IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 2, April 2026, pp. 1371~1380
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i2.pp1371-1380

Blockchain-enabled framework using diversity mutation with siberian tiger optimization for offloading in fog computing

Srikanta Murthy Rajini[1], Reginald Shilpa[2]
[1]Department of Information Science and Engineering, Vidyavardhaka College of Engineering, Mysore, India
[2]Department of Electronics and Communication Engineering, Vidyavardhaka College of Engineering, Mysore, India

Article Info

Article history:

Received May 22, 2025
Revised Jan 19, 2026
Accepted Feb 6, 2026

Keywords:

Computation offloading
Diversity mutation
Fog computing
Internet of things
Quality of service
Siberian tiger optimization

ABSTRACT

Fog computing has developed as a promising framework to support latency-sensitive internet of things (IoT) applications for mobile devices operating in dynamic environments. During the offloading process, malicious activities interrupt the existing methods, which increases the execution time. Therefore, this research proposes a diversity mutation with siberian tiger optimization (DM-STO) for computation offloading in blockchain based fog computing. The blockchain is used to secure offload and attain quality of service (QoS) mobile users with less energy consumption and execution time. The DM-STO can balance workloads among local devices and fog servers. The diversity mutation operation improves the exploration ability to dynamic network conditions, leading to efficient computational offloading in fog computing. The execution time, service cost and energy consumption are evaluated to calculate the performance of the proposed DM-STO with varying numbers of IoT requests such as 50, 100, 200, and 300. For 50 IoT requests with a fixed fog server of 10, the DM-STO achieves an execution time of 18 s, a service cost of 10$ and energy consumption of 5 mJ compared to the BAT algorithm.

Corresponding Author:

Srikanta Murthy Rajini
Department of Information Science and Engineering, Vidyavardhaka College of Engineering
P B No.206, Gokulam III stage, Mysore, India
Email: rajinis@vvce.ac.in

1. INTRODUCTION

Computation offloading is considered an optimal solution for resource-constrained mobile devices sustaining computationally intensive mobile collaboration applications in 5G networks [1]. An increasing number of people are using mobile devices to store data on the internet due to the quick development of information technology [2]. Moreover, conventional architectures are unable to handle huge amounts of which resulting from data explosion. This issue is resolved temporarily based on local storage and processing sources. However, if mobile devices need to upload the data to a distant cloud, users experience significant data delays [3]. With the evolution of internet of things (IoT), production line components such as sensors and actuators are transformed into a cyber-physical manufacturing system connected to the cloud via the internet [4]. The recent development in fog computing and its variations as outsourcing techniques have created an exciting research area [5]. Fog computing takes data processing closer to the source by extending cloud capacities to the network edge, reducing bandwidth usage and latency thereby enhancing overall system efficiency [6]. In fog computing, computation is offloaded to edge devices and these characteristics such as low latency enables to processing of data independently without the concern of a cloud server [7].

Therefore, security limitations and challenges in fog computing contain privacy, security and trust because of a decentralized untrusted environment [8]. This fog computing depends on a centralized cloud for

authentication. Blockchain technology is used to reduce the problem when saving network and computation costs while achieving traceability [9]. The individual characteristics of blockchain such as secure encryption, mutual authentication among nodes, transparency and reliability, are used for encrypted distributed records and databases in which the operations are performed to fog-based authentication system [10], [11]. The computation offloading is intended as a promising approach to fog devices for satisfying the quality of service (QoS) of IoT through less energy consumption [12]. In the offloading procedure, the mobile device offloads its resource tasks to the fog environment to decrease the computation cost and overhead when compared with local execution [13].

The major issues arise from the fluctuations of devices and the differences in the load-sharing requirements in fog computing environments. Resource management activities including load balancing to facilities become a challenging issue in certain circumstances [14]. The fog computing infrastructure is sustained in smart cities with the high processing requirement and the relatively low and steady latency that is typical of IoT applications such as traffic monitoring, gaming, augmented reality, traffic control and management, environmental sensing, and public surveillance [15]. These applications at the sensor layer produce enormous volumes of data such as video frames that must be processed by a fog node which involves computations. Also, these applications require an instantaneous response to quickly react to the varying circumstances [16]. The extra overhead of data scheduling, processing, management and organization on fog nodes causes huge time and expense specifically to maintain processed info in a distributed network [17]. The complexity increases when the fog nodes are implemented with different IoT devices for computed outsourcing [18]. Furthermore, problems arise with the protection of data transmission during information exchange and existing stored information in terms of data security and privacy [19].

The main contributions of this research are:

i) This research presents a blockchain-based fog computing framework to enable secure computation offloading for IoT applications. The blockchain ensures data integrity, authentication and traceability of offloading transactions while maintaining QoS requirements thereby maintaining latency-sensitive and resource constrained fog environments.
ii) Diversity mutation with siberian tiger optimization (DM-STO) algorithm is proposed to optimize task offloading decisions among mobile devices and fog servers. The algorithm effectively balances workloads through considering dynamic network conditions and heterogeneous fog resources.
iii) The integration of diversity mutation enhances the exploration capability of the siberian tiger optimization (STO) algorithm and mitigates premature convergence. As a result, the DM-STO achieves less execution time and energy consumption, the simulation results validate the effectiveness of DM-STO under different IoT workloads.

