mathematics

MDPI

Article

Stable Matching Assisted Resource Allocation in Fog Computing Based IoT Networks

Ahmed S. Alfakeeh [1,*] and Muhammad Awais Javed [2]

[1] Department of Information Systems, King Abdul Aziz University, Jeddah 21589, Saudi Arabia
[2] Department of Electrical and Computer Engineering, COMSATS University Islamabad, Islamabad 45550, Pakistan; awais.javed@comsats.edu.pk
* Correspondence: asalfakeeh@kau.edu.sa

Abstract: Future Internet of Things (IoT) will be a connected network of sensors enabling applications such as industrial automation and autonomous driving. To manage such a large number of applications, efficient computing techniques using fog nodes will be required. A major challenge in such IoT networks is to manage the resource allocation of fog computing nodes considering security and system efficiency. A secure selection of fog nodes will be needed for forwarding the tasks without interception by the eavesdropper and minimizing the task delay. However, challenges such as the secure selection of fog nodes for forwarding the tasks without interception by the eavesdropper and minimizing the task delay are critical in IoT-based fog computing. In this paper, an efficient technique is proposed that solves the formulated problem of allocation of the tasks to the fog node resources using a stable matching algorithm. The proposed technique develops preference profiles for both IoT and fog nodes based on factors such as delay and secrecy rate. Finally, Gale–Shapley matching is used for task offloading. Detailed simulation results show that the performance of the proposed technique is significantly higher than the recent techniques in the literature.

Keywords: Internet of Things; resource allocation; task offloading; security

MSC: 94C15; 05D15; 91B68

Citation: Alfakeeh, A.S.; Javed, M.A. Stable Matching Assisted Resource Allocation in Fog Computing-Based IoT Networks. *Mathematics* **2023**, *11*, 3798. https://doi.org/10.3390/math11173798

Academic Editors: Sujit Biswas and Md. Shirajum Munir

Received: 8 July 2023
Revised: 20 August 2023
Accepted: 2 September 2023
Published: 4 September 2023

1. Introduction

The Internet of Things (IoT) connects physical devices with the Internet and enables applications in the areas of telecommunications, industrial automation, agriculture, and health care [1–8]. Fog computing is one of the basic requirements of next-generation IoT applications which provides computing capabilities near to the end devices. As IoT involves monitoring a large amount of data and developing insights, efficient data storage and task computation techniques in fog computing are needed [9,10].

IoT nodes send their application-related tasks to different fog nodes which execute them on their behalf. The result is then sent back to the IoT nodes or a centralized decision-making server. Task offloading in IoT faces challenges such as security and efficient resource allocation. The task offloading transmission from the IoT nodes to the fog servers is susceptible to many attacks such as eavesdropping, jamming, data integrity, etc. [11–13]. A secure offloading approach considers security of the links as an important criteria to maximize the security of transmissions [14–18]. This is achieved by using techniques such as traditional cryptography and physical layer security (PLS) [19,20]. Similarly, resource allocation of fog nodes is a vital part of the task offloading process [21]. It is important to use the best fog computing nodes such that the system efficiency is maximized. This means maximizing energy efficiency and minimizing the task computational delay.

In this paper, an efficient task offloading technique is developed that minimizes the task delay and maximizes the secrecy rate. The problem is formulated as an optimization problem which is an NP-hard to solve problem. The problem is solved by using a graph

theory-based stable matching algorithm. IoT and fog node preference profiles are developed by using factors such as secrecy rate and energy. The Gale–Shapley stable matching algorithm is used for one-one matching of tasks with the fog node computational free spaces. An IoT and fog computing simulation model is developed in MATLABand the simulation results are presented. Detailed performance analysis clearly indicates that the technique proposed for offloading has significantly improved results in comparison to other related techniques in terms of computational delay and secrecy rate.

The paper organization is given in the following. In Section 2, the literature review is presented. In Section 3, the system model and problem formulation are presented. In Section 4, the proposed task offloading technique is discussed. In Section 5, the performance of the proposed work is presented. In Section 6, future challenges and opportunities are presented. In Section 7, the conclusions are presented.

2. Related Works

The concept of task offloading has been investigated in many areas of wireless communications such as IoT, vehicular networks, and mobile networks. A brief review of recent work in this area is presented in Table 1.

Table 1. Literature review.

Network	Main Idea	Technique Used	Results
Vehicular [22]	Use of mobility dynamic connectivity	Maximum bi-partite matching Kuhn–Munkras algorithm	Task response time
Mobile [23]	Minimize energy consumption Minimize monetary cost	Distributed matching Preference based on task size and revenue	Energy consumption Monetary cost
IoT [24]	Minimize task latency	Parallel offloading many to one matching	Task latency Resource utilization
IoT [25]	Minimize energy consumption Reduce task outages	Task deadline One to many matching Ranking based on multiple criteria	Energy consumption Task outages
Vehicular [26]	Quick task offloading Task priorities	Knapsack algorithm Priority based execution	Task delay
IoT [27]	Task latency reduction	Best preference selection Matching algorithm	Task latency

2.1. Literature Review

In [22], a task offloading technique for vehicular network is proposed that considers vehicle mobility and dynamic network connectivity to improve the task response time. The task offloading problem is transformed into a bi-partite graph and a stable matching algorithm is used. The Kuhn–Munkras (KM) algorithm is used to find the maximum stable matching for the considered scenario. The work in [23] considers a mobile edge computing scenario intending to minimize energy consumption and monetary cost. The joint optimization problem is converted into a graph theory problem for which a distributed matching algorithm is proposed. The preferences of computing nodes are set based on the task size and task computing revenue.

In [24], an IoT network is considered with the parallel offloading scenario. The goal is to divide the tasks into multiple subtasks and transmit them to different fog nodes. A many-to-one matching scheme is used to maximize resource utilization and reduction of task delay. Moreover, the externalities problem is also solved to handle the dynamic preference profiles. The results show improved task latency and resource utilization. The work in [25] considers an IoT scenario with partial offloading of tasks to the computing resources. The objective is to minimize the energy consumption of the nodes and also reduce the number of unserved tasks by the fog nodes (task outages). The proposed

algorithm uses task deadline and one-to-many matching to solve the above problem. For preferences, a ranking is calculated based on multiple criteria.

A vehicular-network-based task offloading scenario is presented in [26]. The goal is to achieve quick task offloading in a priority-based traffic scenario. A knapsack algorithm is used to maximize the efficiency of computational resource allocation with high-priority tasks. The results show that the computational delay for high-priority tasks is increased whereas low-priority tasks are executed before their deadlines. The work in [27] considers an IoT scenario with the goal of reducing task latency. An optimal solution using stable matching of tasks to the computational resource at fog node is presented. The preference of both sides considers the task computational delay. The results highlight the improvement in task latency as compared to other techniques.

2.2. Novelty of the Proposed Work

In comparison to the previous work, the proposed technique considers security as well as task latency for task offloading. Most of the previous techniques do not consider the impact of eavesdroppers while offloading the tasks. While higher task latency may be achieved by offloading the tasks to the high data rate links and fog nodes with free computational space, it may not be a fully secure approach and may compromise task privacy. The proposed technique considers this important aspect which is missing in the current literature and jointly optimizes secrecy rate and total task delay.

3. System Model

This paper considers a scenario where IoT nodes are placed in an industrial environment for data sensing and transmitting. Moreover, IoT nodes are also performing many application-related tasks and their computational capacity is not sufficient for timely computation of these tasks. Thus, fog computing nodes are installed in the industrial setup to efficiently compute the tasks received from the IoT nodes.

As shown in Figure 1, the considered scenario is presented. The fog nodes have several free computational resources which are sufficient to handle many incoming tasks at the same time. However, all tasks cannot be handled by a single fog node, hence requiring the intelligent offloading of tasks to the appropriate computing resources at fog nodes.

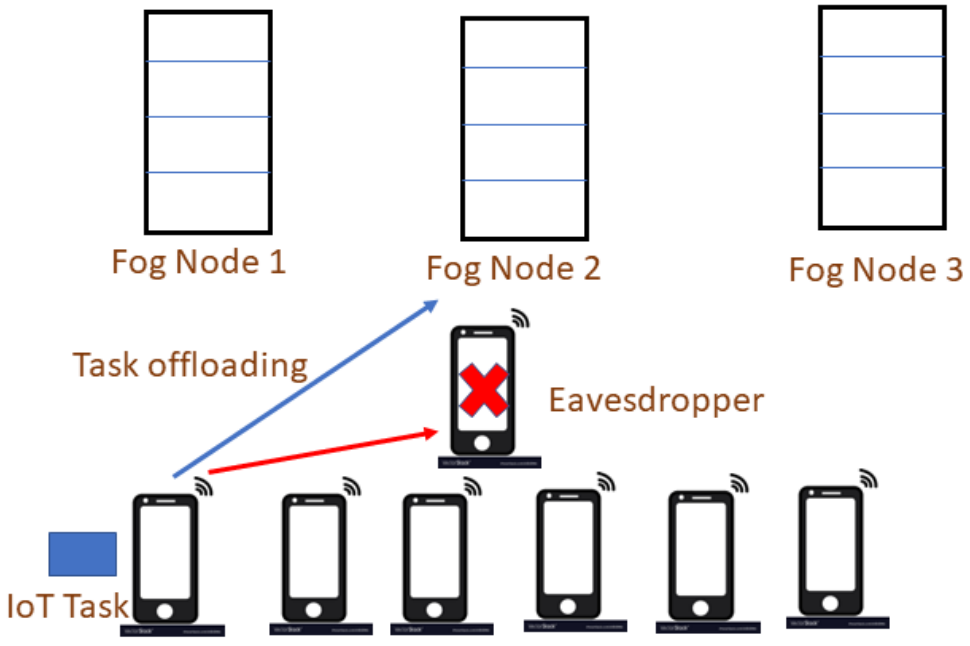

Figure 1. Considered System Model.

In this paper, eavesdroppers are also considered in the network, which are malicious nodes with a goal to intercept the signal and decode the data. These malicious nodes can also launch several other networking attacks; however, the main focus of this paper is on eavesdropping.

To compute the overall delay taken for task computation, several factors are considered. The list of notations used in this paper is given in Table 2 Let us consider a task of size S_t. It is assumed that the fog node's capacity for computational is C_f in bits/s. The total task delay T_t can be given as follows:

$$T_t = T_{tran} + T_{comp} + T_q \tag{1}$$

Table 2. List of notations used in the paper.

Symbol	Meaning
S_t	Size of task
$\mathbb{T}$	Set of tasks
$\mathbb{F}$	Set of fog node resources
C_f	Computational capacity of fog nodes in bits/s
T_t	Total task delay
T_{tran}	Transmission delay
T_{comp}	Computational delay
T_q	Queuing delay
D_r	Data rate
p_i	Transmit power of IoT node i
$h_{i,j}$	Channel gain for IoT node i, Fog node j link
B	Channel bandwidth
N_0	Noise power
C_c	Computational speed of fog node in cycles/s
B_c	Computational speed of fog node in bits/cycle
N	Number of tasks computed in parallel by fog node
N_q	Number of the given task in the queue at the fog node
E_i	Energy consumed by the IoT node i
E_f	Energy consumed by the Fog node f
$R_s^{i,j}$	Secrecy rate for IoT node i and fog node j link in presence of eavesdropper
Q	Quota of fog node
$a_{i,j}$	Task offloading vector

Here, T_{tran}, T_{comp}, and T_q are the transmission, computational, and queuing delays, respectively. It is assumed that the data rate between the IoT node I_i and the fog nodes F_j is given as $D_r^{i,j}$. Then, the transmission delay $T_{tran}^{i,j}$ for a task offloaded from IoT node i to fog node j can be given as:

$$T_{tran}^{i,j} = \frac{S_t^{i,j}}{D_r^{i,j}} \tag{2}$$

The data rate for link i, j can be given as follows:

$$D_r^{i,j} = B \times log_2(1 + \frac{p_i \times h^{i,j}}{N_0}) \tag{3}$$

Here, p_i represents the power of the data transmitted by the IoT node, $h^{i,j}$ represents the channel gain, and N_0 represents the noise. The channel gain depends on the channel conditions on the IoT node-fog node link. It takes into account path loss as well as multi-path fading.

The computation delay t_{comp} is given as follows:

$$t_{comp} = \frac{S_t}{C_f} \tag{4}$$

Here, C_f is the computational capacity of fog nodes in bits/s and can be given as follows:

$$C_f = \frac{C_c}{B_c} \tag{5}$$

where C_c is the speed of computation available when using fog nodes and is given in the number of cycles/s. B_c is the number of bits/cycle that can be computed by the fog node. The ratio of these two parameters provides a measure of how many tasks can be computed in one second by the fog node. Equation (4) gives the total time required to compute the task.

Since the fog node can only compute a limited number of tasks in parallel depending on its computational speed, let N be the number of tasks that can be computed in parallel by the fog node and N_q be the number of tasks that are ahead of the given task at the fog node's queue; then, the queuing delay t_q is given as follows:

$$t_q = \lfloor \frac{N_q}{N} \rfloor \times t_{comp} \tag{6}$$

Using the above equation, the waiting time of a task in the fog node's queue can be given depending on the number of tasks that are ahead of the given task in the queue. The fog node resources are divided into N number of virtual resource units for parallel computation. The maximum parallel computation units by the fog node are called its quota Q.

3.1. Energy Consumption Model

The energy consumed by the IoT nodes and fog nodes depends on the energy consumption in the transmission of tasks, reception of the tasks, and computation of the tasks [25]. Since the IoT node is not computing any task, its energy is given as follows:

$$E_i = p_i \times (t_{tran} + t_{rec}) \tag{7}$$

Here, p_i is the transmission power of the IoT nodes, t_{tran} and t_{rec} are the delay required during the transmission of task and reception of the tasks.

Similarly, for the fog nodes, the energy consumption is given as follows:

$$E_j = p_j \times (t_{tran} + t_{rec} + t_{comp}) \tag{8}$$

Here, t_{comp} is the computational delay of the task. The fog node energy is dependent on the size of the task and the computational capacity of the fog node.

3.2. Secrecy Rate

The secrecy rate for IoT node i for transmission to fog node j is defined as follows:

$$R_s^{i,j} = D_r^{i,j} - D_r^{i,e} \tag{9}$$

Here, $D_r^{i,j}$ is the data rate of the link between IoT node i and fog node j whereas $D_r^{i,e}$ is the data rate of the main link, the link between IoT node i and eavesdropper e. It is to be

noted that task transmission time t_{tran} is considered as part of (13). The higher the secrecy rate for a link i, j, the higher the security of the transmission.

3.3. Problem Formulation

The problem addressed in this work is related to maximizing the security of the transmitted task against eavesdroppers and minimizing the time required to compute the task. The formulated problem can be presented as follows:

$$\max \quad (R_s) \quad \text{and} \quad \min \quad (t_{comp})$$

$$C1 \quad \sum_{f=1}^{F} a_{i,j} = 1 \tag{10}$$

$$C2 \quad \sum_{j=1}^{N} a_j \leq Q \tag{11}$$

$$C3 \quad \forall \mathbb{T} \quad R_s^{i,j} \geq 0 \tag{12}$$

In the above problem, there are three constraints. Constraint number 1 means that a task can only be offloaded to one fog node, i.e., the sum of task offloading vector $a_{i,j}$ overall fog nodes is 1. Constraint number 2 means that a fog node can only accept tasks that are less in number than its quota Q. Lastly, constraint number 3 means that the secrecy rate for all offloaded tasks should be greater than 0.

4. Proposed Task Offloading Technique

The key idea of the proposed task offloading technique is to allocate resources based on factors that enhance the level of security and improve the computation of the tasks. The proposed technique uses a graph theory-based stable matching technique to solve the formulated problem.

4.1. Stable Matching Game

The formulated problem is converted into a stable matching game as follows:

Definition 1. *Let there be two sets of agents, one for the tasks $\mathbb{T}$ and the other for the fog node resources $\mathbb{F}$. A matching game $G(\mathbb{T}, \mathbb{F})$ is defined that associates two agents on each side with each other in a stable manner.*

Definition 2. *Each agent maintains a preference profile containing its preference towards all agents of the other side. Let $\succ_{f_i}$ be the preference of task node i towards all fog nodes in the set $\mathbb{F}$). Let $\succ_{i_j}$ be the preference of fog node j towards all IoT tasks $\mathbb{T}$).*

Definition 3. *A matching is stable if it does not contain any blocking pair of agents. The blocking pair means that the matching game does not pair two agents who prefer each other to be matched but are not currently matched. In the case of blocking pair existence, a stable matching can not be achieved.*

Definition 4. *A Quota Q is defined as the number of tasks that can be accepted by the fog node such that its computational capacity is not exceeded. During the matching process, a fog node cannot accept more tasks than its Q.*

4.2. Fog Node Selection

As there are a limited number of fog nodes with finite computational capacity, and a large number of IoT nodes with different tasks, task allocation becomes a cumbersome process. A fog node can only accept a limited number of tasks. It is critical to note that an IoT task cannot always be offloaded to its preferable fog node. There exists a contention

between tasks and available computational capacity available on the fog nodes. Thus, selecting a fog node for a particular IoT task is a challenging task.

To solve the above problem, we utilize the Gale–Shapley stable matching algorithm to resolve contention between fog node resources and IoT tasks. The stable matching algorithm has many advantages. It can take into account the preferences of both sides for resource allocation, i.e., IoT nodes and fog nodes. Moreover, it provides a stable mapping of resources to the tasks. As a result, each agent gets the best possible resources allocated to it.

A bi-partite graph is developed between the fog nodes and IoT tasks as shown in Figure 2. On the left side of the graph, it shows the IoT tasks and on the right side of the graph, fog nodes are presented. Each IoT node and fog node has a preference set for each other. An edge exists between an IoT node and a fog node if both exist in the preference set of each other. The weight of the edge depends on the preference formulation.

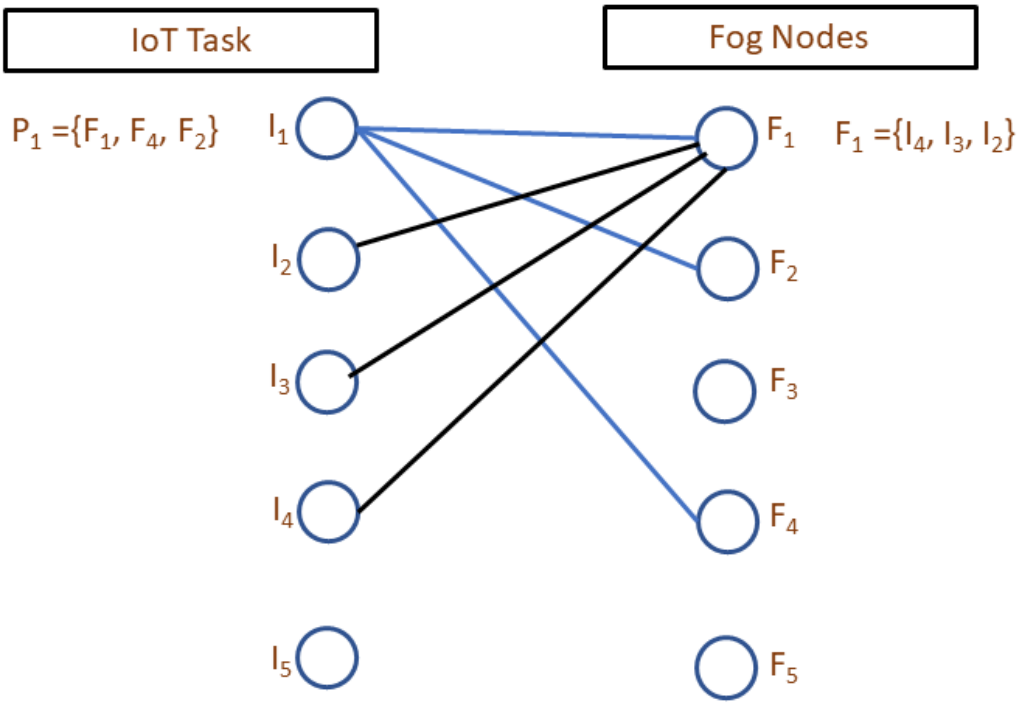

Figure 2. Proposed Matching-based Task Offloading (Here P_1: Preference profile of IoT Task I_1, F_1: Preference profile of Fog Node F_1.

4.3. Preference Profile

The preference set of both IoT nodes and the fog nodes is defined as shown in Table 3. For the IoT nodes, a utility function is proposed that takes into account the secrecy rate and task transmission time. The secrecy rate for IoT node i for transmission to fog node j is defined as follows:

$$R_s^{i,j} = D_r^{i,j} - D_r^{i,e} \tag{13}$$

Table 3. Preference profile of IoT nodes and Fog nodes.

Node	Preference Profile Factors	Reason
IoT	Data rate between IoT node and Fog node Secrecy rate of IoT and eavesdropper link	Reduce transmission delay Increase security of transmission
Fog	Size of Task	Reduce fog node energy

Here, $D_r^{i,j}$ is the data rate of the link between IoT node i and fog node j whereas $D_r^{i,e}$ is the data rate of the main link, the link between IoT node i and eavesdropper e. It is to be noted that task transmission time t_{tran} is considered as part of (13). The higher the secrecy

rate for a link i, j, the higher the security of the transmission. This indicates that the data rate of the main link is much higher than the link of the eavesdropper. Hence, the IoT nodes assign higher preference to the fog nodes with higher $R_s^{i,j}$.

For the fog node, the preference profile considers the IoT task sizes and prioritizes tasks with a lower task size. This is done to conserve the energy of the fog nodes. Hence, the fog nodes sort the tasks in terms of their sizes for preference development.

4.4. Task Offloading Algorithm

The task offloading algorithm is shown in Algorithm 1. Initially, a centralized controller generates a preference for each node in the set $\mathbb{T}$ and $\mathbb{F}$. As discussed in Section 4.3, preferences are generated based on factors highlighted in Table 3.

Algorithm 1: Proposed Task Offloading Algorithm

Preference Generation
Generate a preference list of all IoT tasks for each fog node using equation:
$R_s^{i,j} = D_r^{i,j} - D_r^{i,e}$
Generate a preference list of all fog nodes for each IoT task using task sizes S_t
Stable Matching Algorithm
Initially assign each IoT task i in set $\mathbb{T}$ and fog node resource f in set $\mathbb{F}$ to be free
while *any IoT task in set* $\mathbb{T}$ *is free* **do**
    Let f the top fog node in the preference list of i whom it has not proposed
    if f *is not engaged with any other node* **then**
        allocate i to be matched with f
    end
    else
        if f *has* i *at a higher preference order than its current allocation* i' **then**
            allocate i to be matched with f
            Assign i' to be not engaged to any node
        end
        else
            f does not accept the proposal of i
        end
    end
end
Task Offloading
Record the task allocation vector $a_{i,j}$ based on the stable matching algorithm
Transmit the IoT tasks i to assigned fog node f as per $a_{i,j}$

The second phase of the algorithm is to apply the stable matching algorithm to solve the formulated problem. The algorithm initially assigns each IoT task and fog node resource to be free. The algorithm runs until each task has been assigned a stable fog node resource match. For each free task, the algorithm proposes the top fog node in its preference list for a possible match. If the fog node is not currently allocated, the match is completed.

In case the fog node is already matched with a task, the preference of incoming and already matched tasks for the fog node is checked. If the incoming task has a higher preference, then a new match is established. Otherwise, the older match is retained. The output of the matching algorithm is a stable task allocation vector $a_{i,j}$ which allocates tasks to the fog node resources.

Finally, the tasks are transmitted one by one to the allocated resources. It is to be noted that the task incurs transmission delay to reach the fog node as well as computational delay once it is allocated the fog node computational resources.

5. Performance Evaluation

This section covers the performance of the task offloading technique proposed in this paper. The work is compared with two other recent techniques, namely, the Kuhn–Munkras matching (KMM) algorithm [22], and offloading-matching (Off-Mat) algorithm [23].

5.1. Simulation Model

The proposed technique is implemented in MATLAB and a complete IoT-based task offloading scenario is developed. The simulation parameters are given in Table 4. The total number of fog nodes taken is 5 and the number of IoT nodes varies from 100–300. A single task is generated by each IoT node that needs to be offloaded to one of the fog nodes. It is assumed that the IoT node does not compute tasks by itself. The task size is taken as 3–5 MB.

Table 4. Parameters used in the simulations.

Parameter	Value
Number of Fog nodes	5
Number of IoT nodes	100–300
Task generated per IoT node	1
Task Size	3–5 MB
Computational speed of fog node	6 GHz
Link Bandwidth	10 MHz
Transmit power of IoT nodes	0.5 W
Computational power of fog nodes	0.3–0.5 W
Quota of fog nodes	20–50

The computational speed of the fog nodes is taken as 6 GHz. The link bandwidth is taken as 10 MHz. The transmit power of IoT nodes is taken as 0.5 W. The computational power of fog nodes is taken as 0.3–0.5 W. The quota of fog nodes is taken as 20–50.

5.2. Results

The simulation results plot metrics related to task offloading and are defined as follows:

- **Total Delay:** This is the time taken for the task to be transmitted to the fog node and computed at the fog node, and any queuing delay faced by the task.
- **Secrecy Rate:** This is the difference between the data rates of the IoT-fog node-link and IoT-eavesdropper link.
- **Percentage of Security Outages:** This is the percentage of tasks for which the data rate of the IoT-eavesdropper link is higher than the IoT-fog node link. As a result, the eavesdropper can capture the packet and decode it.
- **Resource Utilization:** This is the computational capacity of the fog node that is used for task offloading. Some computational capacity may remain unmatched as a result of the matching process.
- **Total Energy Consumption:** This is the energy consumed for the transmission and computation of the task.

In Figure 3, the results of the total delay are shown against the number of IoT tasks. It can be seen that the efficient matching process of the proposed technique results in a lower delay as compared to the two other techniques. For example, as the total number of IoT tasks reaches 200, the delay of the proposed technique is 18 ms. This is much lower than the delay incurred by the KMM algorithm and the Off-Mat technique. While KMM experiences a delay of around 24 ms, the Off-Mat technique has a delay of around 27 ms.

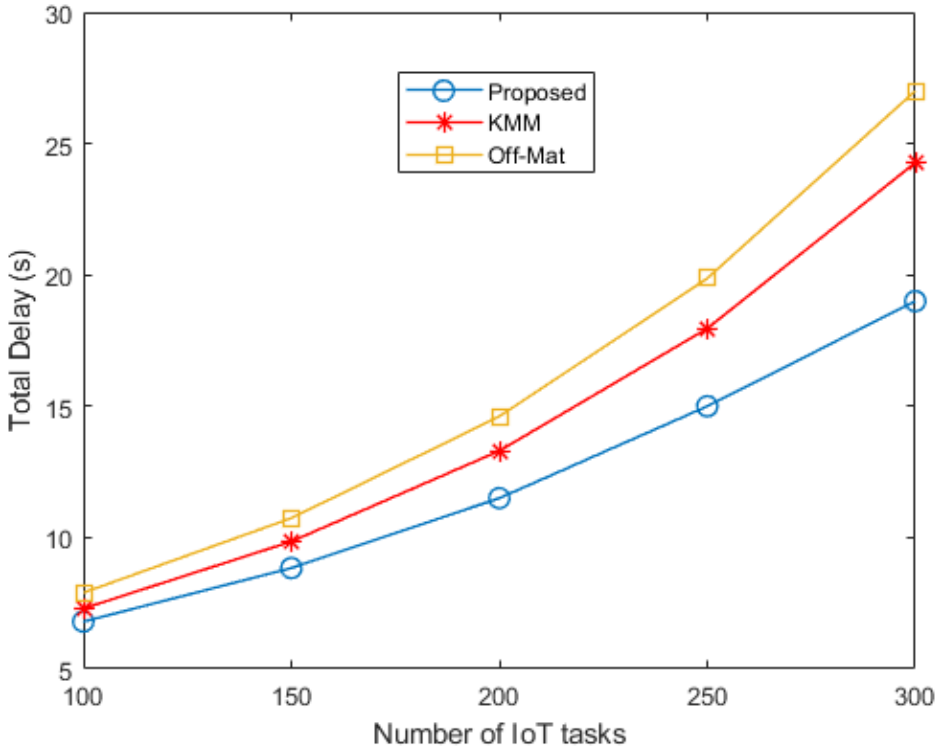

Figure 3. Total delay of tasks vs. number of IoT tasks.

The secrecy rate of all techniques is shown in Figure 4. It can be seen that the proposed technique has a 2.8–5 bps/Hz higher secrecy rate in comparison to the KMM and Off-Mat techniques. The higher secrecy rate of the proposed technique is due to considering security as part of the preference profiles and using it for secure task offloading.

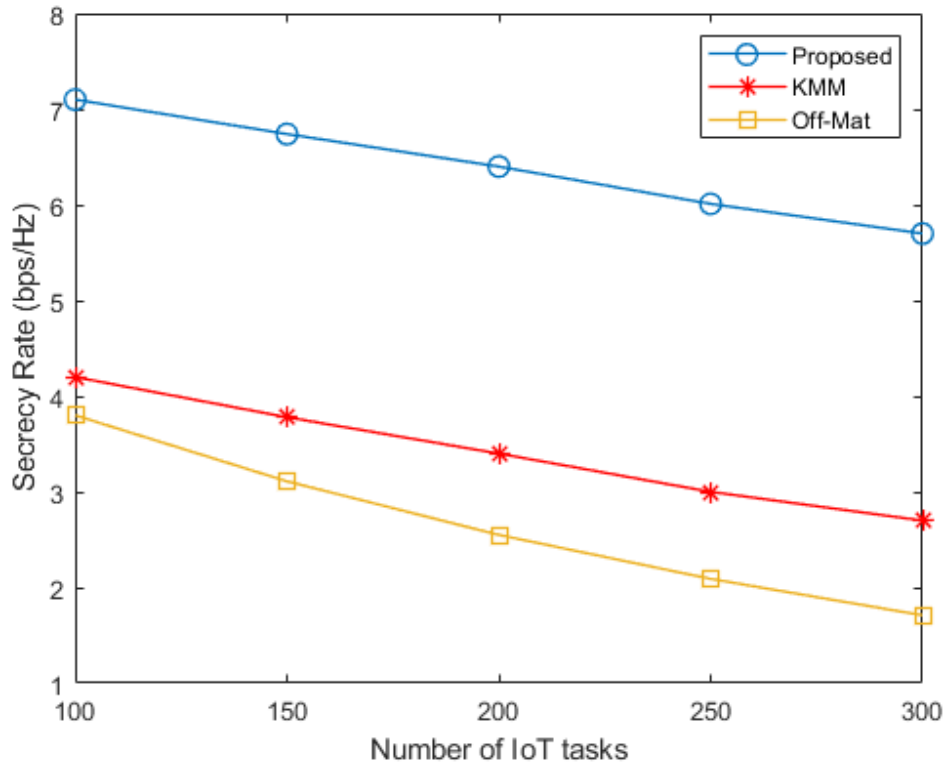

Figure 4. Secrecy rate vs. number of IoT tasks.

In Figure 5, the percentage of security outages is shown. This parameter is very critical as it is an indicator of tasks that eavesdroppers can intercept. It can be seen that the proposed technique shows very few security outages as compared to the other techniques. This is due to the consideration of the secrecy rate while developing the task offloading algorithm. For the KMM and the Off-Mat algorithm, the percentage of security outages goes up to 10–12% when the number of tasks is 300. At this point, the proposed technique only has security outages of less than 1%.

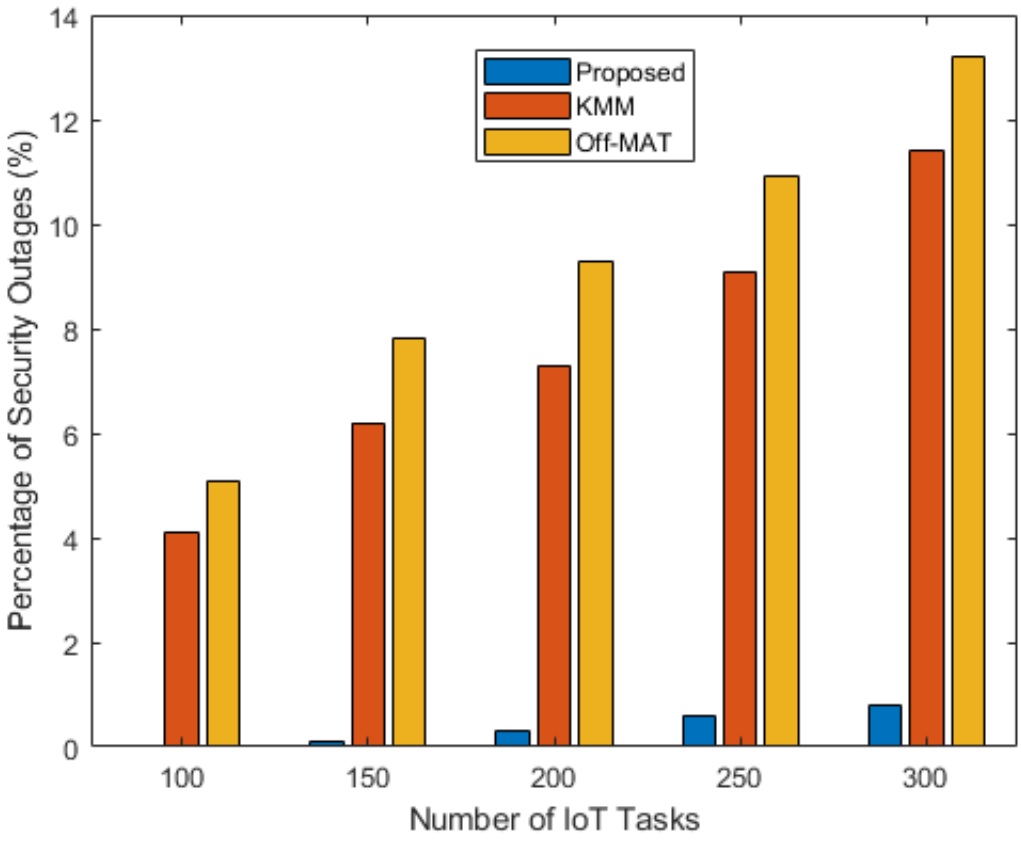

Figure 5. Percentage of security outages vs. number of IoT tasks.

The plot in Figure 6 shows the resource utilization of the fog nodes. It can be seen that the proposed technique shows more than 90% resource utilization for all the scenarios. As compared to the proposed technique, the two other techniques achieve 2–10% less resource utilization at the different numbers of IoT tasks.

The graph in Figure 7 shows the energy consumed when using the proposed technique for the different numbers of IoT tasks. Since the proposed technique has a higher resource utilization, it experiences slightly higher energy consumption too. Of all the techniques, Off-Mat has lower energy consumption due to the lowest resource utilization. On the other hand, KMM has slightly higher energy consumption as it is designed for vehicular networks and does not select fog nodes based on energy criteria.

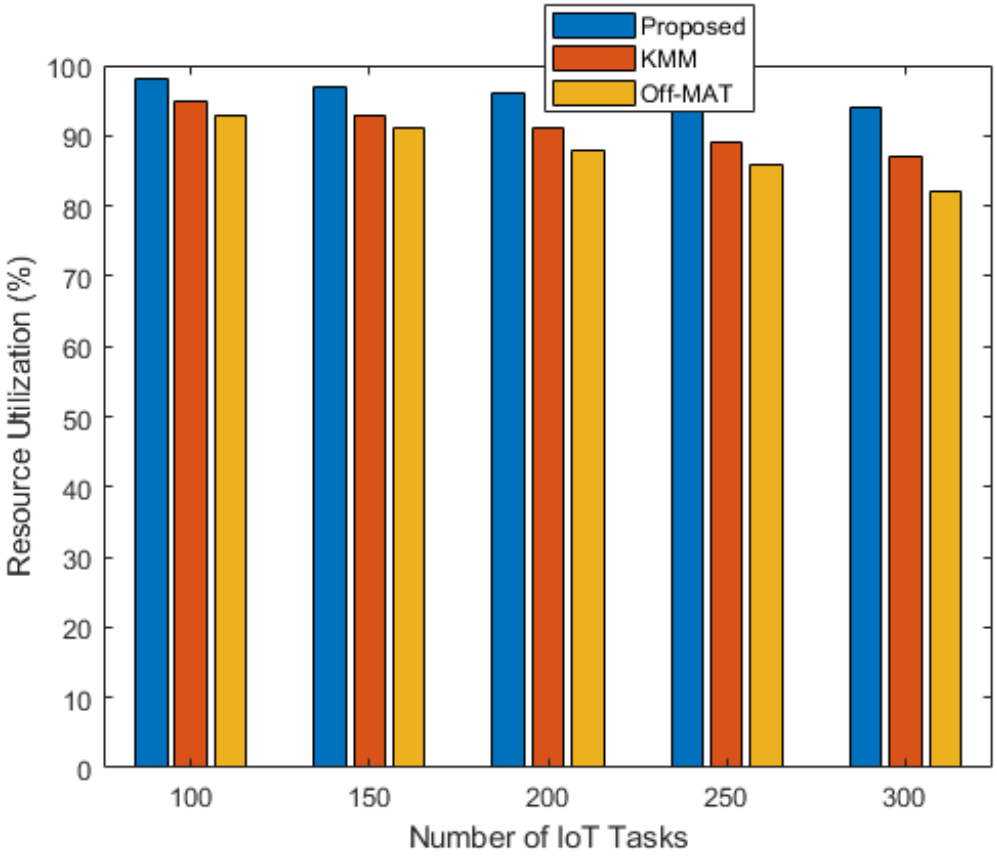

Figure 6. Resource utilization vs. number of IoT tasks.

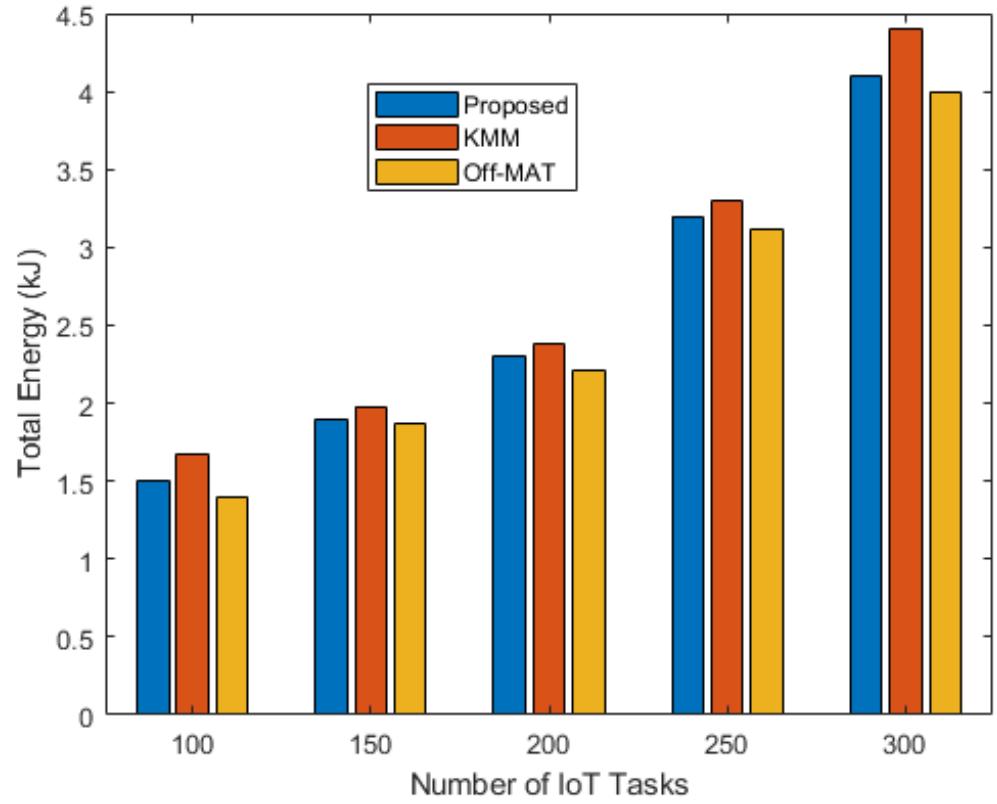

Figure 7. Total energy consumption vs. number of IoT tasks.

5.3. Discussion

From the results, it can be seen that the proposed technique has a better delay and security performance as compared to the two other algorithms. This is because of the efficient preference profile design and matching algorithm of the proposed technique. As it considers delay and secrecy rate metrics while developing the matching algorithm, both of these key metrics are improved. The proposed technique also keeps energy consumption to a satisfactory level, with a slight increase as compared to the two other algorithms. The reason for slightly higher energy consumption is that the proposed technique utilizes fog node resources more efficiently. Since higher computational resources are utilized which would have been otherwise wasted, energy consumption is also increased.

6. Future Opportunities

Some future opportunities related to future task offloading algorithms are highlighted in this section.

6.1. Transmit Power Optimization

The transmit power of IoT nodes is an important metric that can impact the data rate and secrecy rate of the IoT, eavesdropper, and fog node links. A higher transmit power can increase energy consumption; however, it can also improve task security by improving the data rate between the desired source and destination. Hence, optimal transmit power selection is critical for secure task offloading applications and future work is needed in this regard.

6.2. Physical Layer Security with Jamming

Jamming is one potential technique in PLS that can reduce the amount of signal received by the eavesdropper. Intelligent jamming techniques can substantially improve the physical layer security by reducing the interception capability of the eavesdropper. It is important to select which nodes will send the jamming signals and the transmit power of the jamming nodes.

6.3. Channel Estimation

Channel estimation of the IoT-fog node-link and IoT-eavesdropper link is critical for the design of PLS. In scenarios where accurate Channel State Information (CSI) is not available, it is important to develop accurate estimation techniques. As the channel estimation dictates factors such as transmit power, jamming, etc., it is important to have an accurate CSI. Future work is also needed to quantify the security outage percentage when CSI is not accurate.

7. Conclusions

In this paper, a secure task offloading technique for IoT nodes is presented. The proposed technique has two objectives, minimization of the task delay and maximization of the secrecy rate. The formulated problem is converted to a stable matching graph theory problem and the Gale–Shapley matching algorithm is used to solve it. The preferences for IoT and fog nodes are developed and a stable allocation of tasks to the fog nodes is recorded. Simulation results indicate that the technique developed in the paper improves the delay by 33% and secrecy rate of the tasks by 200% as compared to other works in the literature. In the future, we aim to consider different types of security attacks and their impact on task offloading.

Author Contributions: Conceptualization, A.S.A. and M.A.J.; Writing—original draft, A.S.A. and M.A.J.; Writing—review and editing, A.S.A. and M.A.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research work was funded by Institutional Fund Projects under grant no. (IFPIP: 911-611-1443). Therefore, the authors gratefully acknowledge technical and financial support from the Ministry of Education and King Abdulaziz University, DSR, Jeddah, Saudi Arabia.

Data Availability Statement: Data is available from the authors on request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhao, D.; Liu, C.; Xu, G.; Ding, Z.; Peng, H.; Yu, J.; Han, J. A security enhancement model based on switching edge strategy in interdependent heterogeneous cyber-physical systems. *China Commun.* **2022**, *19*, 158–173. [CrossRef]
2. Ali, R.; Zikria, Y.B.; Garg, S.; Bashir, A.K.; Obaidat, M.S.; Kim, H.S. A Federated Reinforcement Learning Framework for Incumbent Technologies in Beyond 5G Networks. *IEEE Netw.* **2021**, *35*, 152–159. [CrossRef]
3. Islam, M.Z.; Ali, R.; Haider, A.; Kim, H.S. QoS Provisioning: Key Drivers and Enablers Toward the Tactile Internet in Beyond 5G Era. *IEEE Access* **2022**, *10*, 85720–85754. [CrossRef]
4. Javed, M.A.; Nguyen, T.N.; Mirza, J.; Ahmed, J.; Ali, B. Reliable Communications for Cybertwin driven 6G IoVs using Intelligent Reflecting Surfaces. *IEEE Trans. Ind. Inform.* **2022**, *18*, 7454–7462. [CrossRef]
5. Jouhari, M.; Saeed, N.; Alouini, M.S.; Amhoud, E.M. A Survey on Scalable LoRaWAN for Massive IoT: Recent Advances, Potentials, and Challenges. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 1841–1876. [CrossRef]
6. Fizza, K.; Banerjee, A.; Jayaraman, P.P.; Auluck, N.; Ranjan, R.; Mitra, K.; Georgakopoulos, D. A Survey on Evaluating the Quality of Autonomic Internet of Things Applications. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 567–590. [CrossRef]
7. Janssen, T.; Koppert, A.; Berkvens, R.; Weyn, M. A Survey on IoT Positioning Leveraging LPWAN, GNSS, and LEO-PNT. *IEEE Internet Things J.* **2023**, *10*, 11135–11159. [CrossRef]
8. Baker, S.; Xiang, W. Artificial Intelligence of Things for Smarter Healthcare: A Survey of Advancements, Challenges, and Opportunities. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 1261–1293. [CrossRef]
9. Zeng, F.; Tang, J.; Liu, C.; Deng, X.; Li, W. Task-Offloading Strategy Based on Performance Prediction in Vehicular Edge Computing. *Mathematics* **2022**, *10*, 1010. [CrossRef]
10. AlShathri, S.I.; Chelloug, S.A.; Hassan, D.S.M. Parallel Meta-Heuristics for Solving Dynamic Offloading in Fog Computing. *Mathematics* **2022**, *10*, 1258. [CrossRef]
11. Zhong, X.; Fan, C.; Zhou, S. Eavesdropping area for evaluating the security of wireless communications. *China Commun.* **2022**, *19*, 145–157. [CrossRef]
12. Pirayesh, H.; Zeng, H. Jamming Attacks and Anti-Jamming Strategies in Wireless Networks: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 767–809. [CrossRef]
13. Ahanger, T.A.; Tariq, U.; Ibrahim, A.; Ullah, I.; Bouteraa, Y.; Gebali, F. Securing IoT-Empowered Fog Computing Systems: Machine Learning Perspective. *Mathematics* **2022**, *10*, 1298. [CrossRef]
14. Li, G.; Lai, C.; Lu, R.; Zheng, D. SecCDV: A Security Reference Architecture for Cybertwin-Driven 6G V2X. *IEEE Trans. Veh. Technol.* **2022**, *71*, 4535–4550. [CrossRef]
15. Javed, M.A.; Zeadally, S.; Hamid, Z. Trust-based security adaptation mechanism for Vehicular Sensor Networks. *Comput. Netw.* **2018**, *137*, 27–36. [CrossRef]
16. Dhelim, S.; Aung, N.; Kechadi, M.T.; Ning, H.; Chen, L.; Lakas, A. Trust2Vec: Large-Scale IoT Trust Management System Based on Signed Network Embeddings. *IEEE Internet Things J.* **2023**, *10*, 553–562. [CrossRef]
17. Bahutair, M.; Bouguettaya, A.; Neiat, A.G. Multi-Use Trust in Crowdsourced IoT Services. *IEEE Trans. Serv. Comput.* **2023**, *16*, 1268–1281. [CrossRef]
18. Liu, Y.; Zhang, C.; Yan, Y.; Zhou, X.; Tian, Z.; Zhang, J. A Semi-Centralized Trust Management Model Based on Blockchain for Data Exchange in IoT System. *IEEE Trans. Serv. Comput.* **2023**, *16*, 858–871. [CrossRef]
19. Soleymani, S.A.; Goudarzi, S.; Anisi, M.H.; Movahedi, Z.; Jindal, A.; Kama, N. PACMAN: Privacy-Preserving Authentication Scheme for Managing Cybertwin-Based 6G Networking. *IEEE Trans. Ind. Inform.* **2022**, *18*, 4902–4911. [CrossRef]
20. Javed, M.A.; Hamida, E.B.; Al-Fuqaha, A.; Bhargava, B. Adaptive Security for Intelligent Transport System Applications. *IEEE Intell. Transp. Syst. Mag.* **2018**, *10*, 110–120. [CrossRef]
21. Bachiega, J.; Costa, B.; Carvalho, L.R.; Rosa, M.J.F.; Araujo, A. Computational Resource Allocation in Fog Computing: A Comprehensive Survey. *ACM Comput. Surv.* **2023**, *55*, 1–31. [CrossRef]
22. Tian, S.; Deng, X.; Chen, P.; Pei, T.; Oh, S.; Xue, W. A dynamic task offloading algorithm based on greedy matching in vehicle network. *Ad Hoc Netw.* **2021**, *123*, 102639. [CrossRef]
23. Patel, Y.S.; Reddy, M.; Misra, R. Energy and cost trade-off for computational tasks offloading in mobile multi-tenant clouds. *Clust. Comput.* **2021**, *24*, 1793–1824. [CrossRef]
24. Malik, U.M.; Javed, M.A.; Frnda, J.; Rozhon, J.; Khan, W.U. Efficient Matching-Based Parallel Task Offloading in IoT Networks. *Sensors* **2022**, *22*, 6906. [CrossRef] [PubMed]
25. Swain, C.; Sahoo, M.N.; Satpathy, A.; Muhammad, K.; Bakshi, S.; Rodrigues, J.J.P.C.; de Albuquerque, V.H.C. METO: Matching-Theory-Based Efficient Task Offloading in IoT-Fog Interconnection Networks. *IEEE Internet Things J.* **2021**, *8*, 12705–12715. [CrossRef]

26. Alvi, A.N.; Javed, M.A.; Hasanat, M.H.A.; Khan, M.B.; Saudagar, A.K.J.; Alkhathami, M.; Farooq, U. Intelligent Task Offloading in Fog Computing Based Vehicular Networks. *Appl. Sci.* **2022**, *12*, 4521. [CrossRef]
27. Liu, Z.; Yang, X.; Yang, Y.; Wang, K.; Mao, G. DATS: Dispersive Stable Task Scheduling in Heterogeneous Fog Networks. *IEEE Internet Things J.* **2019**, *6*, 3423–3436. [CrossRef]