International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Advancing Industrial IoT: A Swarm-Intelligent-Based Job Offloading in Edge Computing

[1]Dr. Krishan, [2*]Dr. Rajni, [3]Dr. Priyanka, [4]Dr. Prachi Chaudhary

Submitted: 10/01/2024 **Revised**: 16/02/2024 **Accepted**: 24/02/2024

Abstract: By integrating detectors, supervision and communication tools into manufacturing processes, the Industrial Internet of Things (IoT) raises productivity, lowers costs and improves the value of products. It is difficult to process enormous amounts of information, which makes it difficult to swiftly transition conventional sectors to edge computing. This paper provides a unique swarm-intelligent technique named boosted beetle swarm optimization (BBSO) for offloading jobs from edge gadgets to edge servers with the lowest latency and energy consumption, taking into account the rapidly growing number of industrialized edge items and heterogeneous edge servers. The presented multi-objective optimization issue considers job performance cost, consumption of energy and latency. The entire cost of assigning each work to a separate mobile edge computing (MEC) server is represented by the fitness coefficient. Using experimentation, the effectiveness of the suggested BBSO-driven offloading technique is contrasted with alternative techniques.

Keywords: *Industrial Internet of Things (IIoT), Edge Computing, Latency, Energy Consumption, Job Offloading, Boosted Beetle Swarm Optimization (BBSO).*

1. Introduction

The industry is currently buzzing with the talk of 5G or the fifth-generation mobile communication infrastructure and many connections are the primary technological obstacles that 5G networks must overcome [1]. The 5G era would unavoidably add a significant volume of data to the network, taxing the fundamental network's centralized processing capacity in addition to placing a strain on the back [2]. The idea of computing on the edge, which is comparable to computer memory and cache, has been implemented in the industry as a result of 5G's low latency needs. Users' commonly used data is placed closer to users on the border of the network to minimize latency and lighten the strain on the main network's infrastructure [3]. When calculation-intensive activities are judiciously sent to a server acting as a proxy that has the processing power to handle them and the computed results are then retrieved from that proxy server, this practice is referred as computational loading [4]. The workloads related to computation are delivered from smart gadgets to the MEC server via computing at the edge outsources. By providing cloud computing services near to people who are roaming at the edge of the internet network, edge computing closes the gap created by usual cloud technology [5]. It enables fast active reaction times as well as provides flexible and universal computing services. The main goal of the current edge computing problem is to figure out how to use intelligent devices to the edge server so that it can take advantage of the services provided by the perimeter network [6]. Several tracking, perceptual capabilities, acquisition, management detectors, mobile networking, intelligent analysis along with other techniques are integrated continuously. This results in considerable

[1]Assistant Professor Department of Electronics and Communication Engineering,Bhagat Phool Singh Mahila Vishwavidyalaya kanpur kalan ,Sonipat, Haryana , India
Krishan.bpsmv@gmail.com
https://orcid.org/0000-0003-3872-5368
[2]Assistant Professor (corresponding Author) Electronics and communication engineering University: Deenbandhu Chhotu Ram University of Science and Technology, Murthal, Sonipat
rajni.ece@dcrustm.org
0000-0002-9925-9420
[3]Associate Professor Department of Electronics and Communication Engineering, Bhagat Phool Singh Mahila Vishwavidyalaya, Khanpur kalan, Sonipat, Haryana, India
anand_priyanka10@yahoo.co.in
https://orcid.org/0000-0003-4363-3003
[4] Assistant Professor Electronics and Communications Engineering College: DCRUST, Murthal, Sonipat , India
prachi.ece@dcrustm.org
https://orcid.org/0000-0001-8308-6820

improvements in efficiency in production, product quality, utilization of resources and cost decreases [7]. Some major developments in the IIoT are the processing of data as well as quick and dependable transmission via the network. It is impractical to send all of the data that smart devices in the IoT ecosystem must process to the core network for processing [8]. The computer vision-based system for detecting product quality is implemented. Smart gadgets are equipped with cameras to identify photos of products [9]. Once the algorithm used is machine learning-trained, it is utilized to identify product appearance photos. Sent data over 5G networks allows for the scheduling of detection jobs to edge computers by configuring those [10].

Industrial internet of things (IIoT) advancement a swarm-intelligent-based job offloading in edge computing seems to be an improvement on IIoT systems by using swarm intelligence to handle task offloading in edge computing settings.

2. Related works

The study [11] explored the potential uses of the IoT in several fields, including connected gadgets, farming, building, health care, farming and environmental surveillance. The paper [12] addressed the applications of the IoT in several domains, including the automotive, embedded, smart grid, agricultural, building, environmental surveillance, and medical care businesses. The study [13] described a complete system that tackles these issues and includes the structure, design, practical implementation and assessment. Through flexible and interoperable methods, the infrastructure offered a means of gathering, organizing and routing data streams from heterogeneous cyber-physical manufacturing platforms. The study [14] presented PriModChain, a system that combines smart contracts, federated ML, ethereal block-chain, differential privacy and trustworthiness to impose privacy and reliability on IIoT data. The study [15] approach was predicated on a registration centre upon a node's initial network membership that creates public and secret information for it. Advanced functions approximating mutual authentication, secure key exchange and communication were carried out autonomously by the participating nodes after enrolment, at which point registering with the centre becomes unnecessary. The study [16] examined a human-centered IIoT-focused efficient light-safe authentication scheme. The study [17] documented to facilitate the choice of communication interfaces by offering a complete impression of the OPC's capabilities and applicability. The outcomes were examined and analyzed, with a focus on the previously described novel paradigms. The paper [18] examined a novel architecture for the Industrial IoT that leverages digital twins (DTs) to enable federated learning by capturing industrial device attributes. Realizing that DTs might introduce estimation errors from the true device state value, federated learning proposed a trusted-based aggregation to mitigate the impact of such errors. The study [19] attempted to IoT-enabled smart systems that can be integrated into smart home systems. The study [20] elucidated the meanings of the terms Industry 4.0, Industrial IoT and IoT. It draws attention to the problems facing the realization of this paradigm shift as well as the opportunities it presented. It pays special attention to the difficulties posed by the need for energy efficiency, real-time performance, relationships, compatibility, safety and anonymity. The study [21] highlighted the major opportunities for DL in IIoT in the paper. Initially, it provided an impression of several DL approaches and convolutional neural networks as well as the way they were applied in diverse sectors. The study [22] examined a safe wireless technique utilizing Blockchain software to store extracted procedures of each record into many blocks, hence maintaining transparency and securing every smart sensing action.

3. Problem definition

The intelligent manufacturing facility in the workplace sets up many devices, including smart phones, cameras with sensors and augmented reality devices, for the Industrial IoT scenarios in the 5G context. This process is known as multi-user. Multiple MEC systems are required in the manufacturing setting to manage the responsibilities of smart devices because of the volume of data that must be processed a term known as multi-MEC. This research aims to confirm that the optimal task offloading site can be found and the task offloading outcomes can be obtained by applying the BBSO offloading technique, which can minimise the overall task process latency. This study examines the IoT context on the assumption that “there are currently M smart devices and N MEC servers in use at the moment”. Only non-divisible jobs are taken into consideration and each gadget can

submit just one job. A single task could be carried out locally or transferred to MEC servers. Consequently, each assignment can be completed at one of $N+1$ potential slot. In other words, offloading to N MEC servers is feasible for both local and implementation.

4. System model

The data centre, three MECs, three example situations for power-related programs, wireless communication, tracking the environment and video surveillance comprise the N EC smart overload model for power-related IoT, as illustrated in Fig.1. It is assumed that there exist MM EC computers and NM OB devices. A service (or local) has to be given every job to complete it and the collection of all tasks can be represented as $N = \{n1, n2, n3, \ldots nl\}$. Here, $M + 1$ $vt = \{0,1,2, \ldots M\}$ has potential outcomes for the offload; if $vt = 0$, it indicates that the current job is being calculated locally; if $us = M$, it indicates that the current work is being transferred to the N^{th} MEC server. The ultimate discharge choice vector $U = \{u1, u2 \ldots ul\}$ represents the optimal assignment of a whole set of data.

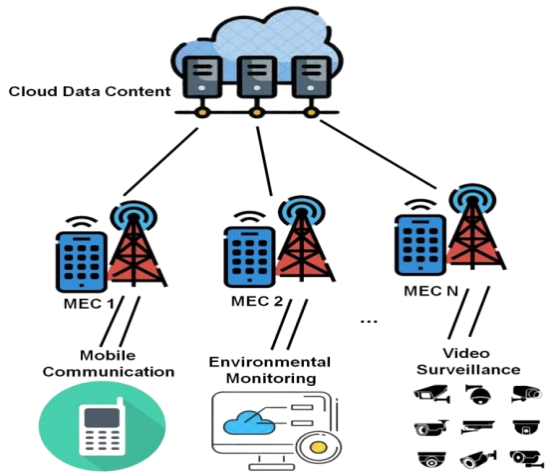

Fig.1. Power IoT MEC Smart Transfer Model

5. Time delay model

The delay between the request side communication, the MEC server surface processing and the response side receiving the processed result back is known as the server's reply lag. The entire task calculation latency is determined by adding the task transfer, MEC computing and output delays. The resultant delay is disregarded in this work as it is significantly less than the initial transmission delay. As a result, we find that the MEC computation delay and the transshipment delay are identical.

Calculation latency: The services demand process described above includes two transfer processes. These procedures are carried out through either the close of a server or a remote server via the network's core. The value $Z = \{w_{ij} =: I \in M, j \in M\}$ is introduced and here $w_{ij} = \{0, 1\}$, equation (1).

$$w_{ij} = \begin{cases} 0, server\ i\ the\ data\ request\ by\ VFj \\ 0, server\ i\ tcaches\ \ the\ data\ request\ by\ VFj \end{cases} \quad (1)$$

$Z = \{z_{ij}: I \in M, j \in M\}$ is an established vector is a term as indicated by the preceding Equation (2) consequently, we have:

$$z_{ij} = \begin{cases} 0, & locally\ executed\ at\ VFj \\ 1, & offloading\ to\ MEC\ server\ i \end{cases} \quad (2)$$

The concept of queues allows one to estimate the lag that the MEC server's CPU experiences while processing its computational operations.

$$D_i^N = \frac{z_{ji} d_j \lambda_j}{v_i - \sum_{j=1}^{N} z_{ji} \lambda_j} \quad (3)$$

Here, $z_{ji} d_j \lambda_j$ represents the overall amount of requests that are sent to the MEC server for the process and $1/(v_i - \sum_{j=1}^{N} z_{ji} \lambda_j)$ represents the mean processing delay for these requests. This fulfils the needs. $v_i - \sum_{j=1}^{N} z_{ji} \lambda_j > 0$. Comparatively speaking, consistency for processing on the consumer's side could be achieved as equation (4).

$$D_i^V = \frac{(1 - \sum_{i \in A_j} z_{ji}) d_j \lambda_j}{\theta_i - (1 - \sum_{i \in A_j} z_{ji}) \lambda_j} \quad (4)$$

While the total amount of computing jobs that need to be handled locally by the user is$(1 - \sum_{i \in A_j} z_{ji}) d_j \lambda_j$.

Transmission Latency: The transmission of data delay is explored and shown in two cases. The first considers the MEC server cache and looks at the transfer of time between the gadget being used and the server's networks following the request. This could be expressed as:

$$S_j^{VN} = \sum_{i \in A_j} f_{VN} \ (z_{ji} d_j + c_j) \quad (5)$$

The back link's data transmission experiences delay when data is sent from a remote data centre over the main network.

$$S_j^{ND} = \sum_{i \in A_j} f_{ND} \left(1 - w_{ji}\right) c_j \quad (6)$$

In the first scenario, denoted by 1 - x. = 0, the MEC can directly reply with the user's query for content requiring a link to its information facility. In conclusion, the sum of execution and transfer delays equals the overall latency created by the users

$$S = \sum_{j=1}^{N} (D_i^V + D_j^V + S_j^{VN} + S_j^{ND}) \quad (7)$$

6. Energy consumption model

This part looks at how much energy the mobile phones on the smart production line need when a lot of industrialized IoT devices communicate actions to the edges of the system, which increases the use of electricity.

Calculated energy consumption: There is a strong relationship between CPU mathematical power and microchip layout, meaning that energy consumption in CPU design iterations is determined. Within the same design, power and rate determine how much energy is used and the quantity of computations determines the amount energy used in individual elements.

$$F_j = D_j * K^2 * e_{vj} * A_j * e_j \quad (8)$$

Where A the processor task's quantity of data is, D is the effective switching capacitance, which K is the voltage, applied to A. e is the number of tasks transmitted and F_j is the present MEC server CPU rate.

Transmission energy consumption: There is a certain power use ratio for every CPU generation and there is a strong relationship between processor computing power and architecture. Electrical consumption in a single design is contingent upon factors such as voltage, frequency and the number of computing tasks executed, hence influencing overall device energy consumption.

$$F_{s,j,i} = O_j^* (S_j^{VN} + S_j^{ND}) \quad (9)$$

The present job must be transferred to the MEC servers for performance if it requires more computing power than the local device can manage. The amount of energy used is determined by equation (10):

$$F_{local} = D_j * K^2 * e_{vj} * A_j * e \quad (10)$$

Where $F_{t,j}$ indicates the calculation of energy consumption, $F_{t,j}$indicates the transmission of the power use of the j^{th} task that was offloaded to the j^{th} MEC server and F_{local} indicates the power usage of the local calculation.

$$F = F_{local} + F_{s,j,i} + F_{t,j} \quad (11)$$

7. Algorithm implementation

7.1. Boosted Beetle Swarm Optimization (BBSO)

Heuristic optimisation techniques like the Beetle Antennae Search (BAS) algorithm are known for their minimal complexity and high efficiency. It travels in a manner similar to that of beetles in the wild, making its way gradually towards food sources. This strategy is similar to optimisation techniques like Genetic Algorithm (GA), Firefly Algorithm (FA), and Particle Swarm Optimisation (PSO). BAS has gained appeal in several optimisation domains by consistently identifying global optimal solutions for complicated issues through iterative motions.

The BBSO approach is thoroughly explained after the introduction of the BAS strategy. BBSO gets around BAS constraints by taking ground risk distribution into account. With M insects, the artificial beetle swarm guarantees variety with adjustable step lengths. For adaptability, each beetle's mobility is enhanced with a proportionate ratio. Each beetle's behavior is intended to be influenced by the lure operator through social indicators.

Adaptive move size for each beetle: The distances that the beetles walk in iteration might be determined by their step sizes, which would result in varying optimization efficiency and accuracy. Equation (10), an adaptive variable as the bug advances, can be used to compute the step size, represented as Step$Rand_j^l$, for every insect j in a

swarm in the k^{th} move. Even though all of the beetles are members of the same swarm, the constant D_j would be different for each insect. $D_j$is set to a positive number that is less than one. Equation (12) can be used to determine an adaptive step size for each insect i in each movement k.

$$F_{local} = D_j * K^2 * e_{vj} * A_j * e \quad (12)$$

Furthermore, the number of iterations and step size are closely connected. It is clear from Equation (10), which could further facilitate the identification of the anticipated optimal solutions, that the step size has decreased significantly in the final little iteration when compared to the initial few repetitions.

Random proportionality coefficient: Equation (13) could be used to compute Q_j^l, a random proportionality coefficient, for a beetle j in the l^{th} motion. For a beetle j, the two constants in this equation, f_2^j and f_1^i, are used to restrict the range of the proportionality coefficient Rik throughout each move. In general, given a beetle j, f_1^i is greater thanf_2^j. Furthermore, for certain beetles, these two factors, f_1^i andf_2^j, can vary. To provide the necessary unpredictability in the ratio coefficient $Rand_j^l$ is employed to produce a random number between 0 and 1.

$$Q_j^l = f_1^i + Rand_j^l \times f_2^j \quad (13)$$

It is possible to discover the connection across the step mass $Rand_j^l$of beetle i in the l^{th} move and the distance Q_j^l between its left and right antenna once a random proportionality coefficient $Rand_j^l$ has been produced. The direct expression of the formula is Q_j^l =$Rand_j^l$.

Optimum attraction operator: The suggested BBSO method uses a swarm of M beetles. More insects are drawn to the existing worldwide excellent answer as the beetle travels. To offset this pull, an ideal want operator is constructed that creates a force that points all beetles in the search space in the direction of the best possible location in the world. Equation (14) represents the ideal attraction operator (AF) that links the world's best-positioned global citizen (GC) with other non-global persons.

$$w_j^l = w_j^{l-1} + B_E.\left(H_D - w_j^{l-1}\right) - Step_j^l . \vec{a} . sign(e(w_K) - e(w_Q)) \quad (14)$$

The precision and efficacy of the suggested BBSO technique are better enhanced than the usual BAS algorithm by integrating the beneficial impacts of a beetle swarm with the unique impact of a beetle.

$$y_3 = \left(\frac{x_2 - x_1}{y_2 - y_1}\right)^2 - y_1 - y_2 \quad (15)$$

$$x_3 = \left(\frac{x_2 - x_1}{y_2 - y_1}\right)(y_1 - y_3) - x_1 \quad (16)$$

Point Doubling (PD) is defined as $2P_1(x_1, y_1) = P_3(x_3, y_3)$,

$$y_3 = \left(\frac{3y_1^2 + b}{2x_1}\right)^2 - 2y_1 \quad (17)$$

$$x_3 = \left(\frac{3y_1^2 + b}{2x_1}\right)(y_1 - y_3) - x_1 \quad (18)$$

Algorithm 1: BBSO Algorithm

Input: *Set up the swarm originally* $X_j(j = 1,2, \dots, n), v, \delta,$
R, *hurry choice*v_{min} *and* u_{max}
Output:x_{best}, e_{best}
Conclude each one substances competence.
While $r < R$)
Conclude the inaction price ω base on
With each explore train
Calculate$e(X_{qt})$*and*$e(X_{lt})$*base on*
Modify an exploit that is iterative ξ *base on*
Make change to the quickening computation U base on
Adjust the search's present position cause base on
End for
Conclude every solo consumer competence $e\ (x)$
Note and stay pathway of each explore consumer address
For *each search cause*
If $f e(x) < e_{obest}$ ***then***
$e_{obest} \leftarrow e(x)$
End if
If $e(x) < e_{hbest}$ ***then***
$e_{hbest} \leftarrow e(x)$
End if
End for

8. Results

The experiments took place using a PC workstation outfitted with an Intel Core i7-7700HQ CPU @ 2.80 GHz processor, 6,144 MB Nvidia GeForce GTX 1060 graphics card and 16,384 MB of RAM. The existing methods such as genetic algorithm (GA) [23], particle swarm optimization (PSO) [23] and simulated annealing algorithm (SA) [23] were used for this study.