This research is organized as follows. Section 2 analyzes the related work. Section 3 explains the proposed method for computational offloading. Section 4 provides results and discussion, Lastly, section 5 provides the conclusion of this research paper.

2. RELATED WORKS

Recently, an extensive number of studies, including numerous frameworks and models have been developed on computational offloading in fog environments which are analyzed in this section. Alam *et al*. [20] suggested a deep reinforcement learning (DRL) approach for computation offloading in blockchain-based systems. In integration with DRL, the blockchain enhances mobile communication efficiency. The DRL in blockchain operation enhances IoT by simultaneously securing transactions and supporting community-based divisibility. The decentralized and effective communication is integrated into DRL and blockchain within wireless services which enables for reliable and scalable resource allocation. But it requires higher computational resources due to its complex architecture which leads to inefficient offloading and affects overall performance. Thangaraj and Sree [21] introduced a mobility-aware secure computation offloading (MSCO) in blockchain-enabled fog computing. The MSCO offered decentralized and secure offloading service for end-users which enables cost-effective offloading to fog servers. The hybrid of genetic algorithm (GA) and particle swarm optimization (PSO) was utilized for offloading process. However, MSCO was interrupted by malicious activities during the offloading process that affected processing time.

Aknan *et al.* [22] developed an artificial intelligence (AI) and blockchain-assisted framework for offloading in fog computing. The BAT algorithm was developed for the offloading process which has a high convergence rate and the ability to consider offloading decisions in run time which enhances the result quality. The blockchain technique secures IoT applications and their data from attacks. However, the BAT algorithm suffered from premature convergence and leading suboptimal solutions which affects the efficiency of computational offloading. Samy *et al*. [23] implemented a secure task offloading in blockchain based DRL. Initially, blockchain was developed to obtain data integrity, confidentiality and security of

offloading in mobile devices. Then, the task of offloading multiple users with mobile devices is optimized for time and energy costs. The DRL was applied to efficiently derive near-optimal task offloading decisions. Nevertheless, DRL suffered from scalability issues which led to suboptimal performance and reduced efficiency in the offloading process.

Sarkar and Kumar [24] presented an energy-efficient computational offloading in heterogeneous fog computing. The total delay and energy consumption of data was initially calculated through a heterogeneous system which formulates the mixed-integer problem to optimize bandwidth allocation and offloading decision. To make offloading decisions, multiple deep neural network (DNN) is applied which use a binary offloading strategy. However, security and high energy consumption are major limitations that make it insufficient for real-time applications. Lin *et al.* [25] suggested an energy-efficient joint resource allocation and computation offloading using the successive convex approximation (SCA)-based interior-point technique in fog computing. The energy-efficient non-orthogonal multiple access (NOMA)-enabled computing offloading was developed which integrated task vehicles, fog access points (F-APs), idle vehicles and auxiliary F-Aps to handle the tasks. Moreover, interior-point technique according to SCA was developed to achieve a sub-optimum solution of task separation and bandwidth allocation. The SCA-based interior-point method improves convergence speed through effectively solving non-convex which makes it suitable for managing complex resource allocation. However, it suffered from local optima convergence because of primary feasible points which limits the performance in high dynamic fog computing.

Kök and Özdemir [26] presented a deep reinforcement learning offloading scheme (DRLOS) for computation offloading in fog computing. The DRL was developed which jointly considered the content type and status of the fog server. Then, a novel virtual layer named FogOrch orchestrates which manages and performs the requirements of fog layer resources using a DRL agent. The DRLOS optimizes the computational offloading in fog computing through learning dynamically and adapting complex environments which leads to effective resource allocation and less latency. However, it suffered from higher computational overhead and long convergence because of complex training in dynamic environments. Dang and Kim [27] developed a distributed computation offloading (DISCO) for fog computing. The distributed and scalable framework with less computational complexity was unachievable through global optimization with centralized data management in fog networks. The DISCO was applied for offloading of divided tasks using matching theory. The DISCO effectively balances computational load among cloud resources to reduce latency and energy consumption. The DISCO increased latency because of overhead of handling distributed resources among numerous fog nodes.

Li *et al*. [28] implemented a subtask partition and resource allocation-based intelligent computation offloading (SPRA-ICO) for user satisfaction in fog computing. The actor-critic network was combined with noise to produce constant result achievement which ensuring controlled chance in deterministic policy exploration. The SPRA-ICO enhances computing through optimizing resource allocation and maintaining less latency thereby securing data and reducing computational complexity. However, it leads to increased overhead in latency and communication due to the complexity of tasks and handling resource allocation among distributed nodes which affects the performance. Liu *et al.* [29] suggested a GA for efficient delay computation offloading in fog computing. The distributed multi-hop computing using GA was developed where tasks are offloaded recursively between network computing points (NCPs). The GA minimizes an optimal space through creating filter criteria that screen nodes before the initialization stage thereby enhancing population initialization. It employs a crossover operator to improve convergence and avoid risk of resource overconsumption because of spherical scheduling. The GA algorithm effectively explores search space due to its ability and frequency tuning thereby leading to optimal resource allocation. However, it suffered from premature convergence that led to suboptimal solutions that do not effectively explore the search space. Table 1 presents the summary table of existing research.

