Journal of Applied Research and Technology

www.jart.icat.unam.mx

Journal of Applied Research and Technology 19 (2021) 66-76

Original

QoS-aware CR-BM-based hybrid framework to improve the fault tolerance of fog devices

P. Sharma • P. K. Gupta*

Department of Computer Science and Engineering,
Jaypee University of Information Technology, Solan HP India

Received 05-02-2020; accepted 01 21 2021
Available online 02 28 2021

Abstract: With the evolution of the Internet of Things (IoT), the use of smart devices has completely changed the day-to-day life of the human being. IoT devices are of flexible use which is implemented to sense the environment and data efficiently. However, these devices have some constrained capabilities concerning fault tolerance, computation cost, and storage. This requires an improved framework and algorithms for performing effective operations. In this paper, a hybrid framework is proposed, which incorporates the various IoT devices in fog environments to enhance fault tolerance. The proposed framework implements a novel QoS-aware technique based on the combination of checkpoints and replication (CR) for diagnosing faults and the bee-mutation (BM) algorithm for optimal placement of service. A fog service monitor is established to observe the performance of fog nodes. Both the CR module and BM module access the service monitor to ensure that the proposed hybrid framework is fault-tolerant. Furthermore, the proposed CR-BM-based hybrid framework has been evaluated for its performance by using various performance metrics. In the comparative analysis, it is observed that the proposed hybrid framework outperforms the existing genetic algorithm-based framework.

Keywords: IoT, QoS, fault tolerance, checkpoint, replication, bee-mutation, service placement

*Corresponding author.
E-mail address: pkgupta@ieee.org (P. K. Gupta).
Peer Review under the responsibility of Universidad Nacional Autónoma de México.

1. Introduction

The people in the technological world are getting more inclined to use smart devices extensively in their day to day activity. The introduction of IoT has revolutionized the usage of these smart devices with effective collection and exchange of data (Nagaraj, 2021). The increase of IoT in various applications is evident as it supports advanced automation and improves the life of human beings. Most of the IoT frameworks enclose sensing devices that monitor the environment it is established (Sethi & Sarangi, 2017). The usage of IoT in an automated environment monitors the complex systems. Hence, there should be a module to detect the fault accurately in the automated system. This fault diagnosing model may save the systems and humans from any catastrophic events(Lo et al., 2019). Among various existing IoT applications, several industrial companies are now adopting IoT to improve their operations. One major use of IoT systems in industry is for monitoring, fault detection, and diagnosis (MFDD) of the operations of the systems to ensure their proper operations (Yen et al., 2017).

As we know, IoT devices have limitations of As we know, IoT devices have limitations of constrained energy, and memory utilization for computation and storage respectively. Therefore, the use of these devices depends upon external paradigms for their enhanced performance. Many researchers have employed cloud computing along with IoT to obtain better performance. However, the recent trend is more towards the use of a hybrid framework of IoT and fog with cloud computing. The use of fog computing provides greater significance to the computation of data and its storage. The other significant characteristics of Fog computing is the reduction of response delay and latency (Yi et al., 2015). Unlike the traditional frameworks, the IoT-fog system consists up of three major parts as fog nodes, IoT devices, and cloud as shown in Figure 1.

The scalability of the fog system provides an increased probability of failures. Under certain cases, hardware faults and software bugs can affect the system's reliability and performance (Garraghan et a., 2019). Along with the scalability, the fog system also encloses complexity and heterogeneity issues that may result in various combinations of faults (Yang et al., 2017). Therefore, to address these issues user-transparent and redundant replications, some effective fault-tolerant deployment and execution approaches are required. In general, two types of fault tolerance known as reactive fault tolerance and proactive fault tolerance are established. In which, reactive fault tolerance reduces the failure impacts and the proactive method identifies the suspected component and replaces it before the occurrence of any failure. In this paper, a novel fault tolerance detection system has been established with the reactive fault tolerance method and the bee-mutation algorithm which is the combination of both the artificial bee colony and genetic algorithm.

The following efforts are made to achieve proper implementation of the proposed fault tolerance detection system as:

- The checkpoint and restart module facilitate the task which is failed and restart it from the point of failure by using the hybrid framework.
- The replication approach is employed to generate task replicas on various resources until the complete task gets fail.
- The placement of service is optimized with the use of the bee-mutation algorithm.

The remaining paper is organized in the following sequence as section 2 encloses the related works on fault tolerance of fog systems, section 3 provides the preliminaries of the approaches used in the context of the proposed framework, section 4 implements the proposed hybrid framework along with novel CRBM algorithms, section 5 provides the detailed result analysis and compares the performance of the proposed framework with existing genetic algorithm. Finally, section 6 concludes the work and provides the future scope

2. Related work

This section summarizes the work of various researchers who have addressed the issue of reliability in the hybrid environment of IoT and fog.

Wang et al. (2020), proposed a mechanism for enhancing the data transfer reliability of fog computing for the health care domain. The proposed mechanism has been established with the help of limited flooding and directed diffusion methods. Also, this mechanism deploys an adoption scheme to control the parameters based on the present condition of fog.

Ozeer et al. (2018) proposed a four-step protocol to manage the fault tolerance in the system. Using the integrated checkpoint and message logging, the proposed protocol saves the service state of the system. This protocol observes the surroundings of the system and generates reports about failures. In case of any failure, the protocol takes the necessary steps to prevent from along with proper decision making. If failures take place, the protocol initiates the reconfiguration of the dependent entities with proper recovery actions. Oma et al. (2018) discusses another fault-tolerant technique, established over the fog framework and uses tree structure. The proposed approach encloses the usage of both replication and non-replication methods. The replication approach enables the processing of similar request through multiple fog nodes whereas the non-replication technique replaces the faulty fog node with a new one.

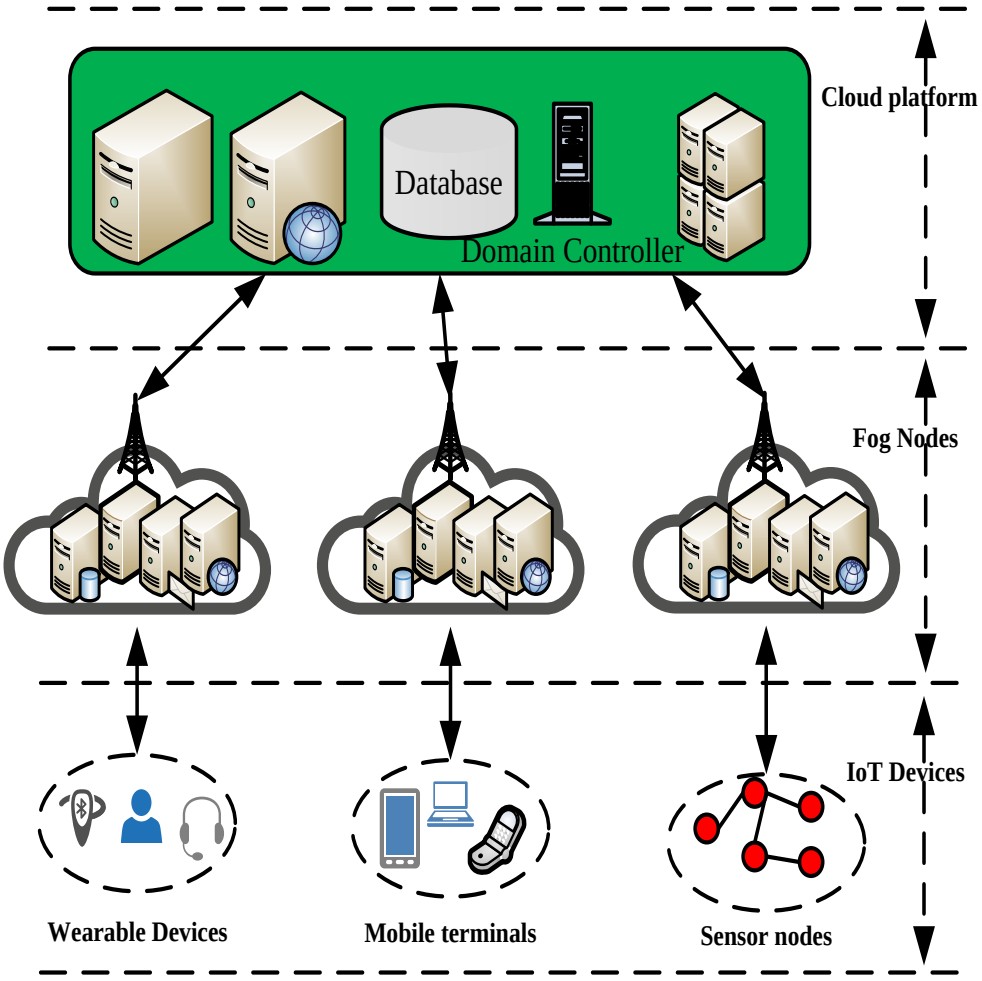

Figure 1. Infrastructure of IoT -fog framework (Gupta & Gupta, 2020).

Alarifi et al. proposed a fault tolerant aware scheduling method that reduces the overheads of service, latency, and increases the reliability of system. The method rely on classifying the devices that can send requests into three classes depending upon the type of services required. These classes are further categorized into time-sensitive, time-tolerant and core class.

Iotti et al. (2017) developed a fog-based network with internet access to assist the cloud or web content placement at edge network dynamically. The proposed model supports proactive storing and traffic policies implementation that result in network infrastructures to communicate smoothly with external applications. In the obtained result, the proposed model shows the better network optimization, QoE, and latency for authenticated users.

Brogi and Forti, (2017), proposed a placement policy along with a QoS-aware application. This policy is based on infrastructure processing speed and responsiveness along and focus on cost issue. Proposed policy locates the multi-component of IoT applications in a hierarchical fog environment. Jiang et al. (2012), addressed the problem of minimizing the resource in route selection and virtual machine placement by using the Markov approximations in data centers. This algorithm initially estimates the virtual machines placement and subsequently chooses the routes among them. The central controller has been established to perform all computations by tracking the best global placement in the system.

Cardellini et al. (2016), addressed the problem of operator placement in data streampProcessing (DSP). The DSP

operator allocation considers the fog nodes along with the applications of optimizing the quality of service (QoS) goal. The DSP placement is further optimized with an integer linear programming (ILP) problem.

Mebrek et al. (2017), proposed an objective assignment problem to address the problem of assigning the fog nodes to IoT devices through the evolutionary algorithm that minimizes the energy consumption in delay constrained mobile environment. They also used several meta-heuristic algorithms like Genetic Algorithm (GA), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO) in the systems that are facilitated with for distributed and dynamic computing.

Canali, C et al. presented an optimization model for mapping of data model over fog nodes which is based on a genetic algorithm and considers the communication latency and existing load between sensor and fog nodes.

Rajalakshmi et al. (2014), proposed a QoE-optimized scheduler for the multiclass system in wireless network. The end-user QoE has been modeled with cost function established on mean flow delay that arranges the requests on service accordingly. Additionally, the policy reports the allocation of the resource over various classes with the sensitivity to flow delay.

Baranwal et al. (2020), used quality of experience (QoE) to measure the fulfillment of the user agreements with respect to quality of the service aspect. Here, QoE monitors stakeholder privileges and sensitivity in fastidious situations

3. Preliminaries

3.1. Checkpoint and restart

In this technique, different checkpoints are established either at the system level or at the application level. The fundamental requirement is the knowledge about the failure instances in the system framework. The recovery from fault is initiated by resuming the task where it left-off.

3.2. Replication

Replication is the frequently employed technique for fault tolerance in storage devices like cloud and fog where failure usually takes place. It is generally employed to improve the availability of the resources in the environment of distributed storage. Major issues in replication are a selection of replica and its placement since the storage system is usually large and intricate in nature (Rajalakshmi et al., 2014).

3.3. Artificial bee colony

The artificial beecolony is an optimization algorithm that is nature-inspired by the foraging behavior of the honey bee. The new neighbor solution Si,j is obtained from the equation (1) as

$$S_{i,j} = N_{i,j} + \phi_{i,j}\left(N_{i,j} - N_{k,j}\right)_{(i\neq k)\ (\phi_{i,j}\epsilon\ (-1,1)} \quad (1)$$

By implementing the greedy selection, the value of Si and Ni are compared to update the solution that possesses better fitness value. The solution selection is based on the probabilistic model as shown in Eq (2):

$$P_i = \frac{F_i}{\sum_{j=1}^{SS} F_j} \quad (2)$$

Where F_i, F_j, are the fitness measure of i[th]and j[th]swarm solution (Xu et al., 2013).

3.4. Genetic algorithm

The GA is a well-known optimizer that operates over the coded chromosomes. The GA usually involves three significant operators' as given below (Amjad et al., 2018) The selection operator of GA takes the probability values as input, and the rank-based selection is employed to get the optimum fitness. However, the proposed method of the selection operation is performed with the artificial bee colony algorithm. The cross-over operator is performed on two fitness values and the cross-over probability ($\mathbf{CO_P}$)is obtained with Eq (3) as,

$$CO_P = \begin{Bmatrix} if\, F_i \leq F_a, then K_1 \frac{F_{max}-F_a}{F_{max}-F_i} \\ if\, F_i > F_a, then K_2 \end{Bmatrix} \quad (3)$$

Where $\mathbf{F_i}, \mathbf{F_a}$are the better of two solutions and average fitness are value in solution population respectively, and$\mathbf{F_{max}}$, is the maximum fitness value. The $\mathbf{K_1}$ and $\mathbf{K_2}$are the overall value that ranges from 0 to 1.

After the termination of the cross-over operator, the mutation operator is employed over its output. Similar to the cross-over the mutation probability (M_P) is estimated by using Eq (4) as

$$\mathbf{M_P} = \begin{Bmatrix} \mathbf{if F_j \leq F_a, then\ K_3 \frac{F_{max}-F_a}{F_{max}-F_j}} \\ \mathbf{if\ F_j > F_a, then\ K_4} \end{Bmatrix} \quad (4)$$

Where $\mathbf{F_j}$is the individual fitness of solution and $\mathbf{K_3}$ and $\mathbf{K_4}$are the overall value that ranges from 0 to 1.

4. Proposed methodology

The proposed CR-BM-based hybrid framework determines the faults in the IoT-fog environment by placing the service as shown in Fig. 2. The proposed system encloses thevarious IoT devices and fog nodes together. Here, IoT devices form the layer that collects the data from the monitoring environment. In addition, the collected data are transferred to the fog nodes with the help of the service broker for performing further computation on it. This computation makes the data available to be processed further by the service and reallocation of the same as per the demand of users. The prime focus for ensuring the scalability and accessibility of data is carried out with effective balancing of load and proper allocation of the resources by using appropriate job scheduling. On the availability of data, as the fog nodes receive any request for transmission of data then the response is sent to the user after processing the available data. The processing of the request is carried out through the best available fog node deployed for IoT services. Selection of best fog node ensures the data integrity.

Therefore, to get quicker response from these fog nodes these nodes must be free from faults or their fault-tolerance should be higher. Proposed framework ensures better fault tolerance by employing two techniques. Here, the first approach is used to establish the checkpoints, whereas the second approach is used to generate different replicas for the existing fog nodes. The placing of service largely depends up on the several QoS parameters like utilization of CPU, RAM along with the processing speed and time. The other parameters like response time and delay can also be taken into account.

Therefore, to get a quicker response from these fog nodes these nodes must be free from faults or their fault-tolerance should be higher. The proposed framework ensures better fault tolerance by employing two techniques. Here, the first approach is used to establish the checkpoints, whereas the second approach is used to generate different replicas for the existing fog nodes. The placing of service largely depends upon the several QoS parameters like utilization of CPU, RAM along with the processing speed and time. The other parameters like response time and delay can also be taken into account. In the proposed CR-BM-based hybrid framework, the optimization of the QoS parameters is carried out using the bee-mutation algorithm. Also, the proposed model includes various IoT devices and fog nodes that constitute the structure of fog-based fault monitoring system.

The functions of the fog-based fault monitoring system are provided as follows:

a. IoT devices collect real-time data and send it to fog nodes for further processing and computation.

b. After computation, data is available for fulfilling the user's requests, and fog nodes perform reallocation with respect to the demand of users.

c. The processing of the request is carried out through the best node among all the available fog nodes that are employed for in the IoT services. The response is swiftly processed without any loss of data via the best node.

d. As per the context of fault tolerance, checkpoints are placed at the different intervals during the processing for analyzing the present status of the fog nodes. Replication technique along with checkpoint is used to protect the node from behaving incorrectly.

4.1. Diagnosing faults using the CR module

The proposed framework carries out the fault diagnosis by using the CR module. This module implements two different models for the detection of a fault. Firstly, checkpoints are placed at different intervals during the processing of fog nodes. These checkpoints analyze the present status of each fog node in the framework and keep the system operational despite any unexpected failure. These checkpoints also access the scheduler and resource server in the fog monitor. Here, the scheduler provides the information about the fog nodes necessary for responding and the resource server is used to freeze out the process under any failure like condition and to continue the process from the neighboring checkpoint. Mechanisms implemented by the CR module results in reduced utilization of resources with better time management. Algorithm 1 represents the implementation procedure for checkpoint and restart.

Secondly, the replication technique employed in the proposed fault diagnosis is a passive replica technique. In this technique, the fog nodes receive the request and form into the primary node for further processing of the request. This technique also deploys the secondary or backup fog nodes in the background along with the primary fog nodes. These backup nodes work continuously and in case any failure of the primary node takes place then the control is transferred to the respective backup node without interrupting the ongoing execution of the request. This replication technique is used along with the checkpointing technique that ensures the effective handling of fault without disrupting any ongoing processing. Algorithm 2 represents the implementation procedure for replication.

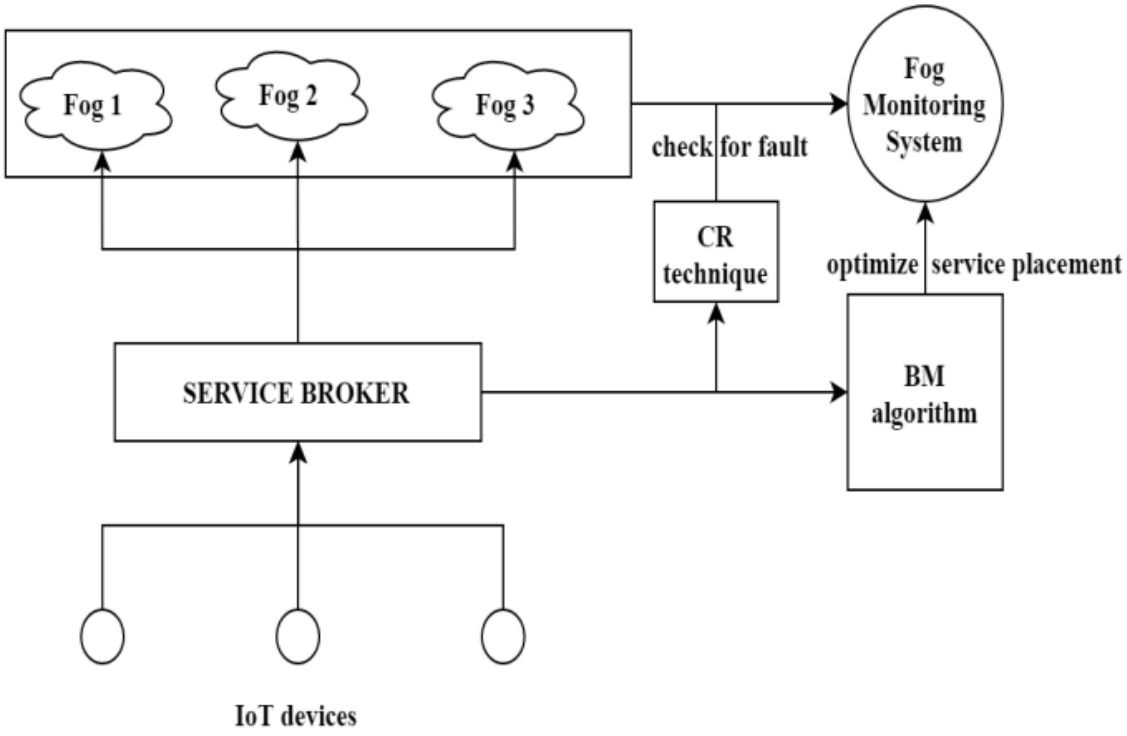

Figure 2. Proposed CRBM framework for fault monitoring using hybrid environment of IoT- fog.

Algorithm 1: Checkpoint and restart

Input: Data from the fog monitoring system
Output: Restart the process
1: Get the status of a node from the fog monitoring system
2: Survey it for inactive node
3: Get the replica ()
4: Process the node
5: end

Algorithm 1: Checkpoint and restart process

Algorithm 2: Replication

Input: Data from the fog monitoring system, K- acceptable failure rate
Output: Replica ( )
1: Get the status of a node from the fog monitoring system
2: Sort the nodes ascendingly based on finish time
3: Estimate the failure rate of each node (F)
4: if (0 ≤F≥ k)
5: initiate replication at one node
6: else
7: initiate replication at two nodes
8: Generate replica ()
9: End
1.1 Optimalserviceplacementwith BM

Algorithm 2: Replicationprocess.

4.2. Optimal service placement with BM

The next important consideration in the proposed framework is the placement of service with the effective fog nodes. For optimal placement of service, the bee-mutation algorithm has been proposed, as shown in Algorithm 3. The QoS parameters of the fog nodes like CPU usage, memory usage, network bandwidth, system time, and processing speed is taken as an input for evaluation of BM algorithm. These input parameters are initially fed into the artificial bee colony algorithm which generates the solutions for the bee algorithm using Eq. (1) and its probability fitness value is determined by using Eq. (2). After getting the probability fitness value, the cross over operation of the genetic algorithm is implemented over the obtained probabilities to get a new breed of values for each parameter of the fog nodes from Eq. (3). The mutation operator is executed with the mutation rate of 10% over the obtained cross overvalues by substituting them with the values obtained through Eq. (4).

The proposed algorithms are iterative in nature and are based on the method of steady expansion that achieve the objectives of a function. In each iteration, the algorithm calls for computation of the fitness functions as shown in Eq. (2). Furthermore, the cross over operation has been performed on two fitness value and the cross over probability which is computed by using Eq.(3), and similarly crossover mutation

probability ha.s been estimated by using Eq. (4). The running time complexity of check pointing and replication algorithms is O(n) and the complexity of iterative bases bee mutation algorithm is O(n2) .

Algorithm 3: Optimal service placement

Input: QoS parameters of fog nodes
Output optimized service placement among fog nodes
1: Load the QoS parameters of the fog node
2: Initialize the random vector from the population (R_i)
3: Estimate the fitness function of the random vector $F(R_i)$
4: For j < n, do
5: if $F(R_i) > F(R_j)$
6: Compute fitness function for next value F (R_{j+1})
7: else
8: change $F(R_i) = F(R_j)$
9: end if
10: end for
11: compute the probability of fitness value using Equation (2)
12: Perform cross over operation on the obtained probability of fitness value using the Equation (3)
13: Compute the mutation values through Equation (4)
14: Repeat step 12 and 13 until it reaches the maximum number of cycle
15: Get the optimized fog nodes for service placing
16: end

Algorithm 3: Optimal service placement using the bee-mutation algorithm.

5. Result and discussion

The proposed model is simulated using the extended version of CloudSim. CloudSim can be implemented on different platforms like Windows and Linux and requires a minimum 8 GB of main memory, Intel Dual-core processor with a frequency of 1.87 GHz or higher. The proposed TheCR-BM-based hybrid framework has been evaluated for its performance by using variousperformance metrics. In this framework, we have considered five fog devices of which performances are provided in Table 1.

5.1. Performance analysis various fog devices

Throughput is one of the significant parameters of ineffective data transmission, particularly in the IoT-fog network. The throughput of the devices has been measured concerning the number of packets that the device transmits within a second. The average throughput among the devices is about 85.6 packets / second. The response time is the measure of time taken by the fog nodes to respond to the received request. The response time is the accumulation of time for node deployment, makespan, and communication time for the request. The response time for the proposed framework is about 78.8 ms. In the IoT-fog system, the ability to process the number of user request without affecting the capacity of fault tolerance is known as scalability. The average scalability of the five devices is 88.6 numbers of user requests processed (URP).

Table 1. Performance of various fog devices.

Parameters	Device 1	Device 2	Device 3	Device 4	Device 5	Average
Throughput (pps)	96	87	81	68	96	85.6
Response Time (ms)	90	78	87	69	70	78.8
Scalability (URP)	85	98	68	99	93	88.6
Performance	92	93	70	89	88	86.4
Availability	99	88	62	77	65	78.2
Usability	73	83	91	63	85	79
Reliability	87	98	78	64	76	80.6
Overhead Associated	68	89	92	63	71	76.6
Cost Effectiveness	89	68	72	67	71	73.4

The overall effectiveness of the proposed system is about 86.4%. For any IoT-fog system, the foremost requirement is the availability of fog nodes to receive the request and process it. It is the ratio of uptime of the node to the total time of the node. In the proposed system, the availability is about 78.2%. The usability is the metric that defines the number of resources consumed to carry out the process effectively whereas the reliability of the system deliberates its efficiency to provide accurate outcomes. The usability and reliability of to provide accurate outcomes. The usability and reliability of the proposed CR-BM-based hybrid framework are 79% and 80.6%, respectively. For any system implemented it is certain to include cost and other overheads. So, the system must be cost-effective with lower overheads. The average overhead associated with the proposed framework is about 76.6% and its corresponding cost-effectiveness is 73.4%. The average performance of all five devices is shown in Fig. 3.

5.2. Comparative analysis

The proposed CR-BM-based hybrid framework has been compared with the fog based on the existing genetic algorithm over the parameters of execution cost and network usage. The execution cost of the proposed framework is about $82020.20 in comparison to the cost of the existing genetic algorithm which is $85042.20 as shown in Fig. 4. The total network usage of the existing GA-based IoT-fog framework is higher which is 635550 Mbps in contrast to 626020 Mbps as obtained by the proposed CR-BM-based hybrid framework. The network usage of both approaches is shown in Fig. 5. The comparison of both existing and proposed algorithms is shown in Table 2.

Figure 3. Performance of CRBM IoT-Fog.

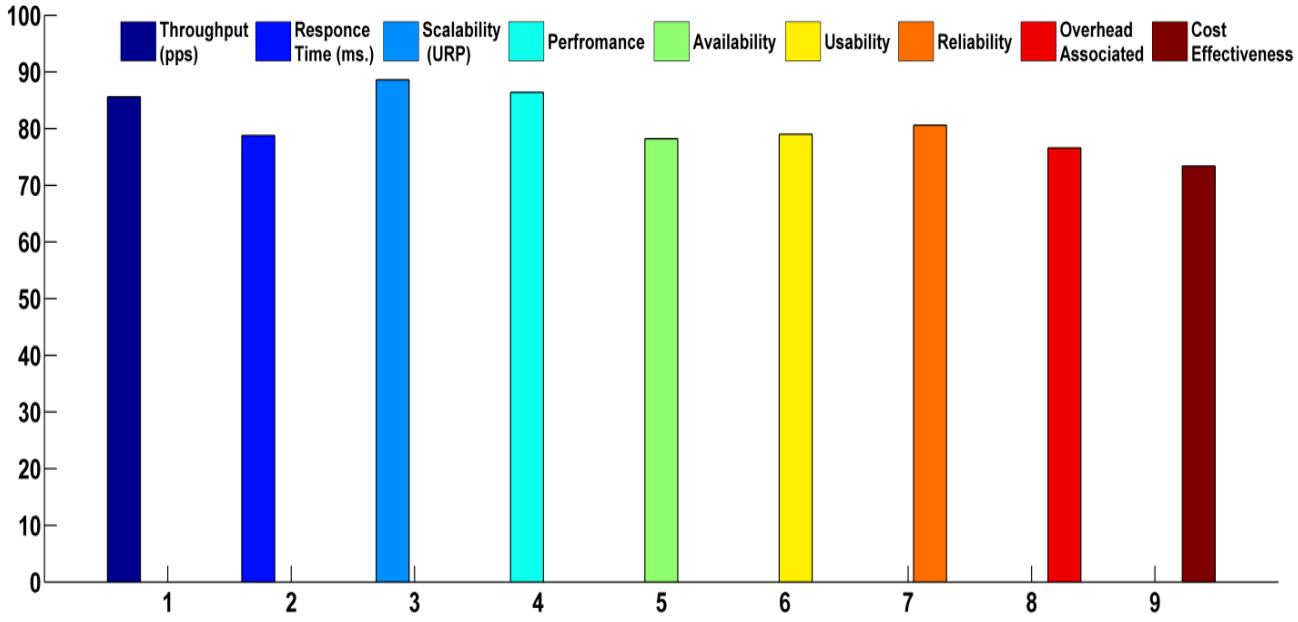

Table 2. Comparison between the proposed and the existing algorithms in IoT-fog.

Parameter	Existing Genetic Algorithm (Skarlat et al., 2017)	Novel Bee Mutation
Execution Cost($)	$85042.20	$82020.20
Network usage(Mbps)	635550.0	626020.0

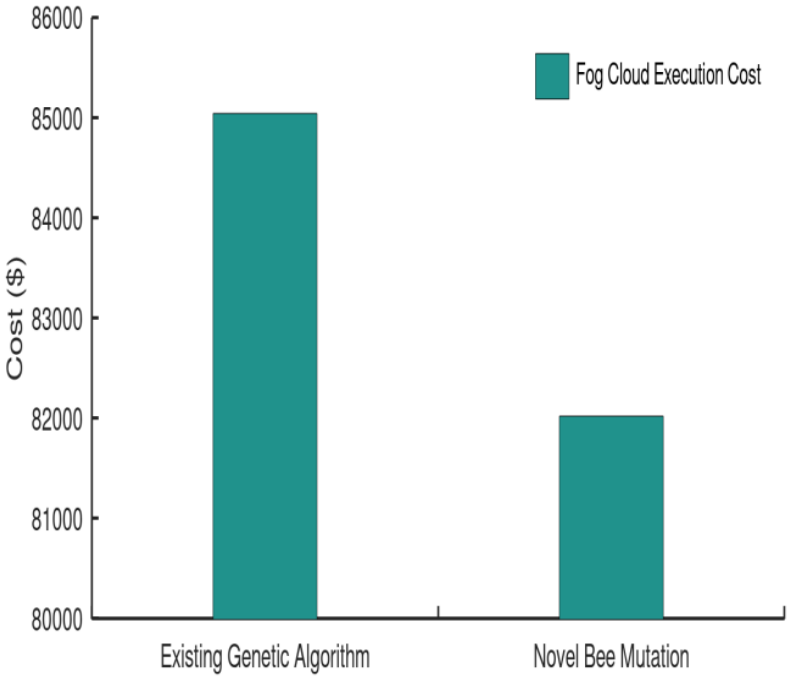

Figure 4. Execution cost for the IoT-fog framework.

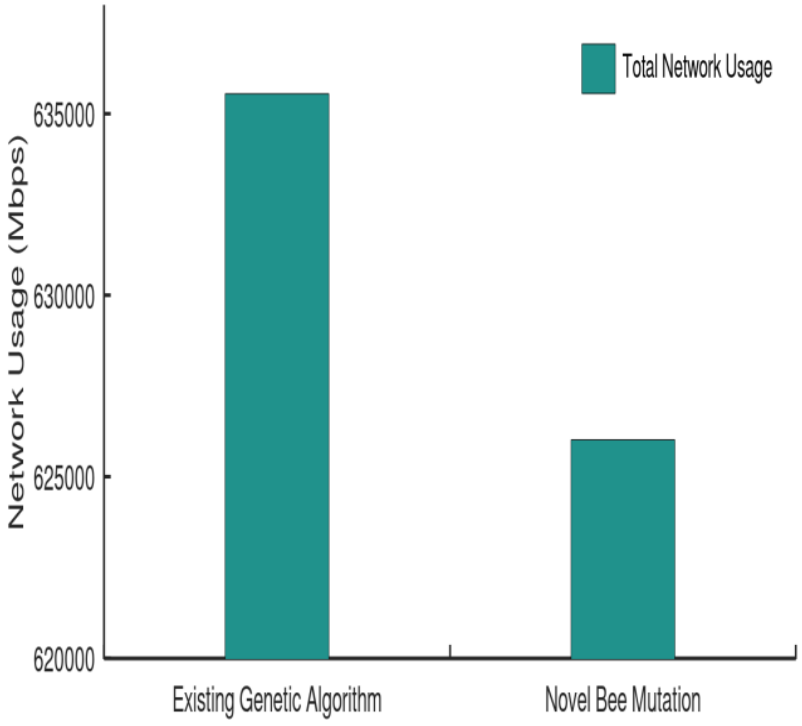

Figure 5. Network use in theIoT-fog framework.

6. Conclusions

A fault-tolerant QoS aware CR-BM-based hybrid framework has been proposed for optimal service placement. The proposed model considers the combination of two renowned fault diagnosing techniques known as checkpoints & replication, and Bee-Mutation techniques. Here, the checkpoint technique begins the process from the established pre-defined checkpoints at the instance of failure, and the replication technique prevents the delay in the checkpoint mechanism. This makes the proposed CR-BM-based hybrid framework fault-tolerant. Another aspect of the proposed model is to place the service effectively among the fog nodes. For optimal service placement, a novel bee-mutation algorithm has been proposed by integrating the artificial bee colony and genetic algorithm. The performance of the proposed CR-BM-based hybrid framework has been evaluated with several different performance metrics like throughput, performance, cost-effectiveness, and several other metrics. We have also provided a detailed comparison of the proposed framework with the existing GA-based fog system over the execution cost ($) and network usage (Mbps). In the obtained results, the proposed framework outperforms the existing model with network usage of 626020 Mbps. and execution cost of $82020.20. Future work will involve the establishment of a pro-active fault diagnosing system for IoT-fog with the federal cloud system to enhance its security and performance.

Conflict of Interest

The authors have no conflict of interest to declare.

References

Amjad, M. K., Butt, S. I., Kousar, R., Ahmad, R., Agha, M. H., Faping, Z., . . . Asgher, U. (2018). Recent research trends in genetic algorithm based flexible job shop scheduling problems. *Mathematical Problems in Engineering, 2018*, 1-32. https://doi.org/10.1155/2018/9270802

Baranwal, G., Yadav, R., & Vidyarthi, D. P. (2020). QoE aware IoT application placement in fog computing using modified-topsis. *Mobile Networks and Applications,25*(5), 1816-1832. https://doi.org/10.1007/s11036-020-01563-x

Brogi, A., & Forti, S. (2017). Qos-aware deployment of iot applications through the fog. *IEEE Internet of Things Journal,4*(5), 1185-1192. https://doi.org/10.1109/JIOT.2017.2701408

Cardellini, V., Grassi, V., Lo Presti, F., & Nardelli, M. (2016). Optimal operator placement for distributed stream processing applications. Proceedings of *the 10th ACM International Conference on Distributed and Event-based Systems*, 69-80. https://doi.org/10.1145/2933267.2933312

Garraghan, P., Ouyang, X., Yang, R., McKee, D., & Xu, J. (2019). Straggler root-cause and impact analysis for Massive-scale virtualized cloud datacenters. *IEEE Transactions on Services Computing,12*(1), 91-104. https://doi.org/10.1109/TSC.2016.2611578

Gupta, P., & Gupta, P. K. (2020). *Trust & Fault in Multi Layered Cloud Computing Architecture*. Springer. https://doi.org/10.1007/978-3-030-37319-1

Iotti, N., Picone, M., Cirani, S., & Ferrari, G. (2017). Improving quality of experience in future wireless access networks through fog computing. *IEEE Internet Computing,21*(2), 26-33. https://doi.org/10.1109/MIC.2017.38

Jiang, J. W., Lan, T., Ha, S., Chen, M., & Chiang, M. (2012). Joint vm placement and routing for data center traffic engineering. *2012 Proceedings IEEE INFOCOM*, 2876-2880. https://doi.org/10.1109/INFCOM.2012.6195719

Lo, N. G., Flaus, J., & Adrot, O. (2019). Review of machine learning approaches in fault diagnosis applied to iot systems. *2019 International Conference on Control, Automation and Diagnosis (ICCAD*), (pp. 1-6).. https://doi.org/10.1109/ICCAD46983.2019.9037949

Mebrek, A., Merghem-Boulahia, L., & Esseghir, M. (2017). Efficient green solution for a balanced energy consumption and delay in THE IoT-Fog-Cloud computing. *2017 IEEE 16th International Symposium on Network Computing and Applications (NCA)*, 1-4. https://doi.org/10.1109/NCA.2017.8171359

Nagaraj, A. (2021). *Introduction to internet-of-things (iot). Introduction to Sensors in IoT and Cloud Computing Applications*, 55-69 https://doi.org/10.2174/9789811479359121010006

Oma, R., Nakamura, S., Duolikun, D., Enokido, T., & Takizawa, M. (2018). Fault-tolerant fog computing models in the iot. Advances on *P2P, Parallel, Grid, Cloud and Internet Computing*, 14-25. https://doi.org/10.1007/978-3-030-02607-3_2

Ozeer, U., Etchevers, X., Letondeur, L., Ottogalli, F., Salaün, G., & Vincent, J. (2018). Resilience of stateful iot applications in a dynamic fog environment. Proceedings of the *15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 332-341.
https://doi.org/10.1145/3286978.3287007

Rajalakshmi, A., Vijayakumar, D., & Srinivasagan, K. G. (2014). An improved dynamic data replica selection and placement in cloud. *2014 International Conference on Recent Trends in Information Technology*, 1-6.
https://doi.org/10.1109/ICRTIT.2014.6996180

Sethi, P., & Sarangi, S. R. (2017). Internet of things: Architectures, protocols, and applications. *Journal of Electrical and Computer Engineering,2017*, 1-25.
https://doi.org/10.1155/2017/9324035

Skarlat, O., Nardelli, M., Schulte, S., Borkowski, M., & Leitner, P. (2017). Optimized iot service placement in the fog. *Service Oriented Computing and Applications,11*(4), 427-443.
https://doi.org/10.1007/s11761-017-0219-8

Wang, K., Shao, Y., Xie, L., Wu, J., & Guo, S. (2020). Adaptive and Fault-Tolerant data processing in HEALTHCARE IoT based on Fog Computing. *IEEE Transactions on Network Science and Engineering,7*(1), 263-273.
https://doi.org/10.1109/TNSE.2018.2859307

Xu, Y., Fan, P., & Yuan, L. (2013). A simple and efficient artificial bee colony algorithm. *Mathematical Problems in Engineering,2013*, 1-9.
https://doi.org/10.1155/2013/526315

Yang, R., Zhang, Y., Garraghan, P., Feng, Y., Ouyang, J., Xu, J., . . . Li, C. (2017). Reliable computing service in MASSIVE-SCALE systems through rapid Low-cost failover. *IEEE Transactions on Services Computing,10*(6), 969-983.
https://doi.org/10.1109/TSC.2016.2544313

Yen, I., Zhang, S., Bastani, F., & Zhang, Y. (2017). A framework for iot-based monitoring and diagnosis of manufacturing systems. *2017 IEEE Symposium on Service-Oriented System Engineering (SOSE),* 1-8.
https://doi.org/10.1109/SOSE.2017.26

Yi, S., Li, C., & Li, Q. (2015, June). A survey of fog computing: concepts, applications, and issues. *In Proceedings of the 2015 workshop on mobile big data* (pp. 37-42).
https://doi.org/10.1145/2757384.2757397