Toward Energy-Efficient Task Offloading Schemes in Fog Computing: A Survey

Moteb K. Alasmari [1†] Sami S. Alwakeel [2††], Yousef Alohali [3†††],
435107774@student.ksu.edu.sa swakeel@ksu.edu.sa yousef@ksu.edu.sa
King Saud University, College of Computer and Information Sciences, Riyadh, Saudi Arabia

Summary
The interconnection of an enormous number of devices into the Internet at a massive scale is a consequence of the Internet of Things (IoT). As a result, tasks offloading from these IoT devices to remote cloud data centers become expensive and inefficient as their number and amount of its emitted data increase exponentially. It is also a challenge to optimize IoT device energy consumption while meeting its application time deadline and data delivery constraints. Consequently, Fog Computing was proposed to support efficient IoT tasks processing as it has a feature of lower service delay, being adjacent to IoT nodes. However, cloud task offloading is still performed frequently as Fog computing has less resources compared to remote cloud. Thus, optimized schemes are required to correctly characterize and distribute IoT devices tasks offloading in a hybrid IoT, Fog, and cloud paradigm. In this paper, we present a detailed survey and classification of of recently published research articles that address the energy efficiency of task offloading schemes in IoT-Fog-Cloud paradigm. Moreover, we also developed a taxonomy for the classification of these schemes and provided a comparative study of different schemes: by identifying achieved advantage and disadvantage of each scheme, as well its related drawbacks and limitations. Moreover, we also state open research issues in the development of energy efficient, scalable, optimized task offloading schemes for Fog computing.

Keywords:
Internet of things, Fog computing, energy efficiency, task offloading scheme

1. Introduction

The Internet of Things (IoT) is expected to become an unavoidable part of human's life. It will be used in our daily life viz. medical sector, industrial automation, smart homes and emergency response etc. The advent of the Internet of Things (IoT) has allowed the interconnection and intercommunication between massive ubiquitous nodes, creating an unprecedented generation of huge and heterogeneous volumes of data, known as data explosions. International Data Corporation (IDC) predicted that sensor-enabled objects linked to the network would exceed 41.6 billion by 2025 [1].

Cloud computing of IoT machine tasks has recently become an appealing option, offering vast data storage and processing with a cost-effective solution [2]. On the other hand, problems such as the real-time demands or latency-sensitive applications and restricting network bandwidth cannot be solved by using cloud computing alone [3]. Many of these issues are primarily caused by the large physical distance between the End-Users (EU) and the Data Centers (DCs) of cloud service providers (such as Amazon Web Services (AWS), Google, ALTUS, Apple, Facebook, TATA, China Unicom Matrix, Microsoft and Bell etc.) [2] [4]. Besides, due to high exponential rate of generated data from an increasing number of IoT connected devices, Cloud computing of IoT machine tasks does face many challenges and obstacles including [5]:

1. Low Latency Requirement: Many IoT applications have strict latency requirements, particularly for Internet-of-Vehicles and industrial applications. Drone control and vehicle to vehicle communications require latency less than a few tens of milliseconds.

2. Limited Link Bandwidth: Recently, link bandwidth is becoming congested with the increasing number of wirelessly connected IoT devices. Due to limited wireless spectrum, there is limited bandwidth to send all data to the cloud. For this, researchers suggest processing most of the generated data at the end point.

3. Limited Device Energy: Due to the cost and energy constraints, the IoT devices have a limited capability of data communication. Thus, device task offload requires to consider and effectively utilize the available energy in the network devices.

In addition, the wasteful aspects of the IoT cloud computing motivate unnecessarily enormous information to be sent to the backhaul of the system, which weakens the cloud infrastructure. ecosystems To overcome all these challenges, a new IoT computing model, known as fog computing, has been introduced, as a complement to the cloud approach [3]. Fog computing is a distributed computing model where computation is done at the edge of the network with seamless cloud infrastructure integration [6]. It overcomes the restriction of the cloud frameworks by improving robustness, efficiency, and performance of cloud infrastructure. More details of Fog computing paradigm follow:

Cloud-Fog-IoT Computing Paradigm

Various researchers have described Fog computing in different ways. Some definition of Fog computing as follows: "Fog computing is a highly virtualized platform that provides compute, storage, and networking services between IoT devices and traditional cloud computing data centers, typically, but not exclusively located at the edge of network" [6].

