance statistics graph of the CP technique test.

Table 1. The calculation parameters.

Parameters	Numbers
Sensor nodes	5
Data transmitted	32 bytes
Per packet	8 bytes
TTL	Random
Maximum Time	8 ms
Minimum Time	1 ms

Figure 2 depicts the CP technique's performance statistics. A total of five sensor nodes were considered. The values of SN1, SN2, SN3, SN4, and SN5 performance metrics were 81, 64, 95, 67, and 63, respectively. The nodes' throughput measurements were 88, 83, 96, 97, and 67, respectively. Almost all the nodes demonstrated high availability; for instance, SN1, SN3, SN4, and SN5 had usability values of 90, 98, 93, and 97, respectively. It was discovered that the network had 95, 96, and 98 percent performance, throughput, and availability. As a result, the proposed technique's efficacy is demonstrated.

Table 2 shows the best cost estimation using the particle bee optimization technique. The cost was calculated based on 30 iterations. As the number of iterations increases, the path cost value decreases, demonstrating the efficacy of the proposed technique. Figure 3 also depicts the fog device tuple CPU execution delay. The delay in the device were taken into account for five states: global, sensor, concentration, EGG, and player. There were only minimal CPU execution delays of 0.06 s, 3.19 s, 3.42 s, and 2.33 s, respectively. As a result, the proposed QoS-EO scheme can be considered efficient.

Table 2. The best cost estimations.

Iteration	Best X	Best Y	Value
0	3.452	0.670	0.253
1	2.521	0.348	0.130
7	3.241	0.564	0.019
11	3.166	0.539	0.009
15	3.166	0.539	0.009
19	3.154	0.540	0.008
21	3.154	0.540	0.008
25	2.980	0.503	0.001
30	3.010	0.503	0.000

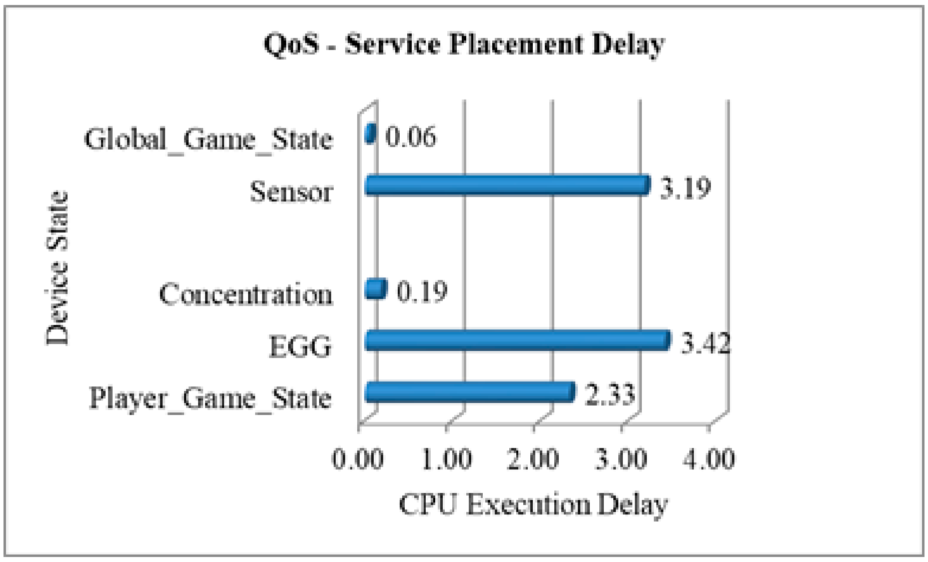

Figure 3. QoS service placement delay.

Further, the evaluation of the proposed scheme is compared with the best achievement demonstrated by the genetic algorithm. Comparing the existing algorithm, the proposed QoS-EO scheme shows better results in fulfilling the objective of the research as described. As the game player is enabled simultaneously for each device, the actuator signals are also enabled to perform the corresponding reply action for each transmission. Furthermore, the proposed QoS-EO scheme's efficiency is compared to that of the existing algorithm, as shown in Table 3.

Table 3. Comparison of the proposed technique with the existing technique.

	Existing Genetic Algorithm	Proposed Qos-EO
Fog cloud energy optimization	593,564.4	449,575.1
Fog cloud execution cost	85,042.2	83,142.2
Total network usage	635,550.0	611,643.3

Figure 6 compares the performance of the novel technique to that of the genetic algorithm. The sensor nodes send requests with timestamps to the server, which are used as input for the QoS-EO scheme. A service-based model combined with fog computing enables the machine-to-machine communication protocol to reliably transfer data. The proposed QoS-EO scheme reliably performs a data transfer in less time. The local backup supports the devices in the event of server failure; therefore, request redundancy and network latency are overcome. The architecture is scalable in supporting the fog environment, offering scope for future research. Multisensor fusion could be used to reduce data transmission issues. With their many constraints, the researchers managed parallel application composition using an existing genetic algorithm-based framework in which chromosomes represent a solution and gene segments represent IoT applications. After each iteration, the aforesaid centralized system in the sink node collects all information and broadcasts it; this was considered to be the benchmarked solution [1]. However, the proposed scheme integrates IoT, entailing resource selection and efficient service workflow, thus providing flexibility, data security, and efficiency at a lower cost. Furthermore, this is the first model of its kind to introduce a concrete methodology for WSN, extended in collaboration with the fog environment. As pointed [36–41] and Fernando et al. [29], optimization in IoT applications is an intriguing research challenge. Here, this proposed scheme utilizes novel techniques, such as a novel checkpoint (CP) technique, along with the novel particle bee optimization. The CP technique is used to test network fault tolerance, while the novel particle bee optimization technique is used to test the quality-of-service placement and optimize the network energy. When compared to the existing genetic algorithm (shown in Figure 4), the energy optimization of the proposed technique was 449,575.1. As a result, the proposed QoS-EO scheme is efficient. According to the comparison results, the proposed QoS-EO scheme has a lower network usage of 611,643.3 and a lower execution cost of the fog cloud of 83,142.2, as shown in Figures 5 and 6. When integrating IoT devices in a fog environment, resource provisioning is a critical concern. This is especially important because one of the reasons for using fog computing in IoT scenarios is to avoid a longer delay. However, the proposed QoS-EO scheme for fog computing needs ensures that the data transfer times and costs are reduced. There will also be a high level of network availability, throughput, and performance guarantee.

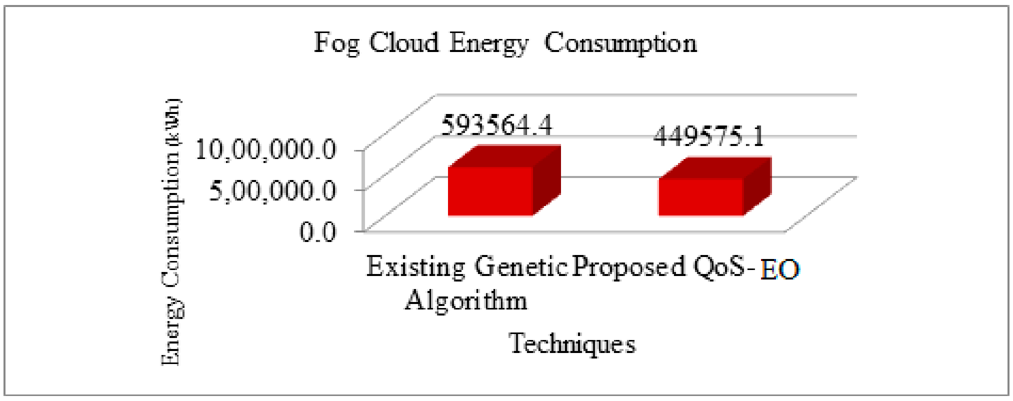

Figure 4. Comparison of fog cloud energy optimization for the existing and proposed methods.