3. METHOD

The secure computation offloading framework in a fog computing environment is detailed in this section. The main aim of this framework is to select an optimal authorized fog server with blockchain to provide QoS constraints in IoT with less energy consumption and time. The proposed framework is shown in Figure 1 which contains IoT, fog, and cloud layer. All three layers are interconnected with the wireless medium. Every layer is explained as follows. In IoT layer, the processing time depends on the performance of the user's mobile devices. Mobile devices are resource-constrained devices which transfer tasks into cloud or fog layer once exceed the process of computing ability. Every mobile device in this layer has a blockchain which enables it to integrate network and offload into fog layer. The fog layer comprises geographically dispersed fog devices such as gateways, micro-data center, routers, and road side unit (RSU) which are used to manage tasks from mobile devices. Fog devices have limited computing abilities, so tasks requiring significant computational power which are offloaded to cloud layer. The cloud layer has extensive stages that progress and store a vast number of data.

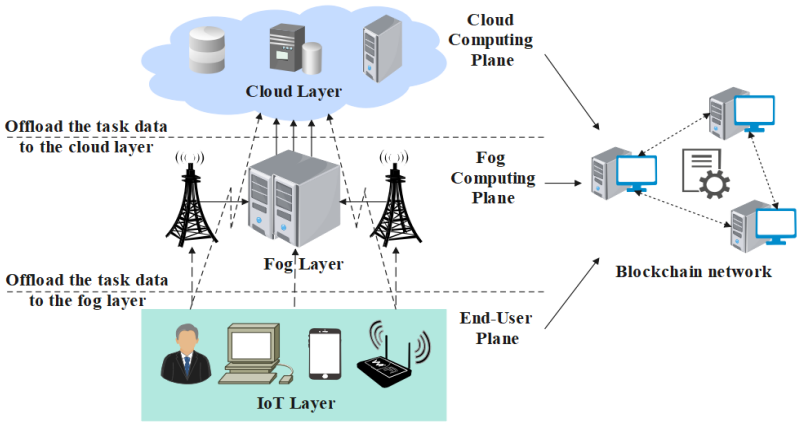

Figure 1. Secure offloading framework in blockchain

3.1. Block generation for offloading

Generally, blockchain is a distributed data in which every data block contains transaction information which is applied for data validation and generates upcoming blocks. The blockchain is applied to monitor unloaded transactions and assure data security. Every computation task is offloaded into the fog server which is recorded as a block transaction and included after proof-of-work (PoW) based unified verification. Moreover, the PoW integrates traceability and verifiability by generating a hash function. Once a transaction is created, it is classified as an unverified transaction for every server. The node in the blockchain resolves PoW to obtain the power to produce blocks and transmit every time-stamp transaction verified in the block to the entire network. The new block is included in the present blockchain once obtaining consent and verification of other nodes. Every block has earlier produced a block of hash values. If the block is altered through an attacker, every earlier produced block is required to be modified.

The resource monitoring of the fog server is done through records in the blockchain which is updated dynamically. The related fog server allocated the task once the request based on the fog server values and its service waiting time. A record is simultaneously generated and reorganized consequently. Through the blockchain technique, the whole transaction history is stored in mobile devices and fog servers. The fog server uses its private key to signature the real updated geographical positions, updated information and workloads. Moreover, mobile devices utilize its private key to sign the offloading transactions. Every mobile device stores a whole transaction history, which is defined easily and the fog server is designated to offload its certain computation through the proposed STO algorithm.

3.2. Computation offloading model

The STO algorithm is used in this research for computation offloading in blockchain based fog computation. The process of position updating for siberian tigers in STO is exhibited in dual various stages based on natural behavior of animals. The STO generates possible solutions to problems from population searching based on iteration [30]. Its population contains Siberian tigers that search for better solutions by altering positions in search space and every siberian tiger is an STO population member. The position of search space denotes the problem variable score. The initial location in search space is determined randomly by using (1). Here, $x_{i,j}$ is jth dimension of x_i in search space, $r_{i,j}$ is a random number in [0, 1], ub_j and lb_j are an upper and lower bound of jth problem variable, N and m are the number of STO members and problem variables respectively.

$$X_{i,j} = lb_j + r_{i,j} \cdot \left(ub_j - lb_j\right),\ i = 1,2,\dots,N; j = 1,2,\dots,m \quad (1)$$