The present research assesses three dumping techniques: the GA, SA and PSO-based offload techniques. It is believed that the MEC server and local hardware are capable of handling tasks in the simulated study. A comparison of the overall system costs of various offload techniques under various energy consumption limitations is shown in Figure 2. Ten MEC servers (N) and fifty devices (M) are configured. It can be shown that the three dumping options' overall system costs will drop, as power consumption limitations rise. The cost of SA is less than that of GA until the energy usage requirement hits $600J/ms$, as consumption constraints grow and the framework becomes less susceptible to petrol consumption difficulties. BBSO regularly beats PSO, GA and SA in conditions with little energy demand. The impacts of SA, GA and PSO do not differ much when the energy consumption constraint is around $1000J/ms$. Comparing BBSO to the previous system, the entire cost is reduced by around 12%. PSO's overall cost is 8.9% lesser than GA's total cost and 22.3% lower when the energy consumption restriction is set at $0J/ms$.

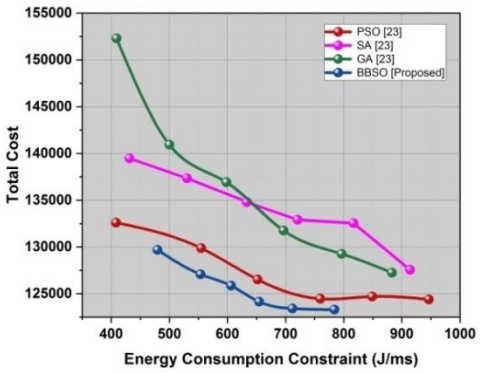

Fig.2. Comparison of Total Cost

The first 40 cycles see a quicker convergence of GA. The global perfect solution may not be found using the SA algorithm, which is a subpar method. It is possible to configure a limitless amount of devices. This study limited the number of devices that might be used in our experiment to 250. Figure 3 displays the comparative findings of the three offloading options' total system costs with varying device counts.

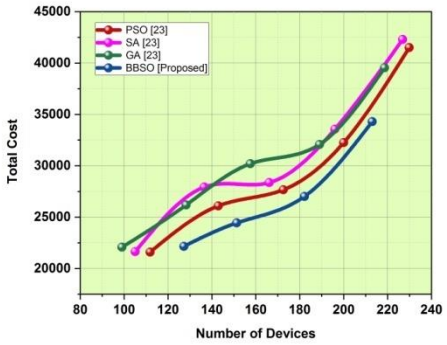

Fig.3. Comparison of average delay

Fig.4. compares the standard delay of three distinct offloading systems under various energy consumption thresholds. Particularly when the limitation is $0J/ms$, BBSO consistently shows much fewer delays than PSO, GA and SA. The average delay for each of the three techniques falls as the energy consumption limitation rises. Interestingly, SA's delay becomes smaller than GA's when the limitation is $850J/ms$. All things considered, BBSO performs around 10% better in terms of average latency at $0\ J/ms$ than GA and SA.

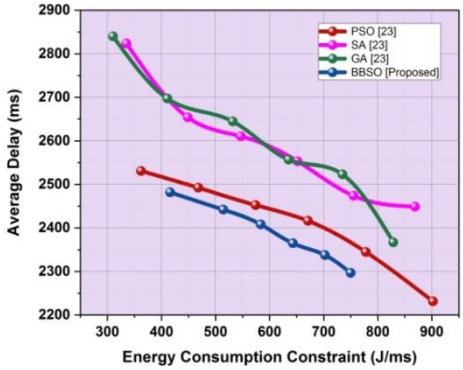

Fig.4. Comparison of Energy Consumption

Execution latency and delay in transmission are the two categories of delay. The performance delays of the three unloading options are compared when the number connected devices varies in Figure 5. The processing delays of BBSO and PSO, GA do not

appear to differ when there are 50 devices. The amount of latency rises with the amount of devices since the total amount of sensors and the volume of data that rises has to be handled. When the total amount of devices is fewer than 200, the time required for delay of the three offloading strategies is the same. When the processing time is considered, BBSO is 30.8% faster than SA and 16.8% faster than GA with 100 devices.

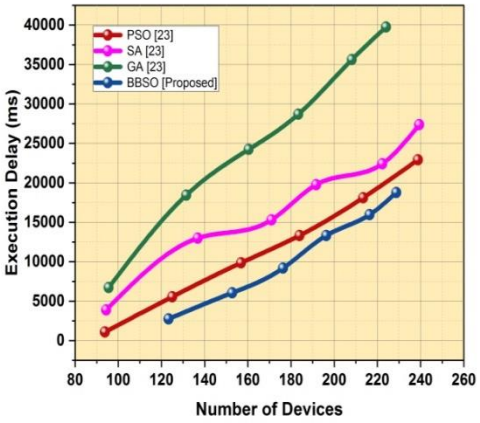

Fig.5. Comparison of Numbers of Devices

9. Conclusion

Energy usage and processing time were balanced by introducing a penalty system to reduce queue delays. In an IoT manufacturing environment, the BBSO technique outperforms GA and SA as job and equipment numbers increase, meeting low-latency demands. However, it could be difficult to manage BBSO settings successfully; it takes experience to maximize results in the face of many obstacles. The research does not examine the service requirements for high dependability; instead, it considers low-latency and low-energy use criteria. According to the analysis of the study's findings, the BBSO approach performs better than the PSO, GA and SA strategies as the number of jobs and equipment increases. This allows it to satisfy the requirements of minimal latency in task processing inside the IoT. Investigate the potential of integrating a machine can be included into the BBSO framework to support ongoing development and adaptive decision-making. This could entail self-optimizing features that adapt settings continuously for greater efficiency based on past experience.

References

[1] Zhao, L., Li, T., Zhang, E., Lin, Y., Wan, S., Hawbani, A., & Guizani, M. (2023). Adaptive Swarm Intelligent Offloading Based on Digital Twin-assisted Prediction in VEC. IEEE Transactions on Mobile Computing.

[2] Pan, I., Abd Elaziz, M., & Bhattacharyya, S. (Eds.). (2020). Swarm intelligence for cloud computing. CRC Press.

[3] Jun, S., Kang, Y., Kim, J., & Kim, C. (2020). Ultra-low-latency services in 5G systems: A perspective from 3GPP standards. ETRI journal, 42(5), 721-733.

[4] Liu, X., Qiu, T., Dai, B., Yang, L., Liu, A., & Wang, J. (2020). Swarm-intelligence-based rendezvous selection via edge computing for mobile sensor networks. IEEE Internet of Things Journal, 7(10), 9471-9480.

[5] Guo, F., Tang, B., Kang, L., & Zhang, L. (2021). Mobile edge server placement based on bionic swarm intelligent optimization algorithm. In Collaborative Computing: Networking, Applications and Worksharing: 16th EAI International Conference, CollaborateCom 2020, Shanghai, China, October 16–18, 2020, Proceedings, Part II 16 (pp. 95-111). Springer International Publishing.

[6] Gupta, S., Singh, P., & Singh, R. M. (2023, May). A firefly based approach for scheduling of tasks for fog networks. In 2023 Third International Conference on Secure Cyber Computing and Communication (ICSCCC) (pp. 356-361). IEEE.

[7] Yang, B., Pang, Z., Wang, S., Mo, F., & Gao, Y. (2022). A coupling optimization method of production scheduling and computation offloading for intelligent workshops with cloud-edge-terminal architecture. Journal of Manufacturing Systems, 65, 421-438.

[8] Kashyap, V., Ahuja, R., & Kumar, A. (2022, November). Nature Inspired Meta-Heuristic Algorithms based Load Balancing in Fog Computing Environment. In 2022 Seventh International Conference on Parallel, Distributed and Grid Computing (PDGC) (pp. 396-401). IEEE.

[9] Wang, C., Yu, W., Lu, J., Zhu, F., Fan, L., & Li, S. (2022). UAV-based physical-layer intelligent technologies for 5G-enabled internet of things: A survey. Wireless

Communications and Mobile Computing, 2022, 1-5.

[10] Li, H., Liu, L., Duan, X., Li, H., Zheng, P., & Tang, L. (2024). Energy-efficient offloading based on hybrid bio-inspired algorithm for edge-cloud integrated computation. Sustainable Computing: Informatics and Systems, 100972.

[11] Malik, P. K., Sharma, R., Singh, R., Gehlot, A., Satapathy, S. C., Alnumay, W. S., ... & Nayak, J. (2021). Industrial Internet of Things and its applications in industry 4.0: State of the art. Computer Communications, 166, 125-139.

[12] Malik, P. K., Sharma, R., Singh, R., Gehlot, A., Satapathy, S. C., Alnumay, W. S., ... & Nayak, J. (2021). Industrial Internet of Things and its applications in industry 4.0: State of the art. Computer Communications, 166, 125-139.

[13] Christou, I. T., Kefalakis, N., Soldatos, J. K., & Despotopoulou, A. M. (2022). End-to-end industrial IoT platform for Quality 4.0 applications. Computers in Industry, 137, 103591.

[14] Arachchige, P. C. M., Bertok, P., Khalil, I., Liu, D., Camtepe, S., & Atiquzzaman, M. (2020). A trustworthy privacy preserving framework for machine learning in industrial IoT systems. IEEE Transactions on Industrial Informatics, 16(9), 6092-6102.

[15] Singh, J., Gimekar, A., & Venkatesan, S. (2023). An efficient lightweight authentication scheme for human-centered industrial Internet of Things. International Journal of Communication Systems, 36(12), e4189.

[16] Tripathi, A. K., Sharma, K., Bala, M., Kumar, A., Menon, V. G., & Bashir, A. K. (2020). A parallel military-dog-based algorithm for clustering big data in cognitive industrial internet of things. IEEE Transactions on Industrial Informatics, 17(3), 2134-2142.

[17] Khang, A., Abdullayev, V., Hahanov, V., & Shah, V. (Eds.). (2024). Advanced IoT Technologies and Applications in the Industry 4.0 Digital Economy. CRC Press.

[18] Sun, W., Lei, S., Wang, L., Liu, Z., & Zhang, Y. (2020). Adaptive federated learning and digital twin for industrial internet of things. IEEE Transactions on Industrial Informatics, 17(8), 5605-5614.

[19] Aheleroff, S., Xu, X., Lu, Y., Aristizabal, M., Velásquez, J. P., Joa, B., & Valencia, Y. (2020). IoT-enabled smart appliances under industry 4.0: A case study. Advanced engineering informatics, 43, 101043.

[20] Sisinni, E., Saifullah, A., Han, S., Jennehag, U., & Gidlund, M. (2018). Industrial internet of things: Challenges, opportunities, and directions. IEEE transactions on industrial informatics, 14(11), 4724-4734.

[21] Khalil, R. A., Saeed, N., Masood, M., Fard, Y. M., Alouini, M. S., & Al-Naffouri, T. Y. (2021). Deep learning in the industrial internet of things: Potentials, challenges, and emerging applications. IEEE Internet of Things Journal, 8(14), 11016-11040.

[22] Liang, W., Tang, M., Long, J., Peng, X., Xu, J., & Li, K. C. (2019). A secure fabric blockchain-based data transmission technique for industrial Internet-of-Things. IEEE Transactions on Industrial Informatics, 15(6), 3582-3592.

[23] You, Q., & Tang, B. (2021). Efficient task offloading using particle swarm optimization algorithm in edge computing for industrial internet of things. Journal of Cloud Computing, 10, 1-11.