Load balancing strategy for medical big data based on low delay cloud network

Huafu Yong

1Department of Electrical & Information Engineering, Sichuan Engineering Technical College, Deyang, Sichuan, 618000, People’s Republic of China

E-mail: yonghuafu@scetc.edu.cn

Abstract: Since the cloud servers are far away from the medical detection terminal and user terminal, the communication overhead such as the delay caused by data transmission is large. At the same time, a large number of medical terminals and user terminals access the cloud servers, which makes the cloud servers overloaded, the overall robustness of the cloud distributed computing centre poor, and the network is prone to failure, which may lead to the work efficiency of doctors cannot be guaranteed, and the waiting time of patients will also increase. To solve the above problems, according to the characteristics of dynamic resource allocation in the medical big data environment, a new cloud network architecture is proposed. To solve the resource scheduling problem, a chaotic algorithm is introduced into the artificial firefly algorithm, and a load balancing optimisation strategy based on a chaotic firefly algorithm is proposed. The simulation results show that the convergence rate of the proposed algorithm is accelerated by adding chaos factor, to avoid the algorithm falling into the local optimal solution. Compared with other load balancing algorithms, the proposed algorithm is more suitable for solving the resource scheduling problem of large-scale tasks in cloud-fog networks.

1 Introduction

With the development of the internet of things technology, especially the development of the internet of vehicles, wearable devices, and other technologies, people's demand for service quality is higher and higher, especially for low latency. As a supplement to cloud computing, fog computing extends cloud computing to the edge of the network, making up for the shortcomings of cloud computing. On the basis of not changing the original network architecture, it provides users with computing, storage, and other services on the edge of the network, reduces the processing delay of user requests, and meets the low delay requirements [1–4]. Fog computing makes up for the lack of cloud computing and has been widely concerned by experts and scholars. In 2012, Cisco proposed fog computing, introduced the characteristics of fog computing and its application in the Internet of things, including the Internet of vehicles, smart grid etc., and analysed the interaction between fog and cloud [5, 6].

Nowadays, the proportion of doctors and patients is seriously out of balance. Mining and analysing big medical data, using the valuable information to assist doctors in the diagnosis and treatment of diseases, and improving the efficiency of doctors’ diagnosis have become an urgent problem to be solved. As the support platform for analysis and processing of medical big data, cloud computing provides strong technical support for hospital information construction [7, 8]. In recent years, experts and scholars have proposed a medical big data mining platform and algorithm based on cloud computing. The authors of [9] studied the initial implementation and application of fog computing platform, and proposed that patients can use fog nodes to store and manage their medical data, but does not solve the calculation service of medical data. In [10], a medical information physical system based on fog computing is proposed by integrating fog computing and the medical information physical system, and the cost-benefit of the system is modelled. The linear programming based on a two-stage heuristic algorithm is proposed to reduce the computational complexity, but how to solve the problem of medical big data processing is not mentioned. In [11], the authors proposed a service-oriented fog data architecture to deal with telemedicine data, which can reduce data storage and transmission power consumption, but the fog data system architecture studied only for wearable medical devices. Considering that the fog network is usually composed of some scattered, weak computing power switches, routers and other network devices, and it is difficult for a single device to effectively process a large number of data. In [12], the authors proposed that the fog nodes carry out distributed computing to reduce the computational complexity and improve the computing speed, but did not propose specific distributed computing strategies. In [13], to improve the satisfaction of the Internet of things users with fog computing, this authors proposed an improved spectral clustering algorithm (ISCM). ISCM algorithm uses a spectral clustering algorithm to reduce the dimension of the matrix and solve the eigenvector. On the basis of the improved k-means algorithm, we add the initial cluster centre selection algorithm and exclude the irrelevant data points when selecting the initial cluster centre to reduce the amount of data calculation. However, because the cloud server is far away from the end-user, the long-distance data transmission will not only occupy a lot of network bandwidth but also increase the transmission delay, which will reduce the efficiency of doctors’ diagnosis and increase the waiting time of patients; at the same time, with the increasing amount of medical data, cloud-based services will increase the burden of cloud servers.