3.2.1. Prey hunting

In this phase, the STO members are updated according to a simulation of the hunting strategy. After prey selection, it attacks and kills prey in a racing procedure. Hence, prey hunting stage is stimulated in dual phases. Initially, population member positions are updated according to selection and attack on prey. This leads to extensive and rapid variations in STO member location which results in increased global search ability and algorithm exploration of search space. In STO, prey location for every Siberian tiger is designated from other population members which have better objective value. The set of positions

of prey is exposed in (2). Here, X_{best} is a best-candidate solution. Then, member from this PP_i is selected randomly as an attacked target through ith siberian tiger and its position is estimated according to attack simulation on prey as exposed in (3). Here, $TP_{i,j}$ is a jth dimension of TP_i, $x_{i,j}^{p1S1}$ is a jth dimension of new position, $I_{i,j}$ is a random number in the interval of $\{1, 2\}$. In STO, a member updating a new position which is suitable if it enhances the values as (4).

$$PP_i = \{k \in \{1,2, \dots, N\} \wedge F_k < F_i\} \cup \{X_{best}\} \quad (2)$$

$$x_{i,j}^{p1S1} = x_{i,j} + r_{i,j} \cdot \left(TP_{i,j} - I_{i,j} \cdot x_{i,j}\right),\ i = 1,2, \dots, N; j = 1,2, \dots, m \quad (3)$$

$$X_i = \{X_i^{p1S1},\ F_i^{p1S1} < F_i;\ X_i,\ else, \quad (4)$$

Where $x_{i,j}^{p1S1}$ is a new location of ith member according to initial phase of STO; and F_i^{p1S1} is an objective score of ith member. In the second stage, the population member position is updated according to a chase procedure. In this phase, tiger alters its location in area and attacks prey. This procedure enhanced algorithm capability in local search and obtaining the best solutions. To simulate chasing procedure, a new location near the attack site is estimated using (5). Based on (6), if the objective function value improves, it newly estimates the position and replaces the previous position of its respective member. Here, X_i^{p1S2} is a new location of ith tiger according to second stage of initial phase, $x_{i,j}^{p1S2}$ is jth dimension of search space, F_i^{p1S2} is objective function, and T is a number of iterations.

$$x_{i,j}^{p1S2} = x_{i,j} + \frac{r_{i,j} \cdot (ub_j - lb_j)}{t},\ i = 1,2, \dots, N; j = 1,2, \dots, m; t = 1,2, \dots, T \quad (5)$$

$$X_i = \{X_i^{p1S2},\ F_i^{p1S2} < F_i;\ X_i,\ else, \quad (6)$$

3.2.2. Fighting with a bear

Observation of normal siberian tigers indicates that these animals fight through black and brown bears because they clash over prey and fight to protect their survival. In this stage, STO population members are updated by stimulating the siberian tiger approach once it fights with a bear. During a fight, tiger initially traps and attacks bear, then engages in combat with bear on battlefield until it kills it. Hence, siberian tigers' fighting approach with bears is stimulated in dual stages such as attack and fight. In attack stage, attack of ith tiger on bear is modelled and remaining population members are taken as bears set. From this possible bear set, the attacked bear position is selected randomly. This leads to rapid and significant alterations in STO member location which enhances global search exploration ability. Hence, the new position is calculated initially to simulate the above concept for ith STO member $i = 1, 2, \dots, N$ as (7). Where, $x_{k,j}$ is jth dimension of bear position, $j = 1, 2, \dots, m$, k is a chosen bear position from set $\{1, 2, \dots, i-1, i+1, \dots, N\}$, X_i^{p2S1} is a new location of ith member of STO, $x_{i,j}^{p2S1}$ is its jth dimension. If the objective function value is enhanced based on (8), the newly estimated location substitutes before one of its respective members.

$$x_{i,j}^{p2S1} = \{x_{i,j} + r_{i,j} \cdot \left(x_{k,j} - I_{i,j} \cdot x_{i,j}\right),\ F_k < F_i;\ \ x_{i,j} + r_{i,j} \cdot \left(x_{i,j} - I_{i,j} \cdot x_{k,j}\right), else \quad (7)$$

$$X_i = \{X_i^{p2S1},\ F_i^{p2S1} < F_i;\ X_i,\ else, \quad (8)$$

Here, F_k is an objective score of bears, F_i^{p2S1} is an objective value of X_i^{p2S1}. In second phase, population member location is updated according to the simulation of fight a struggle. It causes small adjustments in population member position which enhances local search of STO and enhances its exploitation capability. Based on this behavior, initially, a random location adjacent to fight place is estimated by (9). Then, new location is considered for the update procedure which improves the objective function score based on (10).

$$x_{i,j}^{p2S2} = x_{i,j} + \frac{r_{i,j}}{t}(ub_j - lb_j),\quad i = 1,2, \dots, N; j = 1,2, \dots, m; t = 1,2, \dots, T \quad (9)$$

$$X_i = \{X_i^{p2S2},\ F_i^{p2S2} < F_i;\ X_i,\ else, \quad (10)$$

Here, X_i^{p2S2} is a new location of ith tiger of STO, $x_{i,j}^{p2S2}$ is its jth dimension, F_i^{p2S2} is objective score of X_i^{p2S2}. The STO initial iteration is finished after updating every tiger according to the initial and second

stages. Then, it enters to following iteration with attained new scores and its positions are updated until the last iteration. The optimal candidate solutions generated at each iteration are stored in the results as the final solution to the problem. Like other population-based optimization algorithms in later iterations of STO, all the siberian tiger moves near individual optimal regions that result in a population diversity reduction. To minimize the premature convergence probability for STO, a diversity mutation operation is accomplished on the present optimal Siberian tiger individual. Let's assume that, individual $X_i = (x_{i1}, x_{i2}, \dots, x_{id})$ of siberian tiger selects an element $x_k = (k = 1, 2, .., d)$ from individual X_i with $1/d$ probability and produces real number randomly in the range of $[l_i, u_i]$ rather than an element x_{ik} from X_i which generates new individual $X_i' = (x_{i1}', x_{i2}', \dots, x_{id}')$. The diversity mutation operation is given in (11).

$$X_i' = \{l_i + \lambda \cdot (u_i - l_i) \quad i = k\ X_i \ \ otherwise \tag{11}$$

Where, l_i and u_i are upper and lower bounds, $\lambda \in [0, 1]$ is a random number. The DM-STO prevents premature convergence and improves the model's capability to escape local optima thereby leading to optimal solution. The DM-STO algorithm is used in this research for the offloading process due to its ability to effectively balance workloads among fog nodes and local devices. The pseudocode of DM-STO is given as Algorithm 1.

Algorithm 1. Diversity mutation with siberian tiger optimization
start DM-STO
Input: the problem data (objective function, variables, and constraints)
Establish STO size of population (N),
Create the random initial population matrix by (1)
for $t = 1$ to T
for $i = 1$ to N
Stage 1: prey hunting
Update the set of prey for ith STO member by applying (2)
Compute STO member's ith new location depending on 1st stage by using (3)
Update ith STO member by employing (4)
Compute STO member's ith new location depending on 2nd stage by using (5)
Update ith STO member by applying (6)
Stage 2: fighting with bear
Randomly choose one population member as a bear location
Compute STO member's ith new location depending on 1st stage by using (7)
Update ith STO member by (8)
Compute STO member's ith new location depending on 2nd stage by using (9)
Updated ith STO member by applying (10)
end
Perform diversity mutation on current siberian tiger individual using (11)
Save the determined optimal best solution
end
Output: the optimal solution acquired by DM-STO
end DM-STO

4. RESULTS AND DISCUSSION

The performance of the DM-STO algorithm is simulated in Python with a system requirement of 8 GB RAM, i5 processor and Windows 10 OS. The execution time, service cost and energy consumption are considered for calculating the performance of the proposed DM-STO with no. of IoT requests such as 50, 100, 200, and 300. Table 2 shows the simulation parameters for this research. Tables 3 to 5 present the execution time, service cost, and energy consumption results with a fixed fog server of 10.

Table 3 presents the execution time of DM-STO with various no. of IoT requests such as 50, 100, 200, and 300 for a fixed fog server of 10. The grey wolf optimization (GWO), spotted hyena optimization (SHO), and STO are considered to compare the DM-STO performance. The STO enhanced exploration and exploitation balance by diversity mutation operation. The DM-STO quickens the convergence by preserving population diversity and reducing the number of iterations to find optimal solutions thereby reducing execution time. The DM-STO obtains less execution time of 18 s, 97 s, 516 s, and 965 s for 50, 100, 200, and 300 no. of IoT requests respectively.

Table 2. Simulation parameters

Parameters	Values
Fog nodes	5-50
Bandwidth	10-100 Mbps
Number of iterations	100-300
Cloud datacenter	1

Table 3. Execution time (s) for proposed DM-STO with fixed fog server of 10

No. of IoT request	GWO	SHO	STO	DM-STO
50	71	45	26	18
100	160	138	115	97
200	585	555	530	516
300	1026	995	980	965

Table 4 presents the service cost of DM-STO with various no. of IoT requests such as 50, 100, 200, and 300 for fixed fog server of 10. The GWO, SHO, and STO are considered to compare the DM-STO performance. The STO enhanced the exploration ability through its diversity mutation operation compared to GWO, SHO, and traditional STO. It prevents early convergence and enhances the model's capability to escape local optima thereby leading to better solutions and less service cost. The DM-STO obtains less service cost of 10$, 113$, 525$, and 930$ for 50, 100, 200, and 300 no. of IoT requests respectively.

Table 5 presents the execution time of DM-STO with various no. of IoT requests such as 50, 100, 200, and 300 for a fixed fog server of 10. The GWO, SHO, and STO are considered to compare the DM-STO performance. The DM-STO enhances the search diversity, enables the algorithm to avoid local minima issues and leading better convergence to optimal solutions thereby reducing energy consumption. The DM-STO obtains less energy consumption of 5 mJ, 27 mJ, 110 mJ, and 295 mJ for 50, 100, 200, and 300 no. of IoT requests respectively.

Table 4. Service cost ($) for proposed DM-STO with fixed fog server of 10

No. of IoT request	GWO	SHO	STO	DM-STO
50	80	55	25	10
100	195	170	135	113
200	610	585	550	525
300	1045	990	975	930

Table 5. Energy consumption (mJ) for proposed DM-STO with fixed fog server of 10

No. of IoT request	GWO	SHO	STO	DM-STO
50	110	65	20	5
100	135	90	45	27
200	265	220	165	110
300	390	365	320	295

4.1. Comparative analysis

The comparison of proposed DM-STO with existing BAT [22] is given in this section for the fixed fog server of 10. The metrics such as execution time, service cost and energy consumption are considered for calculating the performance of the proposed DM-STO with no. of IoT requests such as 50, 100, 200, and 300. The DM-STO achieves 18 s, 10$, and 5 mJ of execution time, service cost and energy consumption for 50 IoT requests in the fixed fog server of 10. Table 6 shows the comparative analysis.

Table 6. Comparative analysis with fixed fog server of 10

Method	Metrics	No. of IoT request			
		50	100	200	300
BAT [22]	Execution time	21	116	543	1018
	Service cost	16	128	560	975
	Energy consumption	8	35	125	320
DM-STO	Execution time (s)	18	97	516	965
	Service cost ($)	10	113	525	930
	Energy consumption (mJ)	5	27	110	295

4.2. Discussion

The results are taken for three metrics such as execution time, service cost and energy consumption with a fixed fog server of 10. The BAT [22] algorithm suffered from premature convergence and leading suboptimal solutions which affects the efficiency of computational offloading. The DM-STO achieves better when compared to the BAT algorithm in task offloading by integrating various search mechanisms. In DM-STO, diversity mutation improves exploration capabilities which allows it to escape local optima that is significant for task offloading. The BAT algorithm [22] is premature early which limits its ability to find optimal solutions in dynamic fog computing. With the help of its mutation strategies, it dynamically adjusts the search range and enhances the solution space exploration effectively. The presence of diversity mutation in STO enhances the ability to prevent inactivity in local optima whereas BAT algorithm suffered from parameter dependencies and local search tendencies. The DM-STO dynamically adapts non-linearities in fog environments such as processing power and fluctuating bandwidth which ensures effective offloading decisions. Moreover, it reduces energy consumption by adapting to the resource-constrained and heterogeneous nature of fog computing. The diversity mutation mechanism enhances the exploration ability to dynamic network conditions thereby leading to effective computational offloading. It is used for the offloading process because of its ability to balance workloads among local devices and fog nodes. The DM-STO achieves 97 s, 113 $, and 27 mJ of execution time, service cost and energy consumption for 100 IoT requests with a fixed fog server of 10.

5. CONCLUSION

The DM-STO algorithm is proposed in this research for computation offloading in blockchain-based fog computing. The main goal of this research is to select optimal authorized fog sever with blockchain to ensure QoS constraints with less energy consumption and cost. The blockchain is used for securing computational offloading and attaining optimal QoS of mobile users with less execution time and energy consumption. The diversity mutation mechanism enhances exploration ability and mitigates premature convergence thereby enabling reliable offloading decisions among heterogeneous fog nodes. The DM-STO framework is suitable for latency-sensitive and resource constrained applications such as smart cities, healthcare monitoring, industrial IoT and intelligent transportation systems where secure task offloading is significant. The achieved reductions in execution time and energy consumption demonstrates its potential for deployment in real-time fog computing infrastructures. The experimental evaluation was conducted under simulated environments with fixed fog server configurations and real-time network uncertainties such as mobility patterns, large-scale deployments and node failures are not fully considered. The blockchain consensus overhead was not extensively analyzed under higher transaction loads. Future work will focus on extending the framework to large-scale dynamic fog networks, integrating mobility-aware and adaptive blockchain consensus mechanism for offloading decisions.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Srikanta Murthy Rajini	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Reginald Shilpa		✓				✓		✓	✓	✓	✓	✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] K. Zhang, X. Gui, D. Ren, T. Du, and X. He, "Optimal pricing-based computation offloading and resource allocation for blockchain-enabled beyond 5G networks," *Computer Networks*, vol. 203, 2022, doi: 10.1016/j.comnet.2021.108674.

[2] T. Hewa, A. Braeken, M. Liyanage, and M. Ylianttila, "Fog computing and blockchain-based security service architecture for 5G industrial IoT-enabled cloud manufacturing," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 10, pp. 7174–7185, 2022, doi: 10.1109/TII.2022.3140792.

[3] A. Heidari, M. A. J. Jamali, N. J. Navimipour, and S. Akbarpour, "Deep Q-learning technique for offloading offline/online computation in blockchain-enabled green IoT-edge scenarios," *Applied Sciences*, vol. 12, no. 16, 2022, doi: 10.3390/app12168232.

[4] S. Rani, D. Gupta, N. Herencsar, and G. Srivastava, "Blockchain-enabled cooperative computing strategy for resource sharing in fog networks," *Internet of Things*, vol. 21, 2023, doi: 10.1016/j.iot.2022.100672.

[5] J. Shi, J. Du, Y. Shen, J. Wang, J. Yuan, and Z. Han, "DRL-based V2V computation offloading for blockchain-enabled vehicular networks," *IEEE Transactions on Mobile Computing*, vol. 22, no. 7, pp. 3882–3897, 2023, doi: 10.1109/TMC.2022.3153346.

[6] G. Senthilkumar, K. N. Madhusudhan, Y. Jeyasheela, and P. Ajitha, "A novel blockchain enabled resource allocation and task offloading strategy in cloud computing environment," *Automatika*, vol. 65, no. 3, pp. 973–982, 2024, doi: 10.1080/00051144.2024.2314906.

[7] J. Du, W. Cheng, and S. Li, "Joint task offloading and resource allocation in mixed edge/cloud computing and blockchain empowered device-free sensing systems," *Computer Communications*, vol. 209, pp. 38–46, 2023, doi: 10.1016/j.comcom.2023.06.015.

[8] S. Tian, Y. Zhang, Y. Bi, and T. Yuan, "Blockchain-based 6G task offloading and cooperative computing resource allocation study," *Journal of Cloud Computing*, vol. 13, no. 1, 2024, doi: 10.1186/s13677-024-00655-3.

[9] H. Xiaoge, Y. Hongbo, C. Bin, W. Yongsheng, C. Qianbin, and Z. Jie, "Joint optimization of energy consumption and network latency in blockchain-enabled fog computing networks," *China Communications*, vol. 21, no. 4, pp. 104–119, 2024, doi: 10.23919/JCC.fa.2023-0488.202404.

[10] Z. A. Khan and I. A. Aziz, "Dynamic OBL-driven whale optimization algorithm for independent tasks offloading in fog computing," *High-Confidence Computing*, vol. 5, no. 4, 2025, doi: 10.1016/j.hcc.2025.100317.

[11] O. Umoren, R. Singh, Z. Pervez, and K. Dahal, "Securing fog computing with a decentralised user authentication approach based on blockchain," *Sensors*, vol. 22, no. 10, 2022, doi: 10.3390/s22103956.

[12] J. A. Alzubi, O. A. Alzubi, A. Singh, and T. Mahmod Alzubi, "A blockchain-enabled security management framework for mobile edge computing," *International Journal of Network Management*, vol. 33, no. 5, 2023, doi: 10.1002/nem.2240.

[13] K. Moghaddasi, S. Rajabi, and F. S. Gharehchopogh, "Multi-objective secure task offloading strategy for blockchain-enabled IoV-MEC systems: a double deep Q-network approach," *IEEE Access*, vol. 12, pp. 3437–3463, 2024, doi: 10.1109/ACCESS.2023.3348513.

[14] A. M. Rahmani, J. Tanveer, F. S. Gharehchopogh, S. Rajabi, and M. Hosseinzadeh, "A novel offloading strategy for multi-user optimization in blockchain-enabled mobile edge computing networks for improved internet of things performance," *Computers and Electrical Engineering*, vol. 119, 2024, doi: 10.1016/j.compeleceng.2024.109514.

[15] A. A. Khan *et al.*, "Blockchain-enabled infrastructural security solution for serverless consortium fog and edge computing," *PeerJ Computer Science*, vol. 10, 2024, doi: 10.7717/peerj-cs.1933.

[16] N. Premkumar and R. Santhosh, "Secure load balancing in fog computing using improved tasmanian devil optimization algorithm with blockchain," *Wireless Personal Communications*, vol. 136, no. 1, pp. 547–565, 2024, doi: 10.1007/s11277-024-11321-x.

[17] B. Lin, X. Chen, X. Chen, Y. Ma, and N. N. Xiong, "SGCS: an intelligent stackelberg-game-based computation offloading and resource pricing scheme in blockchain-enabled MEC for IIoT," *IEEE Internet of Things Journal*, vol. 11, no. 16, pp. 26727–26740, 2024, doi: 10.1109/JIOT.2024.3360152.

[18] S. Fugkeaw, L. Wirz, and L. Hak, "Secure and lightweight blockchain-enabled access control for fog-assisted IoT cloud based electronic medical records sharing," *IEEE Access*, vol. 11, pp. 62998–63012, 2023, doi: 10.1109/ACCESS.2023.3288332.

[19] O. Umoren, R. Singh, S. Awan, Z. Pervez, and K. Dahal, "Blockchain-based secure authentication with improved performance for fog computing," *Sensors*, vol. 22, no. 22, 2022, doi: 10.3390/s22228969.

[20] T. Alam, A. Ullah, and M. Benaida, "Deep reinforcement learning approach for computation offloading in blockchain-enabled communications systems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 8, pp. 9959–9972, 2023, doi: 10.1007/s12652-021-03663-2.

[21] V. Thangaraj and T. R. Sree, "MSCO: mobility-aware secure computation offloading in blockchain-enabled fog computing environments," *Journal of Cloud Computing*, vol. 13, no. 1, 2024, doi: 10.1186/s13677-024-00599-8.

[22] M. Aknan, M. P. Singh, and R. Arya, "AI and blockchain assisted framework for offloading and resource allocation in fog computing," *Journal of Grid Computing*, vol. 21, no. 4, 2023, doi: 10.1007/s10723-023-09694-7.

[23] A. Samy, I. A. Elgendy, H. Yu, W. Zhang, and H. Zhang, "Secure task offloading in blockchain-enabled mobile edge computing with deep reinforcement learning," *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 4872–4887, 2022, doi: 10.1109/TNSM.2022.3190493.

[24] I. Sarkar and S. Kumar, "Deep learning-based energy-efficient computational offloading strategy in heterogeneous fog computing networks," *The Journal of Supercomputing*, vol. 78, no. 13, pp. 15089–15106, 2022, doi: 10.1007/s11227-022-04461-z.

[25] Z. Lin, Y. Lin, J. Yang, and Q. Zhang, "Energy-efficient joint resource allocation and computation offloading in NOMA-enabled vehicular fog computing," *Mobile Networks and Applications*, vol. 29, no. 5, pp. 1564–1576, 2024, doi: 10.1007/s11036-023-02265-w.

[26] İ. Kök and S. Özdemir, "Content-centric data and computation offloading in AI-supported fog networks for next generation IoT," *Pervasive and Mobile Computing*, vol. 85, 2022, doi: 10.1016/j.pmcj.2022.101654.

[27] H. T. -Dang and D.-S. Kim, "DISCO: distributed computation offloading framework for fog computing networks," *Journal of Communications and Networks*, vol. 25, no. 1, pp. 121–131, 2023, doi: 10.23919/JCN.2022.000058.

[28] Q. Li, B. Tang, J. Li, and S. Chen, "User satisfaction-based energy-saving computation offloading in fog computing networks," *The Journal of Supercomputing*, vol. 80, no. 1, pp. 620–641, 2024, doi: 10.1007/s11227-023-05484-w.

[29] H. Liu, Z. Niu, J. Du, and X. Lin, "Genetic algorithm for delay efficient computation offloading in dispersed computing," *Ad Hoc Networks*, vol. 142, 2023, doi: 10.1016/j.adhoc.2023.103109.

[30] P. Trojovsky, M. Dehghani, and P. Hanus, "Siberian tiger optimization: a new bio-inspired metaheuristic algorithm for solving engineering optimization problems," *IEEE Access*, vol. 10, pp. 132396–132431, 2022, doi: 10.1109/ACCESS.2022.3229964.

APPENDIX

Table 1. Summary table

Author	Method	Advantage	Limitation
Lin *et al.* [25]	SCA-based interior-point method	It improves convergence speed through effectively solving non-convex which makes it suitable for managing complex resource allocation.	It suffered from local optima convergence because of primary feasible points which limits the performance in high dynamic fog computing.
Kök and Özdemir [26]	DRLOS	The DRLOS optimizes the computational offloading in fog computing through learning dynamically and adapting complex environments which leads to effective resource allocation and less latency.	It suffered from higher computational overhead and long convergence because of complex training in dynamic environments.
Dang and Kim [27]	DISCO	The DISCO effectively balances computational load among cloud resources to reduce latency and energy consumption.	The DISCO increased latency because of overhead of handing distributed resources among numerous fog nodes.
Li *et al.* [28]	SPRA-ICO	The SPRA-ICO enhances the computing through optimizing resource allocation and maintaining less latency thereby securing data and reduced computational complexity.	It leads to increased overhead in latency and communication due to complexity of tasks and handling resource allocation among distributed nodes which affects the performance.
Liu *et al.* [29]	GA	The GA algorithm effectively explores search space due to its ability and frequency tuning thereby leading to optimal resource allocation.	It suffers from premature convergence that leads to suboptimal solutions which not effectively explore the search space.

BIOGRAPHIES OF AUTHORS

Srikanta Murthy Rajini is currently an associate professor at the Vidyavardhaka College of Engineering, in Department of Information Science and Engineering. She obtained her Ph.D. in Computer and Information Sciences under Visvesvaraya Technological University, Belagavi, Karnataka, India. Her area of interest includes wireless sensor networks, IoT, big data analytics, and machine learning. She can be contacted at email: rajinis@vvce.ac.in.

Reginald Shilpa received a Bachelor of Engineering in Electronics and Communication Engineering from Visvesvaraya Technological University, Belgaum in 2003, an M.Tech. in VLSI Design and Embedded Systems in 2007, and a Ph.D. degree in the domain of signal processing from Visvesvaraya Technological University, Belgaum in 2019 respectively. She is working as an associate professor at Department of Electronics and Communication Engineering, Vidyavardhaka College of Engineering, Mysore. She has 21 years of teaching experience. She can be contacted at email: shilpa.r@vvce.ac.in.