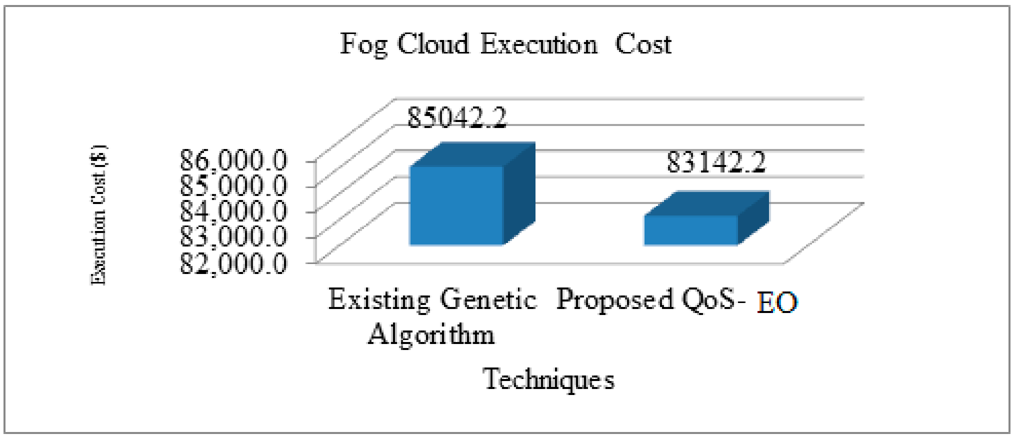

Figure 5. Comparison of fog cloud execution costs for the existing and proposed methods.

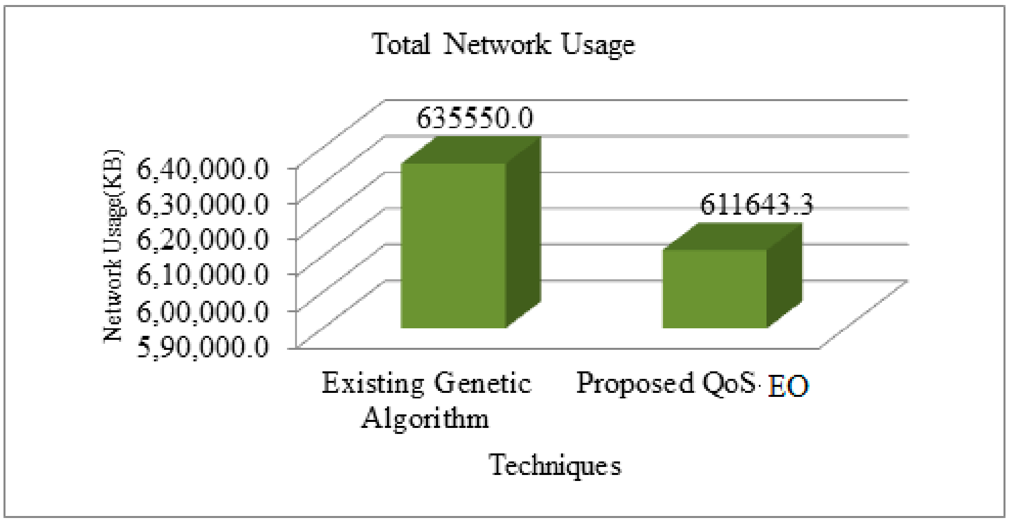

Figure 6. Comparison of total network usage for the existing and proposed methods.

5. Conclusions

The network consists of enormous sensor devices that are all connected; these observe situations and communicate the information to the server, enabling the control center to make informed decisions. The use of data centers for data processing is not feasible, resulting in propagation delays, congestion, high execution costs, and slow delivery times. The results show a maximum turnaround time of 8 ms, a minimum turnaround time of 1 ms, and an average turnaround time of 3 ms. The cost, as calculated, indicates that as the number of iterations increases, the path cost value decreases, demonstrating the efficacy of the proposed technique. There was only a minimum CPU execution delay of 0.06 s. In comparison, the proposed QoS-EO scheme has a lower network usage of 611,643.3 and a lower execution cost of the fog cloud of 83,142.2. Thus, the proposed QoS-EO scheme:

(i) reliably performs data transfer within a short time;
(ii) handles control server failure, making the devices receive responses through the local backup, created by the middleware layer's databases;
(iii) overcomes the network latency, bandwidth, and redundancy requests;
(iv) ensures reduced data transfer time and cost;
(v) provides a high level of network availability, throughput, and performance guarantee.

Author Contributions: Methodology, S.K.; Software, A.R.; Formal analysis, H.M.Z.; Investigation, A.W.M.; Writing—original draft, A.R., S.K., K.A.A., H.M.Z. and A.W.M.; Writing—review & editing, A.R., S.K., K.A.A., H.M.Z. and A.W.M.; Funding acquisition, K.A.A. All authors have read and agreed to the published version of the manuscript.