Alternatively, Fog computing is defined as a scenario where a large number of IoT ubiquitous, decentralized, and heterogeneous devices communicate and potentially collaborate with each other through the nearby network resources to perform storage and processing tasks without third party intervention. Thus, Fog computing aims to selectively moves processing, storing, communicating, controlling, and decision-making tasks closer to the edge of the network to solve the constraints in the current IoT cloud computing infrastructure. As a result, It expands cloud computing infrastructure to the edge of the network, taking processing, connectivity and storage closer to end-users with the aim of enhancing low latency, network bandwidth, accessibility, security and privacy [6].

The need to process a portion of the large information created at the peripheral of the IoT system utilizing sharp technologies in the fog-cloud environments became a hot research subject and produced many new fascinating structures were recorded in the ongoing research writing [7]. Recent published research in this field, showed that the fog computing models are attractive methods for using resources ideally by the IoT devices, stretching out quality of IoT services to the region of the client by accomplish quick handling of IoT- tasks, permitting quick processing of information, allowing simple storage handling , and reduction of bulky system load[7].

IoT, Fog, and Cloud Computing Paradigm Layers and Services
In general, the IoT, Fog, cloud computing paradigm is a 4-level hierarchical structure with particular layers. Figure 1. Shows the hierarchical layers of the Paradigm structure and services at each layer [7].

Manuscript received March 5, 2022
Manuscript revised March 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.3.22

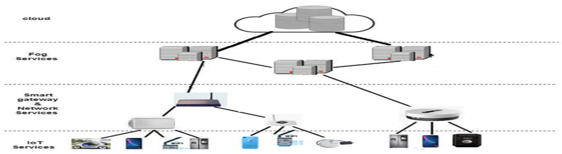

Fig. 1: Could-Fog-IoT paradigm layers

At the base of the structure are the end devices, an assortment of IoT devices consists of sensors, actuators and smart devices, for example, advanced mobile phones [7]. IoT devices comprise advanced sensor parts for detecting the outer environments, computerized converters, storage to store data as well as modules that are used to guarantee system functions. IoT devices effectively senses the surroundings, gathers information regarding perceptions, change and procedure information progressively when supported. Despite the fact that they have limitations in computational power, they can be utilized to achieve forms that may require real-time responsiveness. The next layer is the smart gateways service layer which provide network interface and serves as a connection link between the IOT, Fog and cloud infrastructure. The Fog layer shapes the middle level and the cloud layer frames the upper layer of the structure. Depending on the size and reason, the fog layer supplies the structure with insight that the IoT hubs are not compatible because of absence of adequate computational power. They may integrate artificial capacities that permit them to monitor and control IoT devices, perform traffic movement management. The Fog layer makes the center layer of the IoT-fog-cloud architecture.

The cloud layer mainly provides the services to the IoT and end layer devices in the four -layer architecture. The service is available at anywhere as well as time independent. This layer provides services not only to the IoT devices but also to the fog layer as they need services sometime [6]. For computation intensive tasks, a large volume of data generated by distributed IoT devices are offloaded to remote clouds for processing and results are returned back to data consumers which reduces the burden on IoT devices and prolongs it's battery lifetime [8] [9]. But this causes high delays and a great amount of cost due to the use of cloud-based resources [8]. In this survey, we present a taxonomy, and a comparison for recent published studies that proposed energy efficient schemes for task offloading in fog computing paradigm. We have collected the research papers related to our study from several databases such as Springer, Elsevier, IEEE explore, ACM digital library and Wiley online library that have been published between 2016 and May 2021. To our best of knowledge, this paper is the first survey on energy efficient task offloading protocols in fog computing. The rest of this study is organized as follow: section 2 presents a taxonomy and an overview of energy efficient schemes for task offloading in fog paradigm. and in section three to nine, we present a summary for proposed studies under each category and provide a comparative analysis between them. Finally, section ten presents our study conclusion and future directions.

