Journal of
Data Acquisition and Processing

ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.778241

BLOCKCHAIN-BASED MULTI-FACTOR AUTHENTICATION FOR SECURE FOG-CLOUD ENVIRONMENT

Saurabh Singhal, Asheesh Tiwari
Department of Computer Engineering & Applications, GLA University, Mathura India

Abstract:

It has been suggested that fog computing might be one of the leading possibilities for Internet of Things (IoT) applications in the not too distant future. On an environment with fog aided Internet of Things (IoT), the most important research problem is resource management on large-scale networks. In the context of the fog-cloud environment, a significant amount of research has been carried out on the scheduling and offloading of tasks. None of these publications address the need of maintaining security while managing fog resources. To be more specific, it is possible that illegal users have been making advantage of the fog's resources in more recent times. In this research, we examine the problem described in the above sentence and provide a solution using our innovative triune-layered design. Device layer, fog layer, and cloud layer are the three components that go into the construction of a revolutionary triune layer. The Multi-Factor Blockchain Biometric Authentication (MFB2A) is given at the device layer to identify the authorized users. The Montgomery Curve Encryption (MCE) is proposed to provide overall security for data. The whole work is going to be implemented in the iFogSim simulator so that the performance of the proposed work may be evaluated. The findings that have been made on time efficiency (response time, delay time, and offloading time), throughput, and energy consumption reveal encouraging outcomes in an Internet of Things environment that utilizes fog computing.

1. Introduction

The Internet of Things (IoT) provides a broad variety of possible applications for computing in the fog, a relatively new technology [1, 2]. The bulk of Internet of Things (IoT) applications need low latency and a large user capacity. So, permitting fog computing will be enough to satisfy IoT needs. However, data security and overloading are the main issues in the fog-based Internet of Things environment [3]. There is a higher chance that fog assets will be misused via illegal access when unauthorized people are present. This has the immediate effect of making the resources unavailable to those who are properly allowed to use them. In order to prevent unauthorized users from using the fog resources, protection is necessary [4]. The response time and latency will significantly rise if there are a lot of authorized users present, since the fog nodes will get overwhelmed [5]. Fog offloading is an important consideration for Internet of Things applications that has to be made as soon as feasible. Fog-cloud research has received a lot of attention in order to achieve this aim [6]. The picture serves as an illustration of the ecology between fog and the Internet of Things.1. A substantial amount of research has been done on IoT device authentication in the context of the fog-cloud environment [7, [8]. One of the best ways to solve the issue of stopping unauthorized people from accessing fog resources is usually thought to be authentication. The bilinear parings and key agreement technique has been utilized for authentication purposes in IoT healthcare applications that leverage fog computing [9]. The authentication method also makes use of passwords and fog IDs. A secured mutual authentication system called SMAP is presented [10] for the purpose of authenticating fog-cloud devices. The SMAP protocol was developed using the hash function, a pseudo-random number generator, and timestamps. In this case, eliminating the process of super fog key storage and continually creating keys solves the overhead problem. However, all of these credentials are vulnerable to hacking by the attackers, which exposes them to a number of dangers. As a consequence, a trustworthy authentication system is required to increase the level of security in general.

On the other hand, management of the fog resource entails offloading methods and efficient task scheduling. Additionally to providing the IoT consumers with a guaranteed Quality of Service (QoS) level, it manages the fog resource. Two integer linear programming (ILP) models with task scheduling in cloud IoT with fog were explored in this talk [11]. In this situation, scheduling is done with the goal of increasing the network's energy efficiency. An evolutionary-based task scheduling technique has been created in order to address the problem of job scheduling in fog computing [12]. A particle swarm optimization (MPSO) technique with several variations is provided in order to accomplish this goal. Makspan, execution time, and response time are all taken into account while scheduling the jobs. The possibility of a fog node being overloaded still exists even if task scheduling is in charge of controlling the fog's resources. The unloading of work is a crucial component in this circumstance. The method for job scheduling is accompanied by the FEMTO task offloading algorithm, which is fair and energy-efficient [13]. The fairness scheduling metric serves as the cornerstone of the whole FEMTO algorithm. This statistic considers the energy use of jobs that are offloaded, the historical average energy of the fog node, and the priority of the fog node. The ETCORA algorithm [14] does something similar by combining the processes of work offloading and resource allocation. The ETCORA algorithm considers simultaneous optimization of delay and energy consumption. The offloading node that is found to be the most efficient in this stage is chosen based on a variety of different factors. Combinatorial multi-armed bandits (CMAB) is a framework that has been created to make efficient fog offloading possible [15]. Carefully organizing and prioritizing unloading may help achieve the objective of attaining the lowest latency. As a result, either task management or security measure enforcement has been the focus of earlier research efforts.

1.1 Research Motivation

The main driving force behind this collection of work is the dearth of prior research in various areas of fog-cloud. In reality, several studies have provided a number of approaches for improving task management in a fog-cloud. However, there are very few works that have simultaneously prioritized task management and security. However, it is essential to take into account both of the fog-cloud's components in order to achieve improved performance. Any task management method has the drawback of never being able to handle the work of an illegal user. In order to achieve greater levels of performance, we are motivated to provide a secure solution for task scheduling and offloading in a fog-cloud scenario. Specifically, we said that our objective was "to design a secure fog-cloud architecture that supports necessary QoS for the users." To put it another way, we wanted to guarantee that customers could get the level of service they need.

1.2 Roadmap of the paper

The remaining parts of this essay are structured as follows: In Section II, an overview of the currently available research papers on fog IoT environments is presented. In the third portion of the paper, the research issues and the overarching problem statement are both defined. In section IV, the proposed architecture is broken out in great depth along with all of the algorithms that have been presented. The suggested work is evaluated in Section V via the use of simulations, and a comparative analysis is also included in this section. The conclusion of the contributions may be found in section VI, which focuses on the future research areas.

2. Related Works

Multiple techniques have contributed to the improvement of fog security. For instance, a technique known as Soft Hesitant Fuzzy Rough Set (SHFRS) was given in order to choose the most suitable security service [16]. Combining fuzzy theory with roughest theory resulted in the presentation of the SHFRS. It had submitted a solution to the challenges of decision making based on many factors. The upper and lower approximation operators for SHFRS are presented in this article. This work chooses the most effective security service even for unauthorized users, which renders the fog nodes inaccessible to the users who are really permitted to use them. Multiple authorities were permitted in the fog environment so that users of the Internet of Things could be authenticated [17]. This suggested method is based on a layered fog-cloud system, in which both the fog layer (consisting of smart devices, fog nodes, and local certificate authorities), and the cloud layer (consisting of a trusted certificate authority and a public cloud server), are structured in a layered way. In order to get a certificate for authentication, each and every smart device has to first be registered with both LCA and TCA. During the production of certificates, both LCA and TCA conduct a Rivest Shamir Adleman (RSA)-based key generation for each individual user. A key based on the Paillier algorithm was produced for each user in order to facilitate encryption. The cloud server was used to decrypt the data after it had been encrypted by the smart devices, collected by the fog nodes, and then accumulated. The engagement of RSA and Paillier schemes here causes an increase in the amount of time needed. In addition, both algorithms involve complicated operations, which disqualifies them as candidates for use in lightweight Internet of Things devices.

For the purpose of achieving safe data aggregation in fog-based healthcare applications [18], a lightweight encryption approach was developed. Before being saved to the cloud, the data produced by smart devices went through an encryption process that was handled by a rather simple algorithm. Even if this effort secures the data, it is still difficult for unauthorized users to access the network. This means that any unauthorized user may enter the fog environment to encrypt their data, which results to an excessive waste of resources. In order to authenticate users of IoT devices, a fog-cloud environment has been provided with a mechanism for authenticated key agreement [19]. Authentication of the IoT user was accomplished with the help of the smartcard in this scenario. The user's details, including their ID, password, and biometric information, were implanted in the smartcard. The fog ode will authenticate the user by verifying their credentials every time they input their smartcard into the system. However, in order to access the data, the user is required to carry the smartcard at all times. In addition, the smartcard may be altered or misplaced with relative ease, all of which contribute to the insecurity of this system. It is obvious from the research that the security mechanism must be able to prevent unauthorized user access and must be lightweight enough to be handled by IoT devices. Additionally, it is imperative that this be accomplished without compromising the security of the system.

The literature review leads us to the conclusion that the majority of the publications that are now available either focus on management of tasks or security. Because of this, the work on task scheduling was completed only with the scheduling phase, and it did not take into account the task management process, which is the most essential in a fog-cloud context. Therefore, in order to attain higher levels of performance, it is vital to develop a fog-cloud environment that is safe and has an efficient task management procedure.

Fog security scheme (FSS) was provided with the purpose of ensuring end-to-end security with the efficient authentication method that follows the RSA crypto function [28]. The authentication credentials that are being examined, such as an ID and a password, offer a lesser degree of security and are easy to fake. Particularly, taking into consideration just one of these credentials leads in a significant level of vulnerability. In addition, RSA requires a longer amount of computing time and has a greater overhead, neither of which are acceptable for the resource-constrained IoT devices. With a first-in-first-out (FIFO) policy based multi-level feedback queuing system [29], a deadline and priority aware task offloading (DPTO) technique was developed. The FIFO approach often results in an increase in the amount of waiting time allotted to activities that have both a short execution time and slack time. In DPTO, the priority value of the tasks that are in the low priority (LP) queue are given high priority (HP), and those tasks are then completed, in order to control the task hunger. Increasing the priority of LP jobs will, however, have an effect on the amount of computing time spent on HP tasks. The Firefly method is provided with two goal functions to choose the best fog node for offloading [30]. These objective functions are energy consumption and computing time. Both goals are wholly reliant on the qualities of the tasks to be completed. On the other hand, in order to reduce the frequency of offloading, it is important to take into consideration the peculiarities of the fog. Since not taking into account the present condition of the fog while offloading would make the problem of overloading among fog nodes worse. When doing a local search, the Firefly algorithm is quite inefficient. As a result, the Firefly algorithm is not an appropriate choice for picking the best fog for offloading. In addition, the best solution is influenced by the settings of the firefly control since it has a huge number of parameters that need to be tweaked. A neuro-fuzzy algorithm and a PSO algorithm were used in order to offer a secure offloading mechanism (Existing) for the fog-cloud Internet of Things [31]. PSO has a sluggish convergence and often gets stuck in the local optimal solution, which does not provide optimal results. Processing illegal Internet of Things tasks results in underutilization of available resources.

IV. Proposed MFB2A based Fog-cloud

In this section, we explain the proposed MFB2A architecture in detail. Each process involved in the proposed architecture is shown in fig.1 and described in the following subsections.

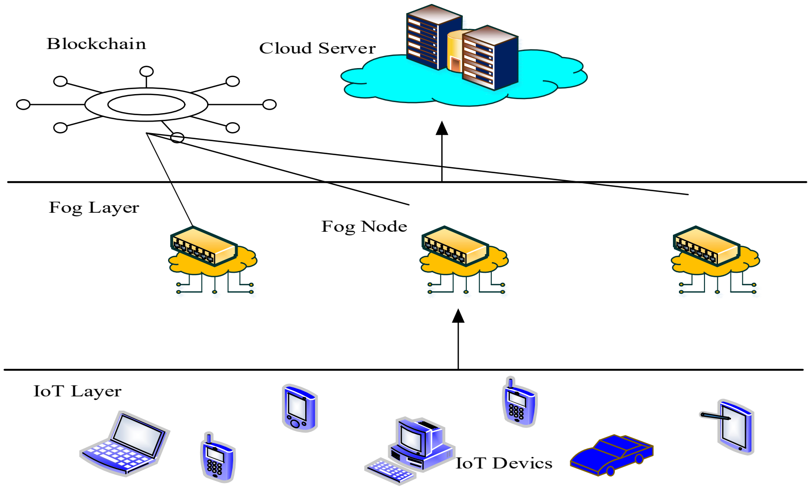

Figure.1 Proposed MFB2A architecture

3.1 MFB2A Model

We offer a special MFB2A architecture to enhance Quality of Service (QoS) in the fog-cloud environment. The MFB2A model is shown in its entirety in the image.2. The three layers that make up the MFB2A's construction may be split down into the following specifics:

There are n separate IoT devices in Layer 1, which is also known as the IoT device layer. IoT devices are available to Internet of Things users, enabling them to assign tasks to the fog layer. From this vantage point, the article's Internet of Things device and user both represent the same thing. The notations D_1, D_2, etc. up to D_n may be used to identify the devices. Since it is believed that Internet of Things devices have limited resources, they delegate their duties to the fog layer in order to save those resources. The task transfer between the device layer and the fog layer may be avoided by passing via the Gateway (g).

Layer 2 is the fog layer, according to the proposed hybrid architecture, and it is where the fog nodes are arranged. The m number of fog nodes (F_1,F_2,..,F_m) and the k number of super fog nodes (M_1,M_2,..,M_k) are grouped together in this layer. The tasks that are sent by IoT devices must be completed by the fog nodes, and the super fog nodes are in charge of overseeing the load that is spread among the fog nodes. We'll assume that km is constantly in use. The generation of k fog groups utilizing super fog nodes takes place in this layer.

Layer 3: At this level, there are cloud servers, which provide the network enough resources.

4.2 Multi-Factor Blockchain Biometric Authentication

The first stage of the MFB2A plan is identifying users who are attempting to utilize the fog resources without authorization. Based on the MFB2A protocol, we provide an innovative approach to authentication. The MFB2A approach, which combines PUF together with a simple encryption algorithm, ensures strong authentication. Device Registration and Authentication are the two main phases of the MFB2A technique, and they are described in the following,

Phase 1 (Registration of Devices) - All Internet of Things users must first sign up with the g. We consider three crucial factors while registering: an ID, a password (PW), and a personal user name (PUN). An Internet of Things device's PUF may be thought of as its own digital fingerprint since each one is different. PUFs are very secure and difficult to hack into. A PUF of the i-th device, marked by the letters "D_i," may be represented by a set of challenges and answers, denoted by the letters "C_iR_i." A different challenge and response combination than the first device is used by a separate device. The PUFs are first and foremost a part of the devices themselves. Because of this, the attackers are unable to steal or decrypt the PUFs. As a consequence, the level of security is greatly raised when PUFs are included. The applicable device's secret key S_K (i) will be generated during this step, and the g will also store the IoT devices' authentication information. The three parts that make up D_i's authentication credentials are ID_i, PW_i, and PUF_i.

Second phase (authentication): The devices are permitted after they have registered with the g, allowing the fog layer to take over the tasks they were doing. The MFB2A approach employs a two-stage authentication procedure. The device's user ID and unique password were both verified in the first phase. A device will go on to the second fold if it can successfully finish the first fold.

4.3. Data Security by MCE

The MCE crypto function, which is the lightweight encryption algorithm, is modified in the first fold of the process. The user begins by providing their ID and password in an encrypted format. Because the ID and password are susceptible to being broken by the adversaries, we encrypt the credentials using the MCE technique. The Elliptic Curve Encryption (ECE) algorithm's method is followed in order to carry out the execution of the MCE algorithm. In a traditional implementation of the ECC method, the creation of the key is performed on the elliptic curve. In the MCE, the elliptic curve is taken to be the Montgomery curve, which is specified by the equation that is shown below,

$$M_{Curve}(A,B): By^2 = x^3 + Ax^2 + x \qquad (1)$$

This is where the curve described by (A,B) over the K field is defined. In such a way that A,B = K, A2,B = 0, and B(A2-4) = 0. On the Montgomery curve, a point (P=(x,y)) may be written as the Montgomery coordinates as P=(X:Z), where x=X/Z, provided that Z is less than zero. The following is a definition of the modular multiplication on the Montgomery curve,

$$Mon(x,y) = xyR^{-1} mod\, \mathfrak{N} \qquad (2)$$

In this case, x and y are used to designate points on the curve, while N is used to denote the modulo. The following is an example of how the process of key creation is carried out,

$$S_K(i) = S_R * P \qquad (3)$$

In which S_K (i) is the public key of the i-th device, and S_R is a random integer that serves as the device's private key. Where S_K (i) is the public key of the i-th device. Another important point on the Montgomery curve is denoted by the letter P. The i-th gadget is responsible for carrying out the first fold encryption in the following manner,

$$En\{Data\} = \{Ct_1, Ct_2\} \qquad (4)$$

Where, cipher text 1 (Ct_1) and 2 (Ct_2) are computed as,

$$Ct_1 = q * P \qquad (5)$$

$$Ct_2 = (ID \oplus PW) + q * S_K(i) \qquad (6)$$

In this case, q is chosen at random within the stated range. Following the completion of encryption, the EnID and PW are sent to the g by the D_i. After that, the g will decrypt the credentials in order to validate the identification of the device. If the credentials that were decrypted from the g are a match for the credentials that were saved in the g, then the device successfully completes the first fold of authentication. At the first fold, all devices that have not yet successfully completed the first authentication fold will be disregarded. For the second fold, many other devices are being examined. The PUF is checked for accuracy as part of the device validation process during the second fold. The equivalent challenge, denoted by C_i, is sent to the D_i by the g. In the second fold, the devices that have the matching answer integrated inside them are the only ones that can produce a correct response. Other devices, or devices that are not allowed, are unable to produce the correct answer, which may be readily detected. Upon being provided with C_i, the D_i will immediately answer with the matching response, which may be found below,

$$Res \rightarrow \{(\mathbb{R}_i \oplus ran), TS\} \qquad (7)$$

In such case the answer is computed using R_i, a random integer, and the current time stamp (TS). During this fold, the g subunit removes the R_i from the Res subunit and verifies the device. If the C_i and R_i values are both correct, then the device has been properly authenticated and is permitted to hand over the duties to the fog layer. If this is not the case, the request will be ignored, and the device will not be granted access to the fog layer.

IV. Experimental Evaluation

We used iFogSim, which is an effective tool for modeling and simulating fog computing, to create a model of the proposed MFB2A. The iFogSim application operates on top of the cloud layer, which makes use of the functionality provided by CoreCloudSim. Since the program is developed in the Java programming language, we must to install the Java Development Kit (JDK) in addition to iFogSim. We have been using the Netbeans IDE to carry out the execution of the iFogSim library. The Windows 7 operating system is being used by each of our applications and solutions here. After the installation has been finished, we will put the project into action by assigning the appropriate software and hardware configuration.

Delay Analysis

The measure of quality known as delay demonstrates how well a task is providing the desired level of quality of service (QoS). The majority of Internet of Things applications cannot tolerate a significant latency.

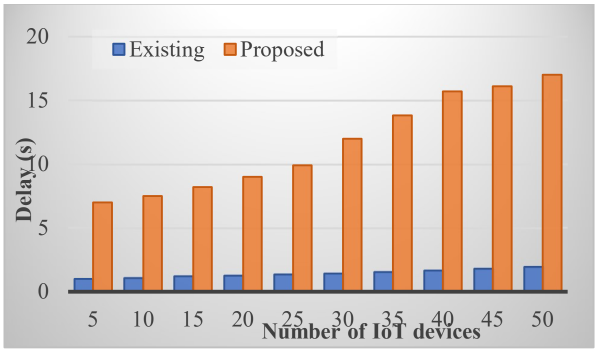

Fig.2 Analysis on delay

Figure 2 compares the delay metric between the existing research activities and the anticipated research tasks. The proposed work keeps the delay for fifty Internet of Things devices under 2 seconds, while the DPTO and Existing work keep the delay for the same number of IoT devices at roughly 7 and 17 seconds, respectively. One definition of delay is that it is the measurement of the time needed to complete an action. Many of the difficulties brought on by IoT devices must be overcome by the fog-IoT system. The amount of jobs that need to be finished determines how long the delay will be. On the other hand, in the work that has been provided, the delay is maintained between s and 2s for a substantial number of actions coming from 50 IoT devices. The main cause of this is because we provide PAS scheduling together with the best queue management technique. scheduling is done in a way that the proposed work, which is its main objective, helps to prevent a task famine that may otherwise happen. First-in, first-out (FIFO) scheduling is used in DPTO, but in Existing, scheduling happens after dangerous data has been found. There is thus a significant delay. Most importantly, when employing either way, the tasks may also be executed from unauthorized devices. This causes all jobs that have been authorized to be significantly delayed. The delay is decreased as a consequence of the planned MFB2A's usage of the TF-PUF approach to get rid of unauthorized devices.

Throughput Analysis

The entire performance of the proposed system may be measured by its throughput. If the newly incoming jobs are handled in an effective manner, then the throughput will be quite high. Security Level Analysis

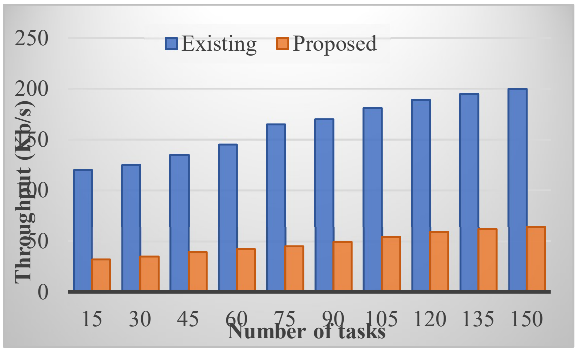

Fig.3 Analysis on throughput

Figure 3 compares the throughput that can be achieved by the present works and the proposed works. The research's conclusions show that the MFB2A that was provided had a greater throughput. The study's results indicate that the recommended MFB2A can outperform earlier research efforts in terms of throughput. The existing technology has a throughput limit of 64 kbps, which is 136 kbps less than what is anticipated for the MFB2A. Compared to the already in use DPTO and Existing methods, the throughput is maintained between 120kbps and 200kbps, which is a major gain. The main driving force behind the given achievement was MFB2A's initial eradication of all illegal jobs, which resulted in a rise in throughput. The biggest problem with earlier efforts is that they include forbidden tasks. It is plainly clear from the throughput study's findings that the recommended MFB2A is capable of attaining increased performance.

V. Conclusion

The usage of fog computing may, in the not-too-distant future, represent one of the most exciting possibilities for IoT applications. The most significant research difficulty in a setting where fog is employed to support IoT technologies is the management of resources on large-scale networks. There has been a lot of research done on the scheduling and offloading of activities within the context of the fog-cloud environment. None of these papers discuss the need of maintaining security while controlling fog resource use. To be more accurate, it's likely that illegal users have begun using the resources of the fog more recently. This would go against the fog's intended function. The problem posed in the above sentence is examined within the purview of this research, and a solution is provided by using our state-of-the-art triune-layered architecture. A gadget layer, a fog layer, and a cloud layer combine to form a breakthrough triune layer. These are the three elements that are used to construct the layer. At the device layer, the Multi-Factor Blockchain Biometric Authentication, also known as MFB2A, is available to identify users who are authorized to access the blockchain. A technique with the potential to provide complete data security is the Montgomery Curve Encryption, or MCE. The whole project will be carried out within the iFogSim simulator in order to evaluate how successful the planned work was. Studies on time efficiency (response time, delay time, and offloading time), throughput, and energy utilization in an Internet of Things context using fog computing point to positive results.

References

1. Fan, Q., & Ansari, N. (2020). Towards Workload Balancing in Fog Computing Empowered IoT. IEEE Transactions on Network Science and Engineering, 7, 253-262.
2. Aazam, M., Zeadally, S., & Harras, K.A. (2018). Offloading in fog computing for IoT: Review, enabling technologies, and research opportunities. Future Gener. Comput. Syst., 87, 278-289.
3. Zhang, P., Zhou, M., & Fortino, G. (2018). Security and trust issues in Fog computing: A survey. Future Gener. Comput. Syst., 88, 16-27.
4. Verma, U., & Bhardwaj, D. (2018). Security Challenges for Fog Computing enabled Internet of Things from Authentication perspective. Proceedings of International Conference on Computational Intelligence & IoT (ICCIIoT).
5. Jia, X., He, D., Kumar, N., & Choo, K.R. (2018). Authenticated key agreement scheme for fog-driven IoT healthcare system. Wireless Networks, 1-14.
6. Pardeshi, M.S., & Yuan, S. (2019). SMAP Fog/Edge: A Secure Mutual Authentication Protocol for Fog/Edge. IEEE Access, 7, 101327-101335.
7. He, Z., Zhang, Y., Tak, B., & Peng, L. (2020). Green Fog Planning for Optimal Internet-of-Thing Task Scheduling. IEEE Access, 8, 1224-1234.
8. Nguyen, B.M., Binh, H.T., & Son, D.B. (2019). Evolutionary Algorithms to Optimize Task Scheduling Problem for the IoT Based Bag-of-Tasks Application in Cloud–Fog Computing Environment. Applied Sciences.
9. Zhang, G., Shen, F., Liu, Z., Yang, Y., Wang, K., & Zhou, M. (2019). FEMTO: Fair and Energy-Minimized Task Offloading for Fog-Enabled IoT Networks. IEEE Internet of Things Journal, 6, 4388-4400.
10. Sun, H., Yu, H., Fan, G., & Chen, L. (2019). Energy and time efficient task offloading and resource allocation on the generic IoT-fog-cloud architecture. Peer-to-Peer Networking and Applications, 13, 548-563.
11. Wang, K., Tan, Y., Shao, Z., Ci, S., & Yang, Y. (2019). Learning-Based Task Offloading for Delay-Sensitive Applications in Dynamic Fog Networks. IEEE Transactions on Vehicular Technology, 68, 11399-11403.

12. Rathore, S., Sharma, P.K., Sangaiah, A.K., & Park, J.H. (2018). A Hesitant Fuzzy Based Security Approach for Fog and Mobile-Edge Computing. IEEE Access, 6, 688-701.
13. Guan, Z., Zhang, Y., Wu, L., Wu, J., Li, J., Ma, Y., & Hu, J. (2019). APPA: An anonymous and privacy Preserving data aggregation scheme for fog-enhanced IoT. J. Network and Computer Applications, 125, 82-92.
14. Ullah, A., Said, G., Sher, M., & Ning, H. (2019). Fog-assisted secure healthcare data aggregation scheme in IoT-enabled WSN. Peer-to-Peer Networking and Applications, 1-12.
15. Chen, C., Huang, Y., Wang, K., Kumari, S., & Wu, M. (2020). A secure authenticated and key exchange scheme for fog computing. Enterprise Information Systems.
16. Sharma, S., & Saini, H. (2019). A novel four-tier architecture for delay aware scheduling and load balancing in fog environment. Sustain. Comput. Informatics Syst., 24.
17. Xu, J., Hao, Z., Zhang, R., & Sun, X. (2019). A Method Based on the Combination of Laxity and Ant Colony System for Cloud-Fog Task Scheduling. IEEE Access, 7, 116218-116226.
18. Bhatia, M., Sood, S.K., & Kaur, S. (2019). Quantum-based predictive fog scheduler for IoT applications. Computers in Industry, 111, 51-67.
19. Boveiri, H.R., Khayami, R., Elhoseny, M., & Manogaran, G. (2018). An efficient Swarm-Intelligence approach for task scheduling in cloud-based internet of things applications. Journal of Ambient Intelligence and Humanized Computing, 1-11.
20. Wang, T., Zhou, J., Liu, A., Bhuiyan, M.Z., Wang, G., & Jia, W. (2019). Fog-Based Computing and Storage Offloading for Data Synchronization in IoT. IEEE Internet of Things Journal, 6, 4272-4282.