Funding: The research is funded by the Researchers Supporting Program at King Saud University, (RSP2023R305).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors present their appreciation to King Saud University for funding the publication of this research through the Researchers Supporting Program (RSP2023R305), King Saud University, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

M_{re}	Successful transmission
$er_{i,j}$	Error rate
$d_{i,j}$	Links degree estimation
i,j	Nodes
α	Decision variable
p	Cumulative probability of sensor failure
λ_s	Failure rate
d_i	Degree of i
d_j	Degree of j
α	Decision variable
c_k	Fault detection accuracy
c_c	Probability of an unsuccessful replacement
p	Cumulative probability
$M(p)$	Function of p
λ_s	Failure rate over period
c	Region coverage factor
t_s	Time
N	Node
SN	Sensor node
TTL	Time to live
ms	milliseconds
d_i	Degree of i
d_j	Degree of j
α	Decision variable
QoS-EO	Quality-of-service-based energy optimization
WSN	Wireless sensor networks
IoT	Internet of Things
CP	Checkpointing
QoS	Quality of service
MILP	Mixed integer linear program

References

1. Naeem, R.Z.; Bashir, S.; Amjad, M.F.; Abbas, H.; Afzal, H. Fog computing in internet of things: Practical applications and future directions. *Peer-to-Peer Netw. Appl.* **2019**, *12*, 1236–1262. [CrossRef]
2. Mahmud, R.; Toosi, A.N.; Ramamohanarao, K.; Buyya, R. Context-aware placement of Industry 4.0 applications in fog computing environments. *IEEE Trans. Ind. Inform.* **2019**, *16*, 7004–7013. [CrossRef]
3. Lopez Medina, M.A.; Espinilla, M.; Paggeti, C.; Medina Quero, J. Activity recognition for iot devices using fuzzy spatio-temporal features as environmental sensor fusion. *Sensors* **2019**, *19*, 3512. [CrossRef] [PubMed]
4. Yu, L.; Lu, Y.; Zhang, B.; Li, Y.; Huang, F.; Shen, Y. Environment-oriented Internet of Things Service Modeling Integrating with User Requirements. In Proceedings of the 2020 IEEE 10th International Conference on Electronics Information and Emergency Communication (ICEIEC), Beijing, China, 17–19 July 2020; IEEE: New York, NY, USA, 2020; pp. 155–163.
5. Tran, M.Q.; Nguyen, D.T.; Le, V.A.; Nguyen, D.H.; Pham, T.V. Task placement on fog computing made efficient for iot application provision. *Wirel. Commun. Mob. Comput.* **2019**, *2019*, 6215454. [CrossRef]
6. Cao, Y.; Chen, S.; Hou, P.; Brown, D. FAST: A Fog Computing Distributed Analytics-based Fall Monitoring System for Stroke Mitigation. In Proceedings of the 2015 IEEE International Conference on Networking, Architecture and Storage (NAS), Boston, MA, USA, 6–7 August 2015.
7. Brogi, A.; Forti, S.; Ibrahim, A.; Rinaldi, L. Bonsai in the fog: An active learning lab with fog computing. In Proceedings of the 2018 Third International Conference on Fog and Mobile Edge Computing (FMEC), Barcelona, Spain, 23–26 April 2018; IEEE: New York, NY, USA, 2018; pp. 79–86.
8. Mangan, N.M.; Brunton, S.L.; Proctor, J.L.; Kutz, J.N. Inferring biological networks by sparse identification of nonlinear dynamics. *IEEE Trans. Mol. Biol. Multi-Scale Commun.* **2016**, *2*, 52–63. [CrossRef]
9. Kishor, A.; Chakarbarty, C. Task Offloading in Fog Computing for Using Smart Ant Colony Optimization. *Wirel. Pers. Commun.* **2021**, *127*, 1683–1704. [CrossRef]
10. Xu, R.; Wang, Y.; Cheng, Y.; Zhu, Y.; Xie, Y.; Sani, A.S.; Yuan, D. Improved particle swarm optimization based workflow scheduling in cloud-fog environment. In Proceedings of the Business Process Management Workshops: BPM 2018 International Workshops,

Sydney, Australia, 9–14 September 2018; Revised Papers 16. Springer International Publishing: Berlin/Heidelberg, Germany, 2019; pp. 337–347.
11. Varghese, B.; Wang, N.; Nikolopoulos, D.S.; Buyya, R. Feasibility of fog computing. *arXiv* **2017**, arXiv:1701.05451.
12. Mohammadi, V.; Rahmani, A.M.; Darwesh, A.; Sahafi, A. Fault tolerance in fog-based Social Internet of Things. *Knowl.-Based Syst.* **2023**, *265*, 110376. [CrossRef]
13. Gill, M.; Singh, D. ACO Based Container Placement for CaaS in Fog Computing. *Procedia Comput. Sci.* **2020**, *167*, 760–768. [CrossRef]
14. Gong, W.; Zu, Y. Research and Simulation Implementation of Fog Calculation Resource Allocation Algorithm. In Proceedings of the 2019 IEEE 5th International Conference on Computer and Communications (ICCC), Chengdu, China, 6–9 December 2019; IEEE: New York, NY, USA, 2019; pp. 1126–1130.
15. Wang, Y.; Zhao, C.; Xu, S. Method for Spatial Crowdsourcing Task Assignment Based on Integrating of Genetic Algorithm and Ant Colony Optimization. *IEEE Access* **2020**, *8*, 68311–68319. [CrossRef]
16. Martinez, I.; Hafid, A.S.; Gendreau, M. Robust and Fault-Tolerant Fog Design and Dimensioning for Reliable Operation. *IEEE Internet Things J.* **2022**, *9*, 18280–18292. [CrossRef]
17. Mohamed, N.; Al-Jaroodi, J.; Jawhar, I. Towards fault tolerant fog computing for IoT-based smart city applications. In Proceedings of the 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 7–9 January 2019; IEEE: New York, NY, USA, 2019; pp. 0752–0757.
18. Ahmed, A.; Abdullah, S.; Iftikhar, S.; Ahmad, I.; Ajmal, S.; Hussain, Q. A novel blockchain based secured and QoS aware IoT vehicular network in edge cloud computing. *IEEE Access* **2022**, *10*, 77707–77722. [CrossRef]
19. Wang, W.; Feng, C.; Zhang, B.; Gao, H. Environmental monitoring based on fog computing paradigm and internet of things. *IEEE Access* **2019**, *7*, 127154–127165. [CrossRef]
20. Kaiwartya, O.; Abdullah, A.H.; Cao, Y.; Lloret, J.; Kumar, S.; Shah, R.R.; Prakash, S. Virtualization in wireless sensor networks: Fault tolerant embedding for internet of things. *IEEE Internet Things J.* **2017**, *5*, 571–580. [CrossRef]
21. Vambe, W.T.; Chang, C.; Sibanda, K. A review of quality of service in fog computing for the Internet of Things. *Int. J. Fog Comput. IJFC* **2020**, *3*, 22–40. [CrossRef]
22. Uviase, O.; Kotonya, G. IoT architectural framework: Connection and integration framework for IoT systems. *arXiv* **2018**, arXiv:1803.04780. [CrossRef]
23. Okafor, K.C.; Achumba, I.E.; Chukwudebe, G.A.; Ononiwu, G.C. Leveraging Fog Computing for Scalable IoT Datacenter Using Spine-Leaf Network Topology. *J. Electr. Comput. Eng.* **2017**, *2017*, 2363240. [CrossRef]
24. Skarlat, O.; Nardelli, M.; Schulte, S.; Borkowski, M.; Leitner, P. Optimized IoT service placement in the fog. *Serv. Oriented Comput. Appl.* **2017**, *11*, 427–443. [CrossRef]
25. Wen, Z.; Yang, R.; Garraghan, P.; Lin, T.; Xu, J.; Rovatsos, M. Fog orchestration for internet of things services. *IEEE Internet Comput.* **2017**, *21*, 16–24. [CrossRef]
26. Deng, H.; Guo, Z.; Lin, R.; Zou, H. Fog computing architecture-based data reduction scheme for WSN. In Proceedings of the 2019 1st International Conference on Industrial Artificial Intelligence (IAI), Shenyang, China, 23–27 July 2019; IEEE: New York, NY, USA, 2019; pp. 1–6.
27. Kashani, M.H.; Rahmani, A.M.; Navimipour, N.J. Quality of service-aware approaches in fog computing. *Int. J. Commun. Syst.* **2020**, *33*, e4340. [CrossRef]
28. Jiang, J.; Li, Z.; Tian, Y.; Al-Nabhan, N. A Review of Techniques and Methods for IoT Applications in Collaborative Cloud-Fog Environment. *Secur. Commun. Netw.* **2020**, *2020*, 8849181. [CrossRef]
29. Fernando, N.; Loke, S.W.; Avazpour, I.; Chen, F.F.; Abkenar, A.B.; Ibrahim, A. Opportunistic fog for IoT: Challenges and opportunities. *IEEE Internet Things J.* **2019**, *6*, 8897–8910. [CrossRef]
30. Ozeer, U.; Etchevers, X.; Letondeur, L.; Ottogalli, F.G.; Salaün, G.; Vincent, J.M. Resilience of stateful IoT applications in a dynamic fog environment. In Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, New York, NY, USA, 5–7 November 2018; pp. 332–341.
31. Hasan, M.Z.; Al-Turjman, F. Optimizing multipath routing with guaranteed fault tolerance in Internet of Things. *IEEE Sens. J.* **2017**, *17*, 6463–6473. [CrossRef]
32. Fei, J.; Xiaoping, M. Fog computing perception mechanism based on throughput rate constraint in intelligent Internet of Things. *Pers. Ubiquitous Comput.* **2019**, *23*, 563–571. [CrossRef]
33. Eyckerman, R.; Mercelis, S.; Marquez-Barja, J.; Hellinckx, P. Requirements for distributed task placement in the fog. *Internet Things* **2020**, *12*, 100237. [CrossRef]
34. Huang, H.; Zheng, Y.R. Node localization with AoA assistance in multi-hop underwater sensor networks. *Ad. Hoc. Netw.* **2018**, *78*, 32–41. [CrossRef]
35. Celesti, A.; Carnevale, L.; Galletta, A.; Fazio, M.; Villari, M. A watchdog service making container-based micro-services reliable in IoT clouds. In Proceedings of the 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud), Prague, Czech Republic, 21–23 August 2017; IEEE: New York, NY, USA, 2017; pp. 372–378.
36. Plebani, P.; Garcia Perez, D.; Anderson, M.; Bermbach, D.; Cappiello, C.; Kat, R.I.; Pallas, F.; Pernici, B.; Tai, S.; Vitali, M. Information logistics and fog computing: The DITAS∗ approach. In Proceedings of the Forum and Doctoral Consortium Papers

Presented at the 29th International Conference on Advanced Information Systems Engineering, CAiSE-Forum-DC 2017, Essen, Germany, 12–16 June 2017; CEUR-WS: Aachen, Germany, 2017; pp. 129–136.
37. Ali, I.; Sarkar, B.; Ali, S.M.; Fügenschuh, A. Editorial: Environmental waste and renewable energy optimization for the sustainable development goals achievement. *Front. Environ. Sci.* **2023**, *11*, 1167835. [CrossRef]
38. Ali, I.; Modibbo, U.M.; Chauhan, J.; Meraj, M. An integrated multi-objective optimization modelling for sustainable development goals of India. *Environ. Dev. Sustain.* **2021**, *23*, 3811–3831. [CrossRef]
39. AlArjani, A.; Modibbo, U.M.; Ali, I.; Sarkar, B. A new framework for the sustainable development goals of Saudi Arabia. *J. King Saud Univ.-Sci.* **2021**, *33*, 101477. [CrossRef]
40. Ahmadini, A.A.H.; Modibbo, U.M.; Shaikh, A.A.; Ali, I. Multi-objective optimization modelling of sustainable green supply chain in inventory and production management. *Alex. Eng. J.* **2021**, *60*, 5129–5146. [CrossRef]
41. Modibbo, U.M.; Ali, I.; Ahmed, A. Multi-objective optimization modelling for analysing sustainable development goals of Nigeria: Agenda 2030. *Environ. Dev. Sustain.* **2021**, *23*, 9529–9563. [CrossRef]