To be able to apply fog computing to the field of medical big data and solve the business processing delay problem of cloud computing centre architecture, this study proposes a hybrid network architecture of medical big data cloud fog for medical big data scenarios. In this architecture, a ‘fog computing’ layer is built between the cloud server and the terminal by using the edge network devices such as routers or switches in the hospital. By moving the computing service of medical data on the cloud to the fog device for processing, the processing delay of medical service is reduced, the computing load of the cloud server is reduced, and the overall robustness of the network is improved. To further optimise the service processing delay of the above network architecture, the load balancing strategy of fog computing network is studied, which can balance the network load and improve the task execution efficiency. The main innovations of this study are as follows:

(i) To reduce the delay of data transmission, hybrid network architecture of a medical big data cloud is proposed. The cloud server as a whole is regarded as a distributed computing node, and the cloud distributed computing network is formed together with
the fog network. The network architecture is mainly divided into three layers: cloud computing layer, cloud computing layer, and perceptual layer to achieve global information sharing.

(ii) In addition, the firefly algorithm in the standard mode has only local optimisation and low accuracy. In [14], chaos algorithm and artificial firefly algorithm were used to design the controller fractional-order proportion integration differentiation, and good results were achieved. In this study, the chaos algorithm is introduced into an artificial firefly algorithm, and a chaos firefly optimisation algorithm is used to solve the above optimisation problems.

The comparison results show that the improved algorithm uses the updated fog computing model and chaos factor to accelerate the convergence speed, avoid the algorithm falling into the local optimal solution, and is more suitable for solving the resource scheduling problem of cloud and fog network with large-scale tasks.

2 Proposed medical big data cloud network architecture

The proposed hybrid network architecture of the medical big data cloud is shown in Fig. 1. The network architecture is mainly divided into three layers: the bottom layer is the perceptual layer, which is mainly composed of health monitoring sensing terminals, such as sphygmomanometer, temperature sensor, smart watch, smart wheelchair, smart home, and other terminal devices. The perceptual layer is responsible for sensing and detecting the data related to the living environment of the ward and the ward, and uploading the detected data to the fog edge server responsible for the area. The middle layer is the fog computing layer, which mainly consists of gateways, routers, base stations, and other idle CPU devices and hard disks that are located at the edge of the network and can provide some distributed computing, storage, and resource management capabilities. The fog computing layer is responsible for collecting the detection data of the terminal sensing and monitoring equipment, and intelligently analysing and processing the detection data from the sensing layer to quickly respond to the task request of the sensing layer. Upward is responsible for uploading the analysis results of the sensing layer data to the cloud for further analysis and learning by the cloud server. The top layer is the cloud computing layer, which is mainly composed of cloud servers with strong computing and storage capabilities. The cloud computing layer is responsible for the final big data analysis and learning of the massive data generated by all terminal access devices of the whole health monitoring network and feeds back the learning experience to each fog server. When processing the sensing layer data, the fog server can respond more quickly and accurately according to the learning experience of the cloud server.

The hardware of fog server usually includes storage module, processor module, power management module, and communication module. The fog server of the system adopts 250 GB solid-state hard disk and cortex A8 processor chip and is equipped with Ethernet port and 3G/4G expansion port, to be able to access the Internet in an indoor and mobile environment. To support the simultaneous access of more mobile devices, the micro server is equipped with a dual antenna wireless large area network communication module in two frequency bands of 2.4 and 5.8 GHz. The main software of fog server includes an operating system, web system for data display, and queuing time calculation module. Among them, Linux 3.0 kernel is used for the operating system; Apache is used for the server of the web system, PHP is used for the server programming language, and MySQL is used for the database; the queuing time calculation module includes three sub-modules: WiFi-based positioning module, queuing behaviour identification module, and queuing time calculation module.

Fog equipment is deployed near each department of the hospital. It connects with wireless access equipment through the wire and cooperates with wireless access equipment for fast data forwarding. At the same time, according to the characteristics of surrounding departments, fog equipment downloads the analysis results of medical big data such as medical images on the cloud server through active caching and stores the medical information transmitted through it. In addition, the fog device receives and stores the medical detection data from the medical terminal, performs comparative calculation and analysis on the medical data according to the analysis results of the cached medical images and other big data, and also uploads the cached medical information and diagnosis and treatment records to the cloud server to realise global information sharing. Doctors and patients can directly obtain medical diagnosis information through computers or medical wearable devices and other terminals. Owing to the large amount of data of medical detection information such as medical image, fog computing equipment can get the diagnosis results for doctors’ reference in a short period of time through distributed computing to balance the load, and quick feedback to doctors, which greatly reduces the time of doctors’ diagnosis and treatment and patients’ waiting time.

3 Theoretical model

In the cloud-fog hybrid network for health monitoring scenario, all data related to the supervised from the perception layer are uploaded to the nearest fog edge server in the region. In the research of fog computing task processing delay optimisation in this study, due to the relatively limited resources and performance of fog edge server nodes, to achieve the goal of completing all data tasks in the sensing layer and ensuring the lowest energy consumption, the fog computing network needs the cooperation of all fog edge server nodes to optimise the deployment of all tasks in the fog computing network. Therefore, this study proposes that the fog edge server receiving the data of the ward and his living environment assigns the received tasks to other edge servers in the same fog computing range for distributed collaborative computing according to the load balancing algorithm, so as to achieve the goal of processing task energy consumption optimisation. The cloud–fog hybrid network composed of fog computing devices is considered, and its network structure is shown in Fig. 2a. The fog network node in Fig. 2a above is abstracted as a weighted undirected graph \( G = (V, E) \) as shown in Fig. 2b, where \( V = \{z_1, z_2, \ldots, z_k\} \) is vertex set, vertex \( z_i \) is fog computing equipment, \( k \) is the number of fog computing equipment. \( E = \{e_{z_1, z_2}, \ldots, e_{z_{k-1}, z_k}\} \) is the edge set, edge \( e_{z_i, z_j} \) represents the communication link between fog computing nodes. Weight \( e_{z_i, z_j} \) on the edge represents the communication delay between fog computing nodes \( \{z_i, z_j\} \).

The calculation capacity of each fog calculation node \( z_i \) in Fig. 2b is assumed to be \( \lambda_i \). In the process of task execution, the user submits task \( D \) to the connected fog computing device \( z_i \) each time, \( z_i \) divides task \( D \) into several subtasks satisfying \( d_i = \delta D \), and assigns them to the fog computing nodes including himself for
This is an open access article published by the IET under the Creative Commons Attribution License.

The total processing delay of the whole computing task \( D \) in the medical big data cloud fog hybrid network can be expressed as

\[
\tau(\delta) = \max \left\{ \frac{\delta D}{x_0} + r_{z_j}m_{z_j} \right\}_{i, j = 1, 2, \ldots, k},
\]

where \( \delta D/\lambda_j \) represents the computing delay generated by the processing subtask \( \delta_i \) of the computing device \( z_i \). \( R_{z_j} = r_{z_j}m_{z_j} \) represents the communication overhead between \( \{z_i, z_j\} \). \( r_{z_j}m_{z_j} \) represents the communication delay. In this section, the communication delay mainly includes the transmission delay, propagation delay, queuing delay, and so on are uniformly classified as other delays; among them, \( m_{z_j} \) indicates whether there is a sub-task allocation relationship between \( \{z_i, z_j\} \). \( m_{z_j} = 0 \) indicates that there is a sub-task allocation relationship, and \( m_{z_j} = 0 \) indicates that there is no sub-task allocation relationship.

Since the total processing time of distributed computing is equal to the maximum processing delay of all subtasks, to achieve the goal of minimum processing delay, we must find an optimal set of \( \delta \) to make the objective function \( \tau(\delta) \) minimum. To sum up, the whole process can be modelled as follows:

\[
\begin{align*}
\min & \left\{ \max \left\{ \frac{\delta D}{x_0} + r_{z_j}m_{z_j} \right\}_{i, j = 1, 2, \ldots, k} \right\} \\
s.t. & \begin{cases} 
\delta_i \neq 0 \\
\delta_i = 0, \sum_{i=1}^{k} \delta_i = 1 
\end{cases}
\end{align*}
\]

If the sub task to be processed on each fog computing node is \( \delta_i = \delta D \), then the sub tasks processed on each of the \( k \) fog computing nodes can form a \( k \)-dimension vector \( \delta = [\delta_1, \delta_2, \ldots, \delta_k]^T \). Considering the specific situation, the requested computing task \( D \) may come from any node, assuming that it comes from a node, from formula (1), the total delay of processing computing task \( D \) in the medical big data cloud network can be expressed as

\[
\tau(\delta) = \max \left\{ \frac{\delta D}{x_0} + r_{z_j}m_{z_j} \right\}_{i, j = 1, 2, \ldots, k}
\]

Therefore, the solution of task \( d_i \), i.e., the solution of task vector \( \delta \), which should be handled by each computing node in the fog network can be summed up as the following optimisation problems:

\[
\begin{align*}
\delta & = \arg \min \{ \tau(\delta) \} \\
\text{s.t.} & \begin{cases} 
\delta_i \geq 0 \\
\sum_{i=1}^{k} \delta_i = D 
\end{cases}
\end{align*}
\]

The search space \( I \) of the above optimisation problem is

\[
I = \prod_{i=1}^{k} [0, D]
\]

where \( d_{\text{min}} \) and \( d_{\text{max}} \) are the minimum and maximum values that can be obtained by subtask \( d_i \).

### 4 Load balancing strategy with time delay optimisation

In this study, we use the improved chaotic firefly optimisation algorithm to solve the above optimisation problem in the medical big data scene. When the chaos optimisation firefly algorithm is used to solve the optimisation problem in formula (4), firefly group \( \{ \lambda(i) \}_i \) moves in the local decision range \( I \) to find the optimal neighbour set \( \lambda \), namely \( \delta \).

Suppose that the position of firefly is \( N \), the position of the \( i \)th firefly is \( (x_i, y_i) \), and the target function of the \( i \)th firefly is \( f(x_i, y_i) \); the value of the \( i \)th firefly is \( T_i \); \( x_j(t) \) represents the position of the \( j \)th firefly of the \( t \) generation; \( I_j(t) \) represents the value of the \( j \)th firefly of the \( t \) generation. The field of vision of the firefly is updated as follows:

\[
R'_j(t+1) = \min \{ R_j, \max \{ 0, R'_j(t) + \beta(n_t - N(t)) \} \}
\]

\[
N_j(t) \geq \min \{ j: ||x_j(t) - x_j(t)\| < R'_j(t) \}
\]

\[
N_j(t) = \frac{\sum_{k \in N_j(t)} I_k(t)}{N_j(t)}
\]

The updating formula of firefly position is as follows:

\[
x_j(t) = x_j(t-1) + \left\{ x_j(t-1) - x_j(t-1) \right\}
\]

\[
I_j(t+1) = (1 - \rho) I_j(t) + \gamma J(x_j(t+1))
\]

The result of chaos optimisation is to add the state of chaos to the optimisation variables, and expand the scope of chaos to the range of optimisation variables. With the increase of the number of iterations, the closer firefly individuals are to \( x_j(t) \), resulting in a loss of differences between individuals. To prevent this phenomenon, the worst position of firefly individuals is chaotic. In this study, linear map [15] is used as the chaos optimisation model, and the iteration formula of the chaos optimisation model is as follows:

\[
x_{n+1} = \mu \times x_n \times (1 - x_n), 0 < x_n < 1
\]
\[ y_i = x_{\text{new}} + (x_{\text{max}} - x_{\text{new}}) \times x_i \]  \hfill (12)

Step 2. For formula (11), the set of multiple iteration chaotic sequences is \( y_i^{\text{new}} \).

Step 3. According to the principle of inverse mapping, formula (12) is introduced, and the set of feasible solutions is \( x_i^{\text{new}} \).

\[ x_i^{\text{new}} = x_{\text{max}} + (x_{\text{max}} - x_{\text{new}}) \times y_i^{\text{new}} \]  \hfill (13)

Step 4. After chaos mapping, fireflies pass through some individuals with probability \( q_t \), of which \( t \) is the current iteration number. The calculation formula of \( q \) is

\[ q = \frac{\ln t}{(1 + \ln t)} \]  \hfill (14)

The solution steps are as follows:

Step 1. The initial number of fireflies and the maximum number of iterations max.

Step 2. Code the individuals in the firefly algorithm and calculate the value of the fluorescein of a single firefly according to formulas (9) and (10). To give priority to the results, the firefly population is divided into \( m \) subgroups.

Step 3. Update the fluorescein value of each subgroup, specifically

(1) According to formula (9), find out the best and worst positions.

(2) According to formula (10), the position of firefly is calculated, and then according to formulas (11)–(14), the firefly is updated to get a new individual.

(3) Calculate \( L(x_{\text{new}}) \) of each firefly, if \( L(x_{\text{new}}) > L(x) \), then \( x = x_{\text{new}} \), otherwise update.

(4) If the number of iterations is less than max, go to step (1).

Step 4. If the termination condition is met, the optimisation process will end, otherwise, it will turn to step 2 to continue optimisation.

Step 5. According to the optimal individual, the optimal scheduling scheme is obtained.

The algorithm flow is shown in Fig. 3.

5 Experiment

The simulation platform used in this study is core CPU 3.0 GHz, 2 GB DDR3, Windows XP operating system. MATLAB is used as the experimental platform. The calculation capacity and communication delay of fog computing equipment in the experiment are set in reference [11]. The number of nodes in the fog layer is set to 10, and the default device is the node receiving tasks and assigning tasks. The relevant parameters of each fog calculation node are shown in Table 1. The communication delay only considers the communication delay between the device receiving tasks and the task assignment device, and the task initialisation adopts a random assignment. The computing capacity of the cloud data centre is 120,000 MIPS, and the uplink and downlink bandwidth are 2 and 1.9 Mbps, respectively. To better verify the feasibility and performance of the improved algorithm proposed in this study, under the same conditions, the improved chaotic firefly optimisation algorithm is compared with the classic load balancing algorithms such as hill clipping load balancing algorithm [16], load balancing algorithm [17], ISCM [10]. Contrast experiments were carried out.

![Improved algorithm flow](image)

**Table 1** Parameters of fog calculation node

| Parameter type            | \( z_1 \) | \( z_2 \) | \( z_3 \) | \( z_4 \) | \( z_5 \) | \( z_6 \) | \( z_7 \) | \( z_8 \) | \( z_9 \) | \( z_{10} \) |
|---------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| computing power, MIPS     | 900       | 1100      | 1000      | 1300      | 800       | 1400      | 700       | 1100      | 1200      | 1000      |
| uplink broadband, MIPS    | 83        | 87        | 78        | 84        | 87        | 79        | 85        | 82        | 85        | 88        |
| downlink broadband, MIPS  | 101       | 103       | 102       | 98        | 103       | 101       | 104       | 102       | 103       | 100       |
| one-way other delay, ms   | 1.0       | 1.1       | 1.2       | 1.2       | 1.0       | 1.2       | 1.3       | 1.4       | 1.1       | 1.2       |
From the simulation figure, it can be seen that the upper limit of delay constraint $t_{\text{max}}(d)$ and task quantity $D$ have an impact on energy consumption $W$. When the upper limit of delay constraint $t_{\text{max}}(d)$ is fixed, energy consumption $W$ increases with the increase of task quantity $D$, and task quantity $D$ is the main influencing factor of energy consumption $W$. However, in the health monitoring application scenario, to ensure the service quality of some applications that are sensitive to the delay requirements in the load balancing of each edge server in fog computing, the energy consumption optimisation of fog computing is carried out, and the impact of the upper limit $t_{\text{max}}(d)$ of the task processing delay of fog computing on the energy consumption is of great significance.

It can also be seen from the simulation diagram that when the task amount $D$ is fixed, the energy consumption of task processing will increase with the decrease of the upper limit $t_{\text{max}}(d)$ of task processing delay, i.e. the improvement of service quality. In particular, when the task size is 1000 MB, the upper limit $t_{\text{max}}(d)$ of delay is 5 s, the task processing energy consumption $W$ of fog computing is 2745.4 J; when the upper limit $t_{\text{max}}(d)$ of delay is 25 s, the task processing energy consumption $W$ of fog computing is 1851.3 J, and when the upper limit $t_{\text{max}}(d)$ of delay is relaxed from 5 to 25 s, the optimal energy consumption is reduced by 32.6%.

When task $D$ is fixed, the delay restricts the feasible solution range of load balancing of each edge server in fog computing. When the delay is small, the optimal energy allocation scheme may not meet the delay constraint condition. When the delay is large, as shown in Fig. 5, task $D$ is 1000 MB, and the delay changes from 20 to 25 s. At this time, the most energy consumption is basically unchanged, because the optimal energy allocation scheme meets the requirements. The time delay has no effect on the optimal energy consumption.

To sum up, the impact of upper limit of delay $t_{\text{max}}(d)$ on the optimal energy consumption $W$ is related to the task amount $D$. If the optimal energy allocation scheme meets the upper limit of delay $t_{\text{max}}(d)$, i.e. within the upper limit of delay $t_{\text{max}}(d)$, each edge server can complete the assigned subtasks. At this time, the upper limit of delay is relatively loose compared with the task amount allocation of the optimal scheme, the upper limit of delay has no impact on the optimal energy consumption; if the optimal scheme can when the scheme of consumption allocation does not meet the upper limit of delay $t_{\text{max}}(d)$, i.e. to say, within the upper limit of delay, each edge server cannot complete the assigned subtasks. At this time, the upper limit of delay is stricter than the task allocation scheme of fog calculation, and the upper limit of delay has a greater impact on the optimal energy consumption.

5.3 Comparison of delay performance between the proposed cloud–fog hybrid networks and other three architectures

Under the condition of unlimited energy consumption, this study simulates delay performance of the proposed scheme network, and compares it with cloud computing network, fog computing network, and single fog node. Among them, the selected single fog node is node $z_i$ that receives tasks. The results are shown in Fig. 6.

The simulation results in Fig. 6 show that when the number of physical therapy terminal requests is <500, the number of tasks to be processed is small. Although the transmission delay of the cloud is high, the processing speed of cloud computing server is much faster than that of fog and the proposed scheme, so the delay performance gap between the four is not obvious. However, with the increasing number of physical therapy terminal requests, the delay of cloud and single fog node is significantly higher than that of fog and the proposed scheme. Since the cloud server is far away from the end-user, and the bandwidth is limited, more physical therapy terminal requests data transmission to cloud for calculation will produce higher transmission delay, which makes the total processing delay significantly increased. Although the single fog node processes the task there is no transmission delay, but the computing power is too weak. In addition, it can be seen from Fig. 6 that the cloud computing centre as a distributed computing node enhances the overall computing performance of the proposed scheme, so when the number of physical therapy terminal requests
is >1500, the proposed scheme's delay is slightly lower than fog, which is very advantageous for processing delay sensitive services.

5.4 Task completion time comparison

In this section, aiming at the dynamic characteristics of the fog computing environment task, two cases of the largest scale and the smallest scale are selected for the effect comparison.

(i) Performance comparison of algorithms for small-scale tasks: Select 10–70 tasks to run on the cloud network system with 10 resources, and get the optimal scheduling scheme to complete all tasks as shown in Fig. 7. It can be seen from Fig. 7 that when the number of tasks is small, the probability of users competing for resources is small and the probability of resource conflict is also small. Each algorithm can obtain a better scheme of cloud–fog network resource scheduling performance, and the performance difference between them is not big.

(ii) Performance comparison of algorithms for large scale tasks: Select 2000–14,000 tasks to run on a platform with 10 cloud computing resources, the task completion time of the cloud–fog network resource scheduling scheme is shown in Fig. 8. It can be seen from Fig. 8 that as the number of tasks increases, the competition between tasks becomes fiercer, the probability of task conflict increases, and the time for all algorithms to complete tasks gradually increases. From the completion time of the whole task, the task completion time of the proposed improved algorithm is far less than that of the comparative algorithm. The comparison results show that the improved algorithm uses the updated fog computing model and chaos factor to accelerate the convergence rate, avoid the algorithm falling into the local optimal solution, and is more suitable for solving the resource scheduling problem of cloud–fog network with large-scale tasks.

6 Conclusion

Aiming at the high delay problem of cloud computing data centre architecture in supporting medical big data services, this study proposes a hybrid network architecture of medical big data cloud–fog. A fog computing layer is built between the medical user terminal and the cloud computing centre, which places the medical detection service closer to the user's fog computing layer, reduces the processing delay of medical service, improves the efficiency of doctors’ diagnosis, and reduces the waiting time of patients. In addition, a load balancing strategy based on the improved chaotic firefly algorithm is proposed. The result of chaos optimisation is to add chaos state into the optimisation variables, and expand the scope of chaos to the value range of the optimisation variables, which effectively solves the resource scheduling problem in the cloud network.

However, the research of fog computing network architecture and load balancing is still in its infancy. The delay of fog network and other performance parameters of fog network are mainly studied, e.g., location awareness, energy consumption, and reliability also need to be further studied.

7 References

[1] Dai, H., Yang, Y., Yin, J., et al.: ‘Improved greedy strategy for cloud computing resources scheduling’, J. Eng. 2020, 804, (1), pp. 46–59.
[2] Sarkar, S., Chatterjee, S., Misra, S.: ‘Assessment of the suitability of fog computing in the context of internet of things’, IEEE Trans. Cloud Comput., 2018, 6, (1), pp. 46–59.
[3] Li, Q., Zhao, J., Gong, Y., et al.: ‘Energy-efficient computation offloading and resource allocation in fog computing for internet of everything’, China Commun., 2019, 16, (3), pp. 32–41.
[4] Sineoepourfard, A., Garcia, J., Manip-Bruni, X., et al.: ‘Data preservation through fog-to-cloud (F2C) data management in smart cities’. 2018 IEEE 2nd Int. Conf. on Fog and Edge Computing (ICFEC), 2018, (10.1109/CFEC.2018.835732)
[5] Gopal Gupta, A.S.A.L.G., Syam Prasad, G., Nayak, S.R.: ‘A new and secure intrusion detection system for detection of anomalies within the big data: intelligent edge, fog and mist computing’ in Das, H., Barik, R., Dubey, H., et al. (Eds.): ‘Cloud computing for geospatial big data analytics’, 9 (Springer, Cham, Germany, 2019), pp. 177–190.
[6] Yang, W., Li, S., Rafique, W., et al.: ‘An offloading approach in fog computing environment’. 2018 IEEE Smart World, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IPDPS), Guangzhou, People's Republic of China, 2018 (doi: 10.1109/SmartWorld.2018.800157)
[7] Tassi, A., Maxromatis, I., Prechocki, R., et al.: ‘Agile data offloading over novel fog computing infrastructure for CAVs’. 2019 IEEE 89th Vehicular Technology Conf. (VTC2019-Spring), 2019 (doi: 10.1109/VTCSpring.2019.8746302)
[8] Yang, H.-K., Cha, H.-J., Song, Y.-J.: ‘A study on configuration of NDN based identifier using block chain in fog computing environment’, Adv. Sci. Lett., 2018, 24, (3), pp. 2020–2025.
[9] Yi, S., Hao, Z., Qin, Z., et al.: ‘Fog computing: platform and applications’. 2015 Third IEEE Workshop on Hot Topics in Web Systems, Washington, DC, USA, 2015.
[10] Gu, L., Zeng, D., Guo, S., et al.: ‘Cost efficient resource management in fog computing supported medical cyber-physical system’, IEEE Trans. Emerg. Top. Comput., 2017, 5, (1), pp. 1–12.
[11] Dubey, H., Yang, J., Constant, N., et al.: ‘Fog data: enhancing telehealth big data through fog computing’. ACM BigData 2015 The Fifth ASE Int. Conf. on Big Data, Kaohsiung, Taiwan, 2015. (10.1145/283440.283471)
[12] Shi, Y., Ding, G., Wang, H., et al.: ‘The fog computing service for healthcare’. 2015 2nd Int. Symp. on Future Information and Communication Technologies for Ubiquitous HealthCare (Ubi-HealthTech), 2015 (doi: 10.1109/Ubi-HealthTech.2015.7203325).
[13] Xu, S.Z.: ‘Research on resource scheduling algorithm of fog computing’. Qufu Normal University, 2018.
[14] Mahdieh, A.R., Mahdi, Y.: ‘Optimum design of fractional order PID controller using chaotic fireflies algorithm for a control CSTR system’, Asian J. Control, 2019, 21, (5), pp. 2245–2255.
[15] Yang, H., Liao, X.: ‘A new color image encryption scheme based on logistic map over the finite field Z N’, Multimedia Tools. Appl., 2018, 77, (16), pp. 21803–21821.
[16] Zahid, M., Javaid, N., Ansar, K., et al.: ‘Hill climbing load balancing algorithm on fog computing’. The 13th Int. Conf. on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC-2018), Taichung, Taiwan, People's Republic of China, 2018 (doi: 10.1007/978-3-308-02607-3_22).
[17] Manju, A.B., Sumathy, S.: ‘Efficient load balancing algorithm for task preprocessing in fog computing environment’. Smart Intelligent Computing and Applications, Singapore, 2019, pp. 291–298. (doi: 10.1007/978-981-13-1927-3_31).