2. Energy Efficient Schemes Overview

In general, we classifies energy efficient schemes for task offloading into seven categories and list their research articles as shown in Fig. 2:

1. Mathematical Scheme
2. Optimization Scheme
3. Graph Scheme
4. Search Scheme
5. Supervised learning Scheme
6. Reinforcement learning Scheme
7. Performance Enhancement Schemes

Energy Efficiency Scheme	**Methods**
Mathematical Scheme	Probabilistic Method [10]
	Gini coefficient [11]
	Maynard replicator dynamics [12]
	Mathematical model [13]
Optimization Scheme	Multi-Objective Evolutionary Algorithm [14]
	Genetic algorithm (GA) and particle swarm optimization (PSO) [15]
	Greedy-heuristic [16]
	UAVs Trajectory Optimization [17]
	interior-point method and Dinkelbach's algorithm [18]
	Lyapunov optimization [19][20]
	Kuhn-Munkres [21]
Graph Scheme	Bilateral matching game [22]
	Matching game [23], [24]
Search Scheme	Bees search algorithm [25]
	Beetle antennae search [26]
Supervised learning Scheme	Fairness metric [27]
	Classification and regression tree Algorithm [28]
Reinforcement learning Scheme	Bandit learning algorithm[29]
Performance Enhancement Scheme	Drop Computing [30]
	offloading policy[9]
	Blockchain [31]
	Crowd computing [32]
	Energy and time efficient computation offloading and resource allocation (ETCORA) algorithm[8]
	Energy-aware cloud fog offloading (ECFO) [33]

Figure 2: Classification of energy-efficient schemes for task offloading in fog computing.

3. Mathematical Scheme

In this section, we review the mathematical-based Schemes that are used to reduce energy consumption for task offloading in IoT-Fog-Cloud paradigm. This scheme utilizes various mathematical functions, formulas, and concepts to optimize task offloading. Mathematical scheme proves to be effective because the algorithms use formulas and expressions with calculated constraints to achieve precise target results. For example, Wang et al. [11] uses the mathematical concept of Gini Coefficient for offloading decision and selection of fog computing node (FCN).

Kim et al. [10] propose a fog server energy optimization (JUFO) offloading scheme as an alternative to the EMPO scheme. The JUFO scheme leverages the PD (Popularity Distribution) of cloud tasks and the EC (Energy Consumption) model to minimize the joint EC of the UE (User Equipment) and fog server. Compared to the EMPO scheme for the amount of energy consumed in cloud tasks, JUFO shows significantly higher energy savings for a wide range of functionalities. These savings result from the fact that JUFO utilizes the profile of each cloud task in the optimized fog server offloading control scheme. The network model and task operation is given in fig. 3

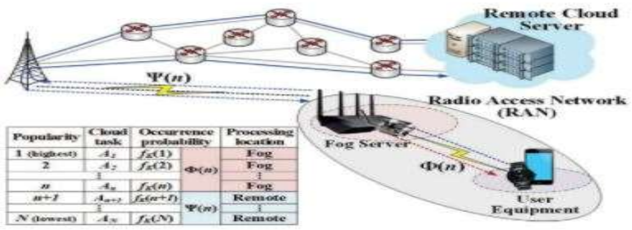

Figure: 3 Cloud network model and cloud task operation [10]

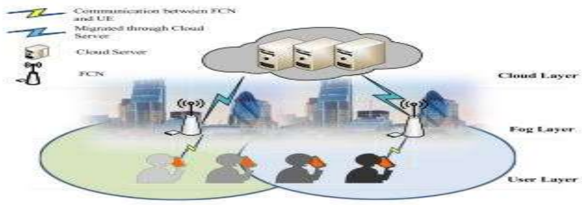

Figure: 4 UEs mobility in the fog computing network [11]

Authors of Mahini et al. [12] have proposed a four-tier architecture based on the evolutionary game approach, in which the IoT gateways have been utilized to solve the task offloading problem. Given the two key optimization factors, two specializations of the general game, namely time games and energy games, have been developed. They are analyzed as evolutionary games and an aggregated solution to both the specialized games is also provided. The Maynard replicator dynamics is the opted dynamic routine for the game, and the analysis shows that their proposed framework successfully decreases latency and energy usage, thus solving the IoT task offloading difficulties.

The performance of CoTs F/FC/C environments, subject to variations in data and queuing policies have been discussed in Aazam et al. [13], to identify factors that can affect the power and performance of the aforementioned environments. The authors state that experimental results can provide optimal solutions to the following- 1) the extent to which a variable can affect the performance, 2) the tradeoffs to be considered, and 3) how to minimize these factors. Mainly, their proposed approach shows that utilizing GG policy for FC environments leads to efficient use of both power and time. The middle ware architecture considered in this study is show in fig. 5.

Scheme	Year	Study	Strategy	Simulatio n Tool	Advantage/achievement	Drawback/limitation
Mathematical Scheme	2019	[10]	Probabilistic Method	MATLAB	Significant energy saving for a wide range of cloud task demands.	Not considered satisfying the stability condition of each network component.
	2019	[11]	Gini coefficient	NA	Maximize the total revenue of user equipment (UEs)	FCN mobility causes incomplete task migration which will cause the extra energy consumption and time delay.
	2020	[12]	Maynard replicator dynamics	MATLAB	Proposed method reduce delay and energy consumption.	the rise of task number the time delay.
	2020	[13]	Mathematical model	SFogSim	Proposed policy saves both time and energy.	Central controller (global gateway) is failure-prone.

Wang et al. [11] offer a method that takes mobility into account while optimizing off-loading decisions and CRA (Computational Resource Allocation) in order to reduce migration risk and increase UE income. The authors proposed two algorithms in the paper: 1) GCFSA (Gini Coefficient based on FCNs Selection Algorithm): for solving the sub-optimal off-loading strategy, and 2) ROAGA (Resource Optimization Algorithm based on Genetic Algorithm): for solving the CRA problem. They claim that their proposed algorithm helps in reducing the migration times for UEs in an FCN (Fog Computing Network). When compared to alternative baseline algorithms, simulations reveal that their

scheme achieves quasi-optimal revenue performance. Fig. 4 shows the three layered fog computing network with UEs mobility.

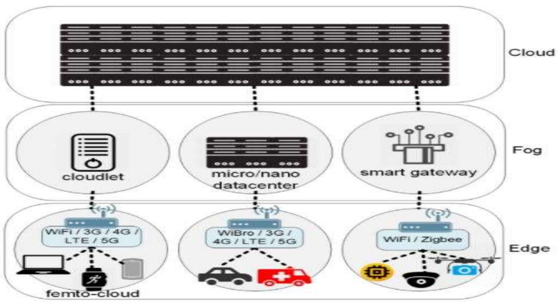

Figure: 5 IoT Middleware technologies [13]

Table I. Comparison of Task offloading based on mathematical Scheme

In this scheme we looked through different mathematical techniques which are used to optimize task offloading and compared their advantages and disadvantages. The mathematical model [13] is proved to be most efficient as it tends to saves both time and energy. However, more work is needed in improving the central controller of the model which being failure-prone seems the only drawback of the approach

4. Optimization Scheme

To achieve the goal of reducing energy consumption for task offloading in fog computing, many modern optimization techniques are presented. These algorithms address various specific application cases and tend to minimize the energy taken for task offloading in the targeted scenario. This section presents a review study of such schemes. Table II shows the Advantages and limitation of each proposed scheme.

Authors Abdullah and Jabir [14] have proposed the Multi-Objective Evolutionary Algorithm (MOEA), which aims to optimize the task offloading process in VFC systems, considering latency and energy objectives under deadline constraints. They have also proposed the RSUs (Road-Side Units) x-Vehicles Multi-Objective Computation offloading method (RxV-MOC), which considers a group of vehicles as fog nodes for executing and transmitting tasks. The widely used all-pair-shortest-path algorithm, Dijkstra, has been used for finding the minimum path between two nodes. Simulation results showed that the proposed scheme, RxV-MOC, outperforms the first-fit, best-fit, and MOC algorithms and significantly reduces the energy usage and latency for VFC systems. The Fig. 6 below shows the comparison results.

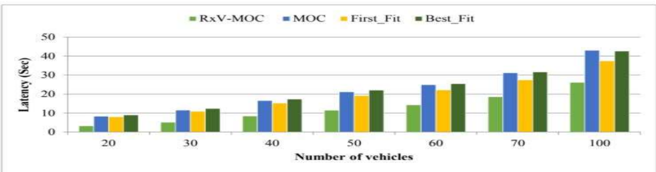

Figure: 6 average latency versus the number of vehicles [14]

Shahryari et al. [15] proposed a task offloading scheme to optimize task offloading decision, fog node selection, and computation resource allocation, while also investigating the trade-off between energy usage and task completion time. The authors have formulated the task offloading problem as an MINLP (Mixed-Integer Nonlinear Program), by weighing the coefficients for the energy and time consumed on the basis of residual energy of the device's battery and the user demands. Considering the NP-Hard nature of the MINLP problem, a sub-optimal solution leveraging a hybrid version of GA and PSO has been proposed. The authors claim that simulation results show that the proposed approach can outperform the standard baseline Scheme. The bar chart in Fig. 7 compares the overhead of the proposed algorithm with base line algorithms.

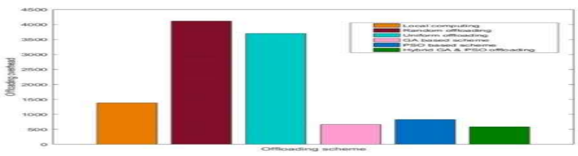

Figure: 7 Comparison of the offloading overhead [15]

A task offloading scheme for software-defined networks SDN , i.e. IoT systems which are connected to fog computing nodes using multi-hop IoT access points, has been proposed in Misra and Saha [16]. To make the best judgments, the suggested scheme uses the SDN controller's global view of the network while simultaneously taking into account the dynamic network conditions. The architecture of the model is shown in Fig. 8. To circumvent the complexities that arise from the nonlinear nature of the task, the authors have presented an ILP(Integer Linear Programming) formulation of the problem, by utilizing a linearization approach. To solve the obtained ILP problem, the authors have presented a greedy-heuristic based method. The proposed approach reduces average delay and energy usage by 12 and 21 per cent, respectively, according to experimental results.

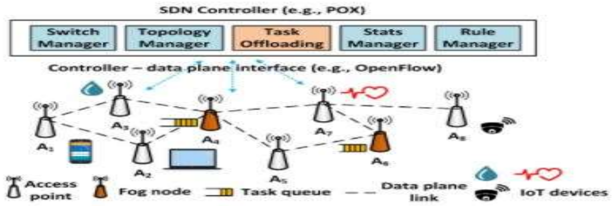

Figure: 8 SDIoT Network architecture [16]

Huang et al. [17] present a task offloading optimization scheme for UAV aided fog enabled IoT networks. The authors aim to reduce the total network overhead, by satisfying the QoS conditions for R-IDs, while simultaneously optimizing the following three parameters: 1) UAV trajectory, 2) Transmission power, and 3) Computation offload radios. To solve the non-convex optimization problem, the authors have proposed the UAV-assisted Task Offloading Optimization algorithm. This algorithm decomposes the original problem into 2 parallel sub-problems, which are then solved alternately. Experiments show that the proposed approach is able to reduce the network overhead, and thus outperform other Scheme present in the literature. The authors in Wang et al. [18] propose an energy-efficient task offloading scheme for massive MIMO-aided multi-pair FCM.

They attempt to reduce the total energy usage by considering realistic imperfect CSI (Channel State Information) to formulate a non-convex joint power allocation and task offloading problem. The arising problem was solved in two steps: 1) Solving the computation resource and computation task allocation for a particular power allocation, and 2) To determine the particular power allocation for minimal energy usage, the authors developed an iterative sequential optimization framework. Simulation results show that compared to benchmark schemes, the proposed method was able to considerably reduce energy usage. Cai et al. [19] propose an algorithm called the JOTE (Joint Offloading of Tasks and Energy) algorithm for fog enabled IoT networks, which aims to minimize task delay and the energy usage for a specific task, in the absence of task queues. According to the authors, as the number of helper nodes increases, it gets progressively more beneficial to jointly offload the energy and task bits.

To minimize the energy usage and task execution delay in the presence of task queues, they developed an online offloading policy on the basis of Lyapunov optimization, which can stabilize the present queues. Numerical experimental results show that JOTE significantly reduces the task delay in fog-enabled IoT networks. An energy-efficient and incentive aware task offloading framework, called D2D fogging, which leverages network-assisted D2D collaboration was proposed in Pu et al. [20]. To minimize the time-energy consumption while maintaining long-term user incentive constraints, a Lyapunov based online offloading policy was developed. The authors devised efficient policies to schedule tasks for every time frame, considering three kinds of system settings.

They claim that their proposed framework is able to achieve offline optimum asymptotically while displaying adaptability to change in task type, task frequency, and user amount.Authors in Yao et al. [21] proposed a scheme called the Kuhn-Munkres based Fair Task Offloading (KFTO) scheme for fog networks with multiple TNs and FNs, which includes a task offloading decision model and an FