Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2023.028473
Article

Trust and QoS-Driven Query Service Provisioning Using Optimization

K. Narmatha[1,*] and K. Karthikeyan[2]

[1]Department of Computer Science and Engineering, SRM Institute of Engineering and Technology, Ramapuram, Chennai, India
[2]Department of Information Technology, Coimbatore Institute of Technology, Coimbatore, India
*Corresponding Author: K. Narmatha. Email: narmathakcool@gmail.com
Received: 11 February 2022; Accepted: 24 June 2022

Abstract: The growing advancements with the Internet of Things (IoT) devices handle an enormous amount of data collected from various applications like healthcare, vehicle-based communication, and smart city. This research analyses cloud-based privacy preservation over the smart city based on query computation. However, there is a lack of resources to handle the incoming data and maintain them with higher privacy and security. Therefore, a solution based idea needs to be proposed to preserve the IoT data to set an innovative city environment. A querying service model is proposed to handle the incoming data collected from various environments as the data is not so trusted and highly sensitive towards vulnerability. If handling privacy, other inter-connected metrics like efficiency are also essential, which must be considered to fulfil the privacy requirements. Therefore, this work provides a query-based service model and clusters the query to measure the relevance of frequently generated queries. Here, a Bag of Query (BoQ) model is designed to collect the query from various sources. Validation is done with a descriptive service provisioning model to cluster the query and extract the query's summary to get the final results. The processed data is preserved over the cloud storage system and optimized using an improved Grey Wolf Optimizer (GWO). It is used to attain global and local solutions regarding privacy preservation. The iterative data is evaluated without any over-fitting issues and computational complexity due to the tremendous data handling process. Based on this analysis, metrics like privacy, efficiency, computational complexity, the error rate is analyzed. The simulation is done with a MATLAB 2020a environment. The proposed model gives a better trade-off in contrast to existing approaches.

Keywords: Cloud computing; IoT; privacy; security; grey wolf optimization

1 Introduction

Internet of Things (IoT) is a platform that links the physical world and cyber world. This IoT technique makes the user gather information through sensors from the physical world to run queries regarding the collected data [1] and analyze the system's performance. Various convenient methods are suggested, such as DeepDirect to bind the direction of learning [2] to monitor the data and examine it in shared circumstances. The IoT architecture called Multi-access Edge Computing (MEC) [3] emerges to comprise

the edge servers. Contrary, MEC techniques perform complex computation and pose huge storage sources from the physical world than the conventional IoT techniques. This work proposed to address the problem of query processing in the recent IoT techniques. One of the essential tasks in the traditional IoT techniques is query processing, such as Wireless Sensor Networks (WSNs). The method has various current tasks [4].

There are two essential procedures: distributed strategy and centralized strategy in query processing for WSNs. Queries are processed with the help of cooperative sensor nodes in the distribution strategy. In contrast, all the sensed data are sent to cloud, responsible for query processing in the centralized system. Moreover, in the new IoT techniques, query processing is more complicated, and the mentioned query processing strategies are not helpful because of the below two difficulties. They are (i) voluminous sensed data [5]. Based on Cisco's report [6], 31 billion connected IoT devices will be there at the end of 2020, and there are 75 billion devices will be available at the end of 2025. IoT devices provides huge amount of sensed data and process the query which leads to complex calculation and data transferring complexity. Data transmission is gradual, and the calculation workload will burden the cloud or sinks. Hence, the centralized strategy is not suited in wireless sensor networks. (ii) Complex Queries–The existing system made an attempt in query processing in the shared fashion in wireless sensor networks like range query, curve query, and top-k query [7]. Moreover, the queries are more complicated than those used in the conventional wireless sensor networks in the latest IoT techniques. The data is preprocessed through the sequence of services, including speech recognition, image processing, integration of data, and a node can deploy the AI model to answer the query in the new IoT technique and process the services in the data processing. For instance, if a user selects a vehicle's license number that moves faster, the vehicles' speed data can sort to attain the fast vehicle and recognize the vehicles' image data to achieve the license number. The outcomes of mentioned two stages have to join to reach the outcome [8]. Moreover, the services mentioned earlier in data processing cannot be able to carry out at every sensor node since the sensor nodes have restricted calculation in wireless sensor networks, energy, and storage resources. It makes wireless sensor networks unable to process complicated queries in a shared fashion. Henceforth, the distribution strategy is not suited in wireless sensor networks.

Considering Edge Assisted IoT Data Monitoring System (EDMS) to process complicated queries in the new IoT technique (See Fig. 1). This EDMS can deploy to observe the security, traffic, environment change, and so on. However, EDMS can deploy to monitor the industrial parameters in the big factory. Fig. 1 represents the overview of EDMS [9]. The sensed data are gathered from various sources and are deposited in the shared fashion where the edge servers are linked with the remote cloud. It is used to translate the query into a sequence of services in data processing like information integration, image processing, top-k query processing, etc. These services are cached at the edge server in the data processing. It creates a plan for the query and allocates the work to a few edge servers if the query comes nearer to the cloud. The voluminous sensed data and complex queries should be processed in the shared fashion; since the edge servers require more storage resources and more calculation. In EDMS, the distributed query processing can better utilize resources and minimize the transmission and cost of computation [10]. Query processing is still tricky in heterogeneous system with various servers with multiple calculations and abilities in communication. Further, every query is interconnected in the data processing services, and the result of one service may be the input of other benefits. Henceforth, the unnecessary response latency is raised due to an improper query plan. This work investigates how to create a plan for query to boost the response delay of query in the EDMS. It is the first study examining query processing as per our knowledge. The benefaction of this work is described below.

- To model a query service mechanism for the IoT-based data monitoring system and to establish privacy of the network model;

- To analyze the trust and query service using a meta-heuristic optimization approach. Here, Gray Wolf Optimization (GWO) approach is adopted for managing the query service;
- To simulate the performance of the anticipated model over MATLAB 2020a simulation environment and compare the performance with various existing approaches.

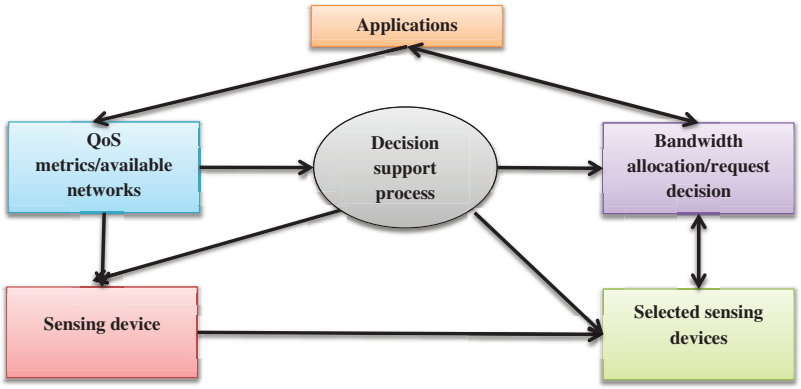

Figure 1: IoT data edge monitoring system

The work is structured as: Section 2 gives a comprehensive analysis of various existing approaches and discusses the pros and cons of the prevailing models. Section 3 gives the detailed analysis of the anticipated model in query processing, Quality of Service (QoS)-aware heuristic model and privacy to the network model. In Section 4, the numerical outcomes of the work are provided with graphical representation and work summary in Section 5.

2 Related Works

The IoT technique has distinguished the state of data processing and collection. The gathered data has the Spatio-temporal context to perform an essential part in processing any analytics outcome and aligns the decision-making with two aspects [11]. The difficulties found that need to serve in the domain with data management effectively and the number of devices. A group of attempts disclose the chances for the distributed data management or distributed nodes' management available in IoT. The identification of nodes is focused efficiently based on the request depending on the static criterion that explains the nodes or the data themselves [12]. The difficulty is obtaining a glance at the characteristics of nodes and the available data's statistics. Moreover, edge nodes and IoT have revealed the various hardware and software features (for example, middleware). The researcher suggests a Distributed Data Service (DDS) that supports collecting and processing data [13]. This primary goal is to enable various and different IoT middleware systems to allocate the data services in common to cover the interoperability problems.

The execution of queries in parallel enhances the processing speed. It can efficiently distribute the application. Moreover, the data is partitioned to realize the parallel execution in the provided network environment. l. This setting is the typical scenario if considering the nodes or edge infrastructure. Multiple attempts are carried out to provide the separation algorithms of data over the top of batches or streams. The authors choose a sliding window technique [14]. Streams are divided on the fly by considering the query semantics. It proposes a multi-route optimizer and uses inter-stream and intra-stream correlation to generate the partitions efficiently. The authors present the separations of streams into a group of sub-streams on the operators of queries implemented in parallel. The group of properties are used to characterize the suggested partitioning functions that are structural properties (for example. Fast

lookup, compactness), balance properties (for example, Processing, memory and communication), and adaptation properties (for example, Low migration, fast calculation).

The process of deploying the calculation and storage resources nearer to the mobile device or user by the MEC [15] helps to release the pressure on the workload of the cloud. Sensed data sent from the devices to the servers to analyze further. The authors in [16,17] examine the offloading issue and one mobile device and one MEC server. Some studies [18,19] discuss the offloading task issue in a MEC system with mobile devices and one MEC server. Few studies are concerned about the placement of service and the request routing issues. The authors aim to examine how to move or place the edge services to minimize latency response in in-service placement [20]. The request routing issue examines the user's sending process and requests to suitable edge servers to enhance performance. Moreover, no works are mentioned about processing queries at edge servers.

The Directed Acyclic Graph (DAG) represents relationships among services. The author utilizes a directed edge in this paper to present the dependence among two services or two tasks. Every directed edge (f_i) denotes that f_j can process if f_i is processed. Moreover, the authors are not concerned entirely with the dependence of data among services in this paper. The author suggests that the Data-Based DAG considers the data dependence among services in this paper. An essential technique called query processing helps the users approach the data, which is regarded as a traditional problem in wireless sensor networks [21–23]. The author examines collecting the sensed data in the battery-free sensor networks and energy harvesting networks. The author suggests a few secure domains about the algorithms in processing the query in WSNs. An adaptive clustering technique is proposed with the two spatial-correlation and aggregation algorithms that support query aggregation processing. It processing is examined depending on the itinerary-based approach. The proposed system examines the curve in processing the query for WSNs.

The author [24] investigated the QoS parameters used for selection and service composition by evaluating some QoS parameters over the application layer: response time, execution time, reliability, cost, price, throughput, latency, and reputation. Moreover, the author does not discuss methods or techniques utilized to resolve the selection issues. The author performed the systematic analysis of SSA over the IoT environment which is classified as hybrid, decentralized and centralized classes. The author concentrates on some techniques to resolve the selection issue and QoS parameters used to evaluate the disadvantages and advantages of every algorithm. The author predicts the most influencing and challenging issues in the selection process over IoT environment.

The author [25] proposed a classification process to fulfil the IoT requirements and facilitates optimization over the various IoT layers. The proposed architecture composed of three layers like application, network and sensor layers. The conventional QoS attributes are merged with various IoT architectural characteristics, i.e., network cost, coverage, energy consumption, information accuracy and network deployment. The conventional QoS classification is not suitable to the complexity and heterogeneity of the IoT architecture. The sensor layer specifies the physical IoT infra-structure comprising of various edge nodes link like RFID tags, data centres, mobile devices, sensor networks and other heterogeneous devices. The layers perform sensing process as independent services and facilitates IoT environment to offer sensing and actuating abilities modelled as services with edge-node services via the cloud computing system. The QoS related layers include the sensor selection with essential infra-structure based on application/user requirements and sensing abilities. Therefore, this layers attempts to handle the scheduling and resource allocation process. For instance, the QoS in sensor layers include system lifetime, energy consumption and resource optimization. The SSA optimization is essential for QoS optimization for various sensor services. An optimal SSA is essential for sensor layer as multiple devices are suitable with some variation in quality that fulfils the application and user requirements. The service availability with these layers describes the failure or success of service request [25,26].

3 Methodology

3.1 System Model

Data Monitoring System in IoT is depicted in Fig. 1. This system evolves the three main elements. They are (i) a network comprised of servers $E+ = \{E_0,\ E1,\ E_2,\ \ldots,\ E|E|\}$ that includes remote cloud E_0 and edge servers $E = \{E_1,\ E_2,\ \ldots,\ E|E|\}$, (ii) users U, and (iii) region under monitoring R. The edge servers are deployed in E that is near to the monitored region and gather the sensed data. A group of services in data processing is furnished with every edge server to process the sensed data. The user submits the query requests to recover the information of the physical world. The submitted query requests are handled by the cloud and provide the outcome to users. Thus the cloud generates plan for query processing and transfer it to the edge servers. The edge servers $(E_1\ and\ E_2)$ with the interested data by users will send the data with suitable data processing service (E_3) to the edge servers processed after receiving the processing plan. After processing the data, the query outcome is sent to the cloud.

The sensed data are gathered at the edge servers in a shared fashion. The gathered sensed data are the collected raw data from various sources like microphones, sensors, cameras, etc. Some preprocessing steps are carried out by the data like data integration, data sorting, speech recognition, image processing, and so on to validate query processing [27–30]. Moreover, if processing the queries in the cloud, complete sensed data needs to gather that will consume time, involves complex processing queries that will consume resources. A data query is considered the sequence of data processing tasks in this scenario due to the services in data processing located at edge servers. Then, processing the queries in a shared manner minimizes the query response latency and mitigate the cloud's workload. [31,32]

3.2 Sensory Data

Suppose that data sources D types are available in the monitored region like the surveillance cameras gives image data, temperature sensors provide temperature data, speed detectors gives speed data, etc. The sensed data created in the monitored region denotes $S = \{S(i)|1 \le i \le D\}$, where $S(i)$ indicates sensed data gathered from the data source type $i(1 \le i \le D)$. Suppose every sensed data is not stored in a shared fashion at the edge servers. The $S(E_j) = S(1 \le i \le D)$, where $S(i)\ (E_j)$ denotes the group of sense data placed at E_j, where $S(i)\ (E_j)$ is the group of data type i. For each two edge servers $E_k,\ E_j \in E$, we are having $S(i)\ (E_j) \cap S(i)(E_k) = \varnothing$. For the complete edge servers, as having $S\ (E_j) \in E\ S(i)(E_j) = S(i)$.

3.3 Communication

The remote cloud and the edge servers interface using a backbone network. The communication among the cloud and edge servers are more decelerated (without loss of generality). For every link (i, j) where $E_j \in E+$, let $l_{i,j}$ denotes the delay in communication while transferring one unit of data. It implies $min\{10,\ i|E_i \in E\}\ max\{\ l_{i,j}\,|E_i,\ E_j \in E\}$. Assume that E_i sends a group of data O to E_j. The delay in transmission depicts as $L_{i,j}(O) = |O|l_{i,j}$.

3.4 Computation

The remote cloud and edge servers give various data processing services like data integration, speech recognition, and image processing for processing the sensed data. Suppose F denotes the universal set of services. For every $E_i \in E \cup \{E_0\}$, let C_i means the computing resource of E_i and $F_i = \{f_1, f_2,\ \ldots, f|F_i|\} \subseteq F$ represents the group of services given by E_i. E_0 represent the cloud that offers complete services in F that is $F_0 = F$. Suppose every $E_i \in E+$ can take one service only simultaneously, and the benefits are not disrupted. Let I and O is input/output data, for service $f \in F_i$ that is also referred to as $f(I) = O$, and

$C(f(I))$ denote the computing resources required with I input by f. Then the latency processing $f(I)$ expresses as in Eq. (1):

$$L_i(f(I) = \frac{C(f(I))}{C_i} \tag{1}$$

The computing resources are required that are noticed by $f(I)$, and the output's size is O that relates to the input size I. Henceforth, it is shown in Eq. (2):

$$L_i(f(I) = \frac{C_f(|I|)}{C_i} \tag{2}$$

and the connectivity among $|I|$ and $|O|$ is expressed in Eq. (3):

$$|O| = g_f(|I|) \tag{3}$$

where $c_f\ (.)$ and $g_f\ (.)$ are the functions that relate to service f, and also consider if $(\cdot)$ and cf $(\cdot)$ are convex functions. Consider that every service in data processing found in F is processed in a shared way that is either partially processed in a distributed way or not possible to process in a distributed manner.

3.5 Query Phase

This phase is composed of communication direction initiated from the user to Cluster Head (CH) via the sensing server (SS). It includes two messages: message from the user to SS and SS to CH. The first protocol message is designed on the user side and delivered to the SS. The query is encrypted with global public key pk_P and decrypted by the CH. Assume that the incoming data to be encrypted is specified as $m\epsilon\ G_T$. The random user samples $r \in Z_q$ and offers ciphertext $CT = (CT_1,\ \ CT_2)$ is expressed as in Eq. (4):

$$Enc_P\ (m) = (g^r,\ \ m..(pk_P)^r) = (g^r,\ \ m\ .\ Z^{pr}) \tag{4}$$

The essential factor while determining the actual protocol messages is any sort of key (asymmetric or symmetric) to the query responder (sensing device), i.e., the initiator is unauthenticated. The query responder has a public and global public key. Thus, initiator uses the public key to set some secure channel for response with the encryption of random key K with the public key over the sensing platform (query Q). Thus, the message protocol known as M_1 produced by the user, i.e., query encryption Q, and random key K is utilized for sensing the response.

3.6 Query Response Phase

It possesses the decryption and query delivery of the actual recipient and the user response. Here, the CH_i receive message M_2 composed of ciphertext $CT^1 = \left(CT_1',\ \ CT_2'\right)$ and decrypts the secret key as given below in Eq. (5):

$$Dec_i\ (CT') = CT_2'.\ \left(CT_1^1\right)^{-1/sk_i} = m\ .\ Z^{pr}(Z^{prx_i})^{-1/x_i} = m \tag{5}$$

The output m is parsed as the original query Q and response key K. The cluster delivers message m to the destination; however, it is subjected towards the traffic analysis where the transmission is inspired from the meta-heuristic optimization where an optimizer like Grey Wolf Optimization (GWO) works effectually is modelled to attain global optimizer and fulfils exploration and exploitation. All the CH uses the deterministic mapping function to select *the k* destination. This process receives an identifier and returns the set of k identifiers. Thus, CH_i must either forward the incoming query encrypted with the sharing key to the k-sensing device with the mapping or query forward to $k-1$ destinations. The CH needs to share the secret key to encrypt the original message in some cases. Therefore, it avoids external attacks. After

reaching the k sensing device as an outcome, the device receives the query and replies with synthetic data. At last, the data aggregated by the CHs are relayed to CH, which initially receives the query and filters outs the message. The CH encrypts it using key K to offer message M_3; subsequently, the sensing device needs to encrypt the response provided the CH transfers the key K along with the query. At last, the message M_3 is provided to the SS and message is forwarded to the user. The user needs to receive the message M_4 from the sensing server and decrypts the message using key K to retrieve the response from the generated query. Fig. 2 depicts the flow diagram of the proposed model.

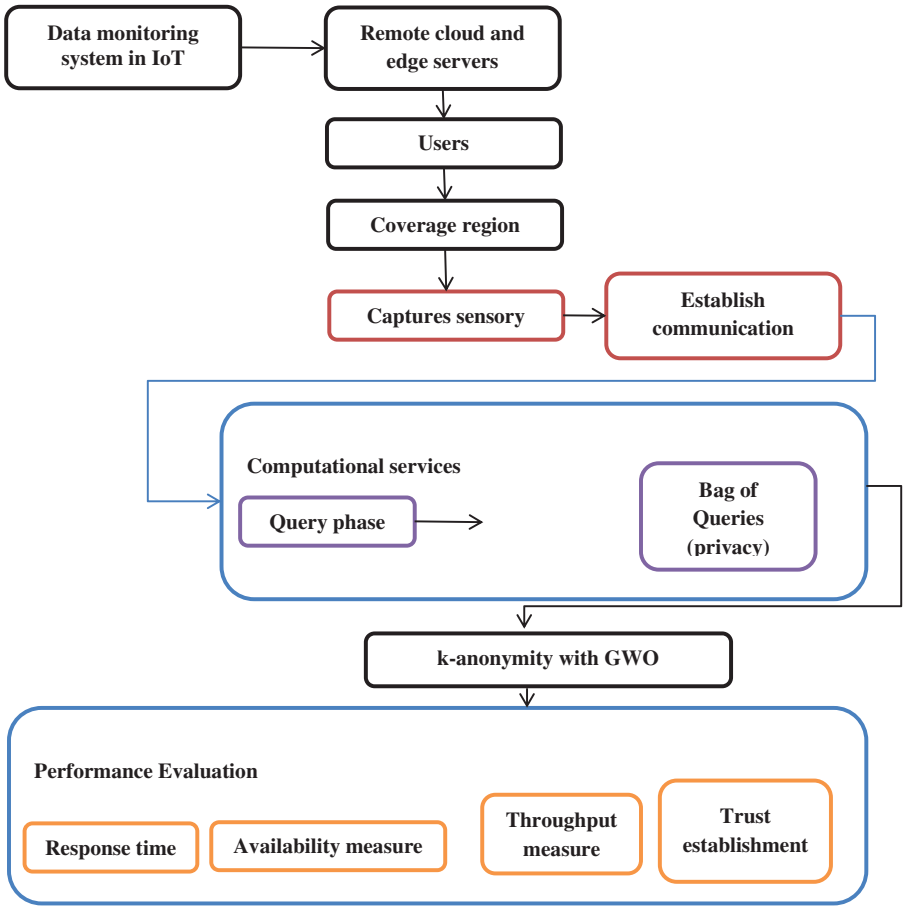

Figure 2: Flow diagram of proposed model

3.7 BoQ for Query Privacy

The SS cannot acquire information from content analysis; however, it needs statistical analysis. In the query privacy phase, the sensing server needs to learn the external colluders where the user generates a query to the connected nodes k to the query response. When k is larger sufficiently, query privacy is fulfilled with the mapping and $l-$diversity/$t-$closeness properties. The mapping does not the notions and the attackers are not aware of query response; however, it learns the users need. The queries are resolved to various devices with single execution. Moreover, when the user is querying a device and the SS regularly, the adversary cannot determine the elements in the anonymity set. When the SS intends to be cheated by the chosen CH, it cannot diminish the anonymity set as every CH adopts a similar mapping function. The sensing server alternates its queries (incoming bag of queries) and submits them to the successive network. This process is possible as this research considers only the public sensing network. The attacker learns the

mapping from a specific node is not a secret and the node senses every data. The incoming data is not sensitive as they are acquired from the query provided by the SS and not specific user. Finally, the SS and devices are not collaborated due to the network model, i.e., the user identity is well-known. The user identity is protected with network anonymity to diminish the risk of privacy exposure.

3.8 k-Anonymity with GWO

Here, Grey-wolf Optimizer is applicable for the IoT environment to establish privacy, QoS-aware trust service using two diverse mechanisms. They are storage analysis with Parento solution and Parento archive. The former provides a novel solution to control one or more IoT files, and the dominative or redundant incoming data is eliminated. Therefore, a new solution is offered to the archive. When the generated new solution and k-archive members are not dominated, the new solution is added to the archive. Similarly, when the archive is complete, the similarity among the dominative schemes is computed with the Euclidean distance with the prediction of two or more methods with only similarity and eliminating one amongst them. The similarity of the new solution is lesser to enhance the diversity of the final approximation. While the latter mechanism provides a unique solution in all iterations; therefore, it is essential to use the Parento archive to store the Parento solution. When the number of Parento solutions exceeds the number of services, it specifies the archive size, and the archive is clipped based on the crowding distance. Thus, it leads to an NP-hard problem.

The QoS-aware trust establishment in IoT is quantified based on the individual grey wolf and the prey in the multi-objective grey wolf, where the location corresponds to the service coordination scheme. The leader wolf role is to control the direction of grey wolves motion. Thus, it increases service coordination by fulfilling the service requirements and provides optimal solutions finally.

Algorithm 1:

Input: Online available web service dataset with service parameters; // availability, throughput and response time;

Output: Set of Parento solutions;

1. **Set** Gray wolves location;
2. **Initialize** the population X_i $(i = 1,\ 2,\ \ldots,\ n)$
3. **Compute** the search agent fitness and initialize archive size;
4. **Compute** alpha, beta and gamma of GWO; //$(\alpha,\ \beta,\ \gamma)$
5. Include alpha and beta to the available archive size;
6. Set T *as* 1;
7. **While** $(t < maximal\ number\ of\ iterations)$
8. **For** all search
9. Update the position of wolves; //$(X \propto,\ X\beta,\ X\gamma)$
10. **End for**
11. Evaluate objective functions of all agents;
12. Predict non-dominant solutions;
13. **Update** archive //$a,\ A\ and\ C$
14. **if**-archive is full
15. Perform similarity mechanism to eliminate the archive members; //$t = t + 1$

(Continued)

Algorithm 1 (continued)

16. Include the new solution to the archive;
17. **end if**
18. **if** any solution added to the archive
19. Update the novel solutions;
20. **End if**
21. Update alpha, beta and gamma;
22. Add alpha and beta to the archive; $//\left(X_{\propto},\ X_{\beta,}X_{c}\right)$
23. Set $T = T + 1$;
24. **End while**
25. **Return;**

4 Numerical Results and Discussion

This section describes the performance evaluation of the suggested algorithm using extensive simulations. The execution of the proposed algorithms on various situations has been carried out in a MATLAB simulation environment. The multiple edge servers differ from 5 to 30. Let C denote the average capability of edge server computation. The C value is assumed with the domain of [10, 40 GHz]. The edge server capability calculation is assigned in a random way based on average value C. Further, the analysis on the cloud capability E_0 is assumed to be five times bigger than the edge servers. The data rate among every two servers differs from 200 to 1000 Mbps is assumed. The multiple data sources $|SQ|$ and the services $|FQ|$ varies from 2 to 8, and 4 to 14 is accepted. The specific functions cf (·) and gf (·) refer to the linear functions for every $f \in FQ$, which are created randomly. Assume $|S|$ be the average data set types. Consider $|S|$ differs from 20 to 70 GB in the simulations. The system's parameters incorporate the multiple edge servers, multiple services, multiple data sources, every data set's average size, average capability calculation, and latency in transmission. The investigation of various system parameters is carried out in the suggested algorithm. The upper/lower bound of latencies are considered. Figs. 3–8 depicts the node creation and cluster formation in the IoT environment.

This work creates the various criteria in the set of simulations and differ the multiple servers (5 to 30) with average capacity of 20 GHz; the average rate of data of every communication is 50 F. The execution of the suggested algorithm on these kinds of circumstances computes the lower/upper bound of the latency in query processing. The latency occurred with the proposed algorithm is 50.3% reduced and better other approaches. There is calculation of resources if there are huge edge servers over the network environment. Henceforth, the performance of the anticipated algorithm is nearer to the lower bound (30 edge servers). Moreover, the proposed algorithm's latency is 2.20 times higher than the lower bound (five servers). However, the upper bound of latency has no remarkable change if the size of $|E|$ maximizes since the centralized method for query processing is not significantly correlated (See Tabs. 1–4). Some metrics like response time, trust establishment, throughput and node availability are evaluated and graphically represented in Figs. 9–12.

Tab. 1 depicts the comparison of the response time. The comparison is done for successive iterations like 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4 and 4.5 respectively. The response time of BoQ-IQWO during 4.5 iteration is 2.6 s which is 1.0, 6.4, and 1.8 s lesser than MODA, PSO and MOGWO. The availability of BoQ-IGWO is 7.5 which is 0.5, 3 and 1.8 higher than other approaches. The throughput of the anticipated model is

6 which is 2.4, 2.5, and 1.7 higher than other approaches. The trust of the anticipated model is 6.2 which is 2.4, 2.3 and 0.4 higher than other approaches.

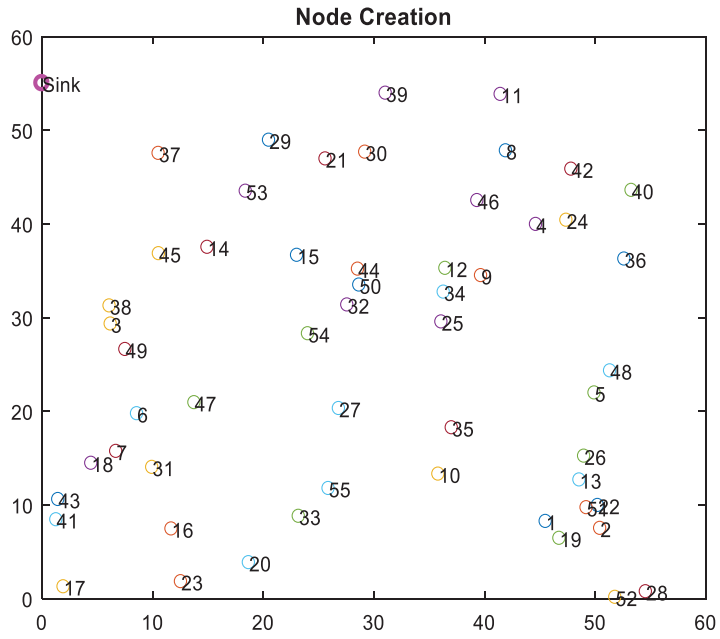

Figure 3: Node creation

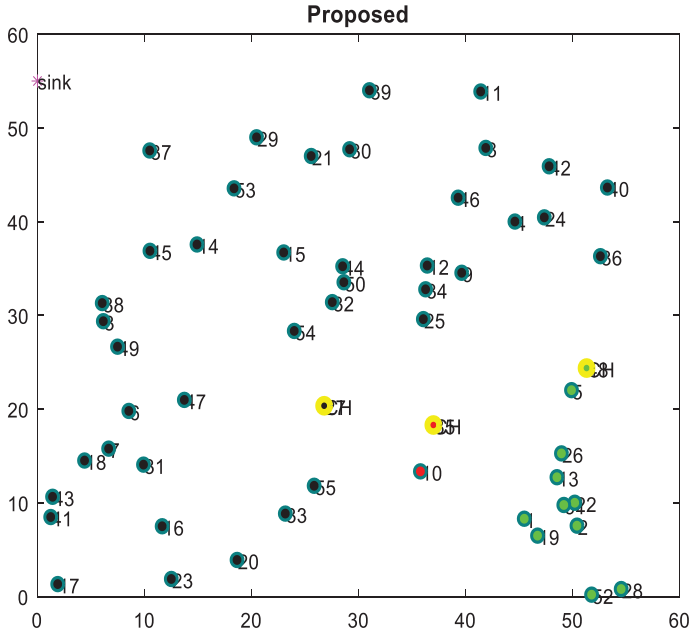

Figure 4: Cluster formation

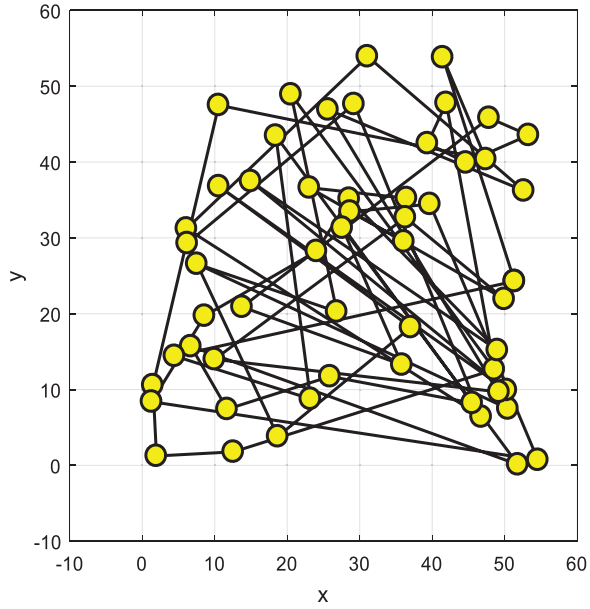

Figure 5: Node connectivity

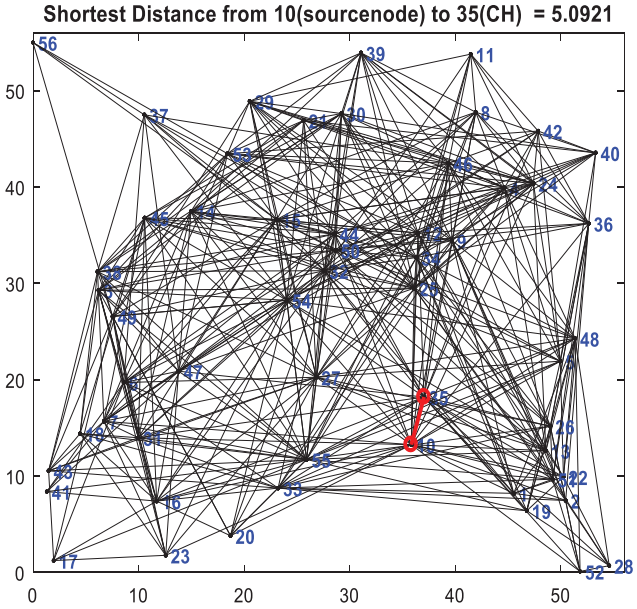

Figure 6: Distance measure from source to CH = 5.0921

Table 1: Response time comparison

Approaches	Iterations									
	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5
MOGWO	4.1	4	4	4	4	4.5	3.2	3.4	4.4	4.4
PSO	9.2	9	8	8.1	8.2	8	8.5	8.4	8.6	9
MODA	2.1	2.1	2.5	2.4	3	2.8	3	3.2	3.5	3.6
BoQ-IGWO	1.1	1.2	1.3	2.1	2.2	2.2	2.4	2.5	2.5	2.6

Table 2: Availability measure comparison

Approaches	Iterations									
	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5
MOGWO	5.4	5.4	5.5	5.5	5.5	5.5	5.6	5.6	5.6	5.7
PSO	5.7	5.5	5.7	5.8	5.5	5.8	5.2	5.9	4.7	4.5
MODA	5.8	6.3	6.3	6.3	6.3	6.5	6.6	6.7	6.8	7
BoQ-IGWO	6.5	6.5	6.6	6.7	6.8	6.9	7.2	7.3	7.4	7.5

Table 3: Throughput comparison

Approaches	Iterations									
	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5
MOGWO	3.4	3.4	3.4	3.5	3.5	3.5	3.6	3.6	3.6	3.6
PSO	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5	3.5
MODA	3.2	3.3	3.4	3.6	3.5	3.5	3.5	4.2	4.2	4.3
BoQ-IGWO	3.8	4.3	4.2	4.5	4.6	5.2	5.4	5.6	5.8	6

Table 4: Trust establishment comparison

Approaches	Iterations									
	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5
MOGWO	3.1	3.2	3.1	3	3.3	3.4	3.5	3.6	3.7	3.8
PSO	3	3	3	3	3.2	3.3	3.4	3.5	3.8	3.9
MODA	2.4	2.5	3.2	3.4	3.5	3.6	4.2	4.4	5	5.8
BoQ-IGWO	3	3.4	3.5	3.6	3.8	4.2	5.2	5.4	5.6	6.2

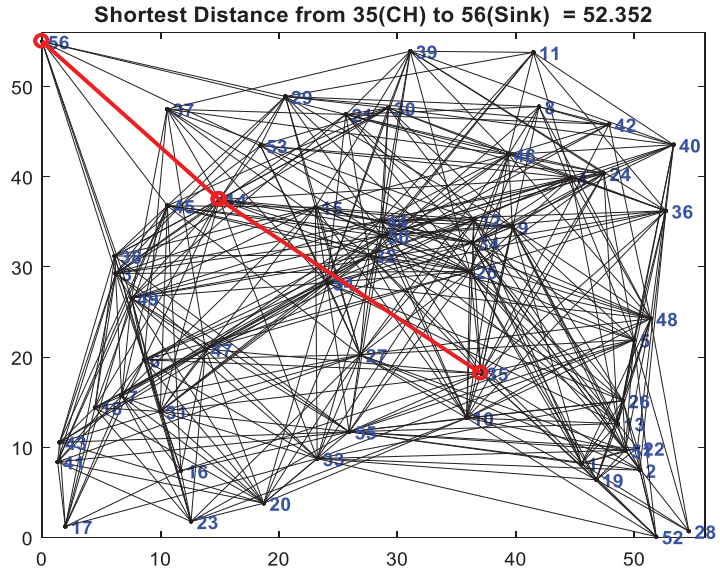

Figure 7: Distance measure from source to CH = 52.352

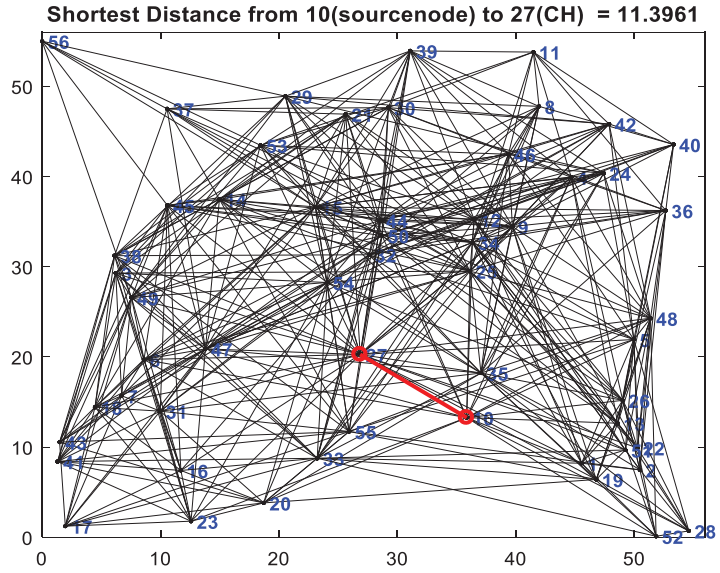

Figure 8: Distance measure from source to CH = 11.3961

Consider the situation with ten edge servers in the group of simulations that differs the multiple services in every acyclic graph from 4 to 14. The system parameters values are identical to the mentioned set of simulations. The specific number of services is created randomly, depending on the suggested algorithm. The proposed algorithm acquires the latency is 43.4% smaller than the upper bound, 3.05 times greater than the lower bound is considered. The ratio between the developed and optimal latency is smaller than the approximation ratio in the proposed algorithm. The upper bound maximizes the maximum number of services since the capability calculation of the cloud is stronger. The acquired latency maximizes from 174 to 372 s if the amount of services differs from 4 to 14 in the proposed algorithm. Further, the lower bound has no remarkable change if $|FQ|$ increases since the subset of services provide the lower bound's latency. The size of $|SQ|$ differs from 2 to 8 in the set of simulations. The other system parameters'

settings are identical to the settings of system parameters in the previous simulations. The proposed algorithm has the queries that need to be processed and demonstrates the average latency in query processing. It is shown that the acquired latency in query processing is 42.7% smaller than the upper bound depicted in the proposed algorithm. It is important to note that the amount of data sources grows if the acquired latency in query processing decreases in the proposed algorithm. The cause of this scenario is, the dependence between services gets weaker if the size of SQ is more prominent and the query is more suitable to be processed in a shared way. The effect of the average size of every data set is the size of every data set that identifies the latency in transmission among two servers. These are the key parameters that affect the latency in query processing. Assume, there are four sensed data sets belong to diverse data sources $\{S(1),\ S(2),\ S(3),\ S(4)\}$ and adjusting the average size of data sets differs from 20 to 70 GB. However, the average capability calculation of every edge server is 20 GHz, the average data rate of every communication link is 1000 Mbps, the amount of services and the number of edge servers is 10. After executing and computing the lower and upper bound latencies, the relationship between the data set size and the acquired latency in query processing are considered. The proposed algorithm helps to achieve the latencies of the lower bound and upper bound in query processing that increases when the average data set size gets higher. The latency of query processing rises by 4.4 times if the size differs from 20 to 70 GB. The size of data creates a significant effect since the proposed algorithm processes the query in a distributed way. Moreover, the suggested algorithm is even better than the centralized data processing method. The performance is 32% superior to centralized method or upper bound of data processing obtained.

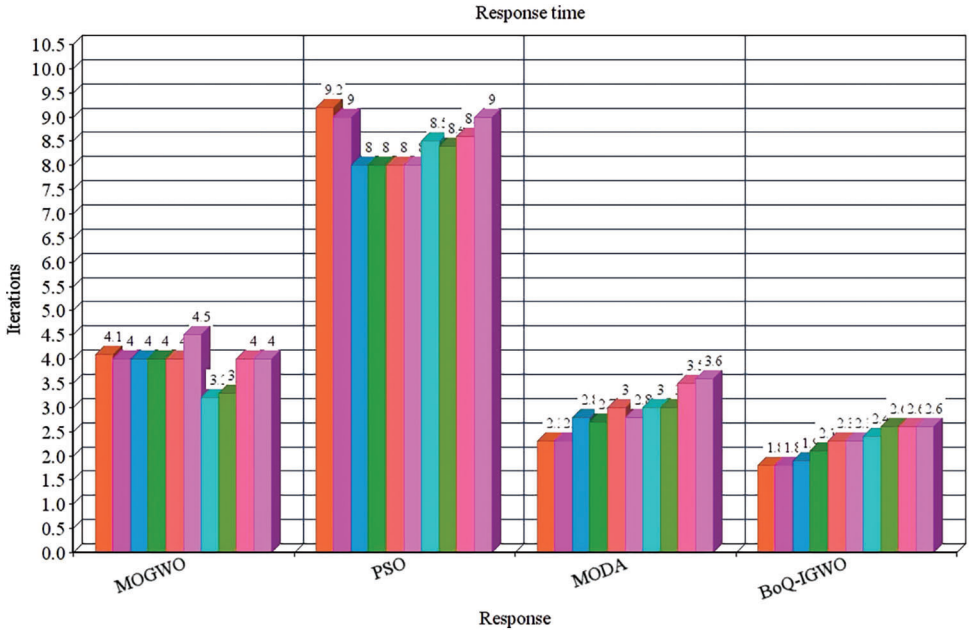

Figure 9: Response time comparison

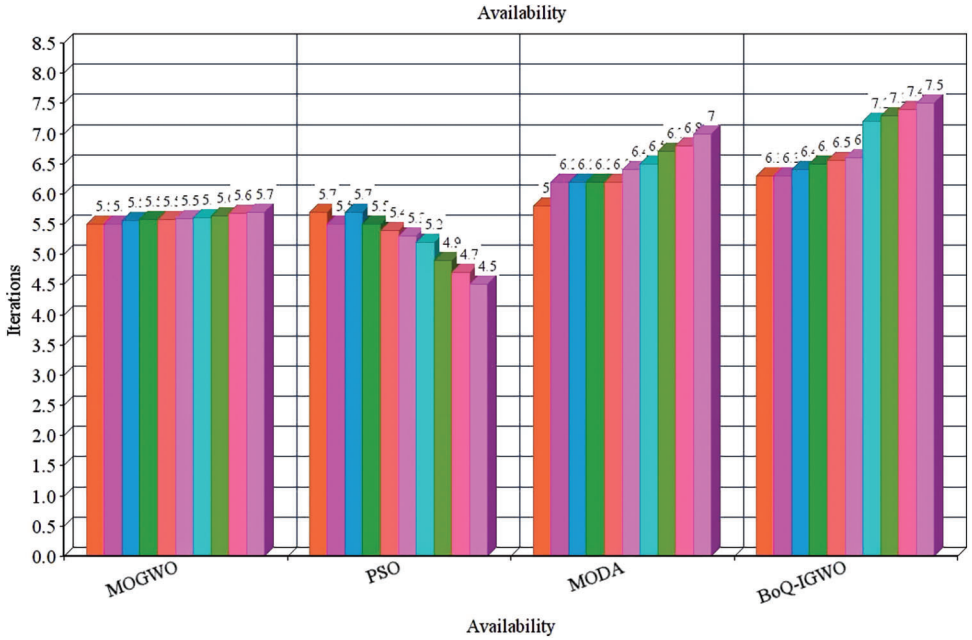

Figure 10: Availability measure

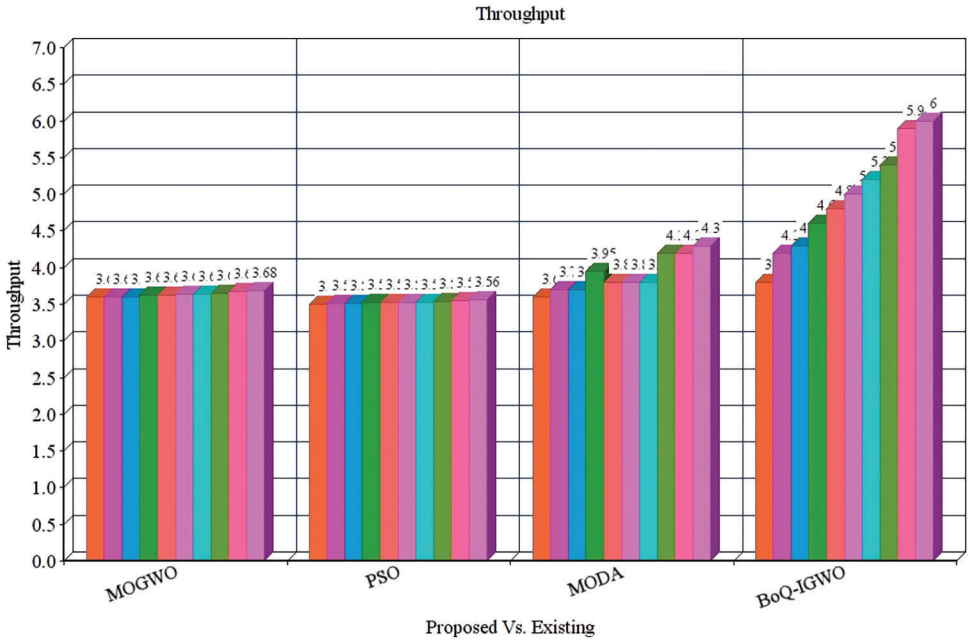

Figure 11: Throughput comparison

The capability calculation of every edge server identifies the latency in analysis and also influences the latency of query processing. The ten edge servers should be deployed as $E1$, $E2$, $\ldots E10$ in the set of simulations. The capability calculation of every edge server is generated in a random way that depends on the average value of C where $C = 110\ P\ 1 \leq i \leq 10\ C_i$. The c values can differ from 10–40 GHz. However, assuming that the capacity calculation is five times greater than the average, then the amount of

data sources and the services are 10 and 4 correspondingly. The latency calculation of every server gets minimized if the capability calculation gets higher. Lower and upper bound latencies decrease in the proposed algorithm by considering the latency of query processing. The latency in processing the query minimizes 24% in the proposed algorithm if the capability calculation maximizes from 10 to 40 GHz. Further, the proposed algorithm produced a latency of 29.7% less than the upper bound.

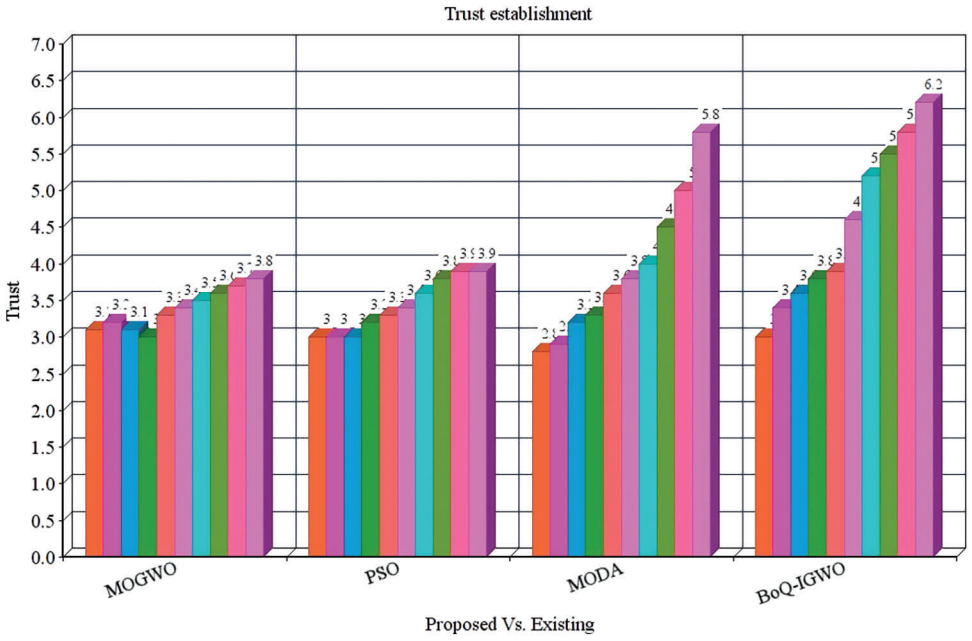

Figure 12: Trust establishment

The rate of data transmission greatly influences the latency of query processing. The data transmission rate depends on the 5G network in the group of simulations. Assume, an average data transmission rate differs from 200–1000 Mbps, the latency to transmit 1GB data differs from 1 to 5 s. Various situations are generated and execute the suggested algorithm that depends on the average data transmission rates. The latency and upper bound are maximized remarkably if the data rate gets lower. Moreover, the lower bound maximizes gradually if the data rate is minimized since the services are considered to be carried out entirely in a distributed way. Further, the acquired latency is 35.7% lesser than the upper bound in the proposed algorithm if the data rate grows from 200 to 1000 Mbps, and the latency gets increased almost five times in the proposed algorithm. As a result, the latency of every communication link significantly influences the latency during data processing.

5 Conclusion

The proposed system describes that the distributed query processing issue is investigated in the distributed system. The problem in query processing is defined that targets to derive the query processing plan with less latency of query response. The minimal issue in query processing is NP-Hard. Primarily, exceptional cases with fewer query processing issues are investigated and confirmed. The optimal solution for the minor problem in query processing in polynomial time is obtained for these two cases. Hence, an approximation algorithm is proposed to resolve the minor issue in query processing and gives the analysis that this algorithm creates a suitable approximation ratio. The simulation outcomes are illustrated to determine the algorithm's performance, and the simulation outcomes implicit that this

algorithm is efficient. This study concentrates on reducing latency, and our future research is to examine the energy-efficiency algorithms in a distributed management system.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] X. Zeng, G. Xu, X. Zheng, Y. Xiang and W. Zhou, "E-AUA: An efficient anonymous user authentication protocol for mobile IoT," *IEEE Internet Things Journal*, vol. 6, no. 2, pp. 1506–1519, 2019.

[2] R. Paulet, M. G. Kaosar, X. Yi and E. Bertino, "Privacy-preserving and content-protecting location-based queries," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 5, pp. 1200–1210, 2013.

[3] T. Ma, J. Jia, Y. Xue, Y. Tian, A. Al-Dhelaan *et al.,* "Protection of location privacy for moving kNN queries in social networks," *Applied Soft Computing*, vol. 66, pp. 525–532, 2018.

[4] M. Elkhorn, S. Shahrestani and H. Cheung, "A review of mobile location privacy in the internet of things," in *Proc. 10th Int. Conf. on ICT and Knowledge Engineering*, Bangkok, Thailand, pp. 266–272, 2012.

[5] J. Zhou, Z. Cao, X. Dong and A. V. Vasilakos, "Security and privacy for cloud-based IoT: Challenges," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 26–33, 2017.

[6] X. Pan, J. Xu and X. Meng, "Protecting location privacy against location-dependent attacks in mobile services," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 8, pp. 1506–1519, 2011.

[7] C. Zhou, T. Wang, W. Jiang and H. Tian, "Practical k nearest neighbor query scheme with two-party guarantees in road networks, " in *Proc. 17th IEEE Int. Conf. Trust, Security and Privacy in Computing and Communications, 12th IEEE Int. Conf. on Big Data Science and Engineering (TrustCom/BigDataSE)*, New York, USA, pp. 1316–1321, 2018.

[8] T. Wang, M. Z. A. Bhuiyan, G. Wang, L. Qi, J. Wu *et al.,* "Preserving balance between privacy and data integrity in edge-assisted internet of things," *IEEE Internet Things Journal*, vol. 7, no. 4, pp. 2679–2689, 2020.

[9] T. Tao, M. Z. A. Bhuiyan, M. A. Rahman, T. Wang, J. Wu *et al.,* "TrustData: Trustworthy and secured data collection for event detection in the industrial cyber-physical system," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3311–3321, 2019.

[10] T. Wang, H. Luo, X. Zheng and M. Xie, "Crowdsourcing mechanism for trust evaluation in CPCS based on intelligent mobile edge computing," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 6, pp. 1–19, 2019.

[11] T. Wang, D. Zhao, S. Cai, W. Jia and A. Liu, "Bidirectional prediction based underwater data collection protocol for end-edge-cloud orchestrated system," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4791–4799, 2019.

[12] Q. A. Arain, I. Memon, Z. Deng, M. H. Memon, F. A. Manga *et al.,* "Location monitoring approach: Multiple mix-zones with location privacy protection based on traffic flow over road networks," *Multimedia Tools and Applications*, vol. 77, no. 5, pp. 5563–5607, 2018.

[13] W. Eltarjaman, R. Dewri and R. Thurimella, "Private retrieval of POI details in top-K queries," *IEEE Transactions on Mobile Computing*, vol. 16, no. 9, pp. 2611–2624, 2016.

[14] X. Meng, H. Zhu and G. Kollios, "Top-k query processing on encrypted databases with strong security guarantees," in *Proc. IEEE 34th Int. Conf. on Data Engineering (ICDE)*, Paris, France, pp. 353–364, 2018.

[15] K. A. Hafeez, L. Zhao, J. W. Mark, X. Shen and Z. Niu, "Distributed multichannel and mobility-aware cluster-based MAC protocol for vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 8, pp. 3886–3902, 2013.

[16] T. Wang, J. Zeng, Y. Lai, Y. Cai, H. Tian *et al.,* "Data collection from WSNs to the cloud-based on mobile fog elements," *Future Generation Computer Systems*, vol. 105, pp. 864–872, 2020.

[17] Y. K. Wu, H. Huang, Q. Wu, A. Liu and T. Wang, "A risk defence method based on microscopic state prediction with partial information observations in social networks," *Journal of Parallel and Distributed Computing*, vol. 131, pp. 189–199, 2019.

[18] B. Zhou, J. Li, X. Wang, Y. Gu, L. Xu *et al.,* "Online internet traffic monitoring system using spark streaming," *Big Data Mining and Analytics*, vol. 1, no. 1, pp. 47–56, 2018.

[19] C. Wang, C. Wang, Z. Wang, X. Ye, J. X. Yu *et al.,* "DeepDirect: Learning directions of social ties with edge-based network embedding," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2277–2291, 2018.

[20] Y. Xiao, Y. Jia, C. Liu, X. Cheng, J. Yu *et al.,* "Edge computing security: State of the art and challenges," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1608–1631, 2019.

[21] S. Cheng, Z. Cai and J. Li, "Curve query processing in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 11, pp. 5198–5209, 2014.

[22] T. Shi, S. Cheng, Z. Cai and J. Li, "Adaptive connected dominating set discovering algorithm in energy-harvest sensor networks," in *35th IEEE INFOCOM*, San Francisco, CA, USA, pp. 1–9, 2016.

[23] X. Chen, L. Jiao, W. Li and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2015.

[24] X. Cao, F. Wang, J. Xu, R. Zhang and S. Cui, "Joint computation and communication cooperation for mobile edge computing," in *16th Int. Symp. on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, Shanghai, China, pp. 1–6, 2018.

[25] K. Poularakis, J. Llorca, A. M. Tulino, I. Taylor and L. Tassiulas, "Joint service placement and request routing in multi-cell mobile edge computing networks," in *IEEE Int. Conf. on Computer Communications*, Paris, France, pp. 10–18, 2019.

[26] W. Sun, G. Z. Dai, X. R. Zhang and X. Z. He, "TBE-net: A three-branch embedding network with part-aware ability and feature complementary learning for vehicle re-identification," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2021. https://doi.org/10.1109/TITS.2021.3130403.

[27] W. Sun, L. Dai, X. R. Zhang, P. S. Chang and X. Z. He, "RSOD: Real-time small object detection algorithm in UAV-based traffic monitoring," *Applied Intelligence*, pp. 1–16, 2021. https://doi.org/10.1007/s10489-021-02893-3.

[28] P. Muneeshwari and M. Kishanthini, "A New Framework for Anomaly Detection in NSLKDD Dataset using Hybrid Neuro-Weighted Genetic Algorithm," *Journal of Computational Science and Intelligent Technologies*, vol. 1, no. 1, pp. 29–36, 2020. https://doi.org/10.53409/mnaa.jcsit1105.

[29] R. Mugesh, "A Survey on Security Risks in Internet of Things (IoT) Environment," *Journal of Computational Science and Intelligent Technologies*, vol. 1, no. 2, pp. 1–8, 2020. https://doi.org/10.53409/mnaa.jcsit20201201

[30] H. Q. Alatawi, S. F. Aluneizi, A. S. Makki, M. M. Alshamrani and N. M. Albalawi, "An Effectiveness of AI Approaches in Human Disease Diagnosis for Increasing Efficiency of Medical Systems- Review," *Journal of Computational Science and Intelligent Technologies*, vol. 2, no. 3, pp. 1–8, 2021. https://doi.org/10.53409/mnaa/jcsit/2301.S.

[31] S. P. Sasirekha, A. Priya, T. Anitha and P. Sherubh, "Data Processing and Management in IoT and Wireless Sensor Network," *Journal of Physics: Conference Series*, vol. 1712, no. 1, pp. 012002, 2020. https://doi.org/10.1088/1742-6596/1712/1/012002.

[32] R. Khilar, K. Mariyappan, M. S. Christo, J. Amutharaj, T. Anitha *et al.*, "Artificial Intelligence-Based Security Protocols to Resist Attacks in Internet of Things," *Wireless Communications and Mobile Computing*, vol. 2022, no. 1440538, pp. 1–10, 2022. https://doi.org/10.1155/2022/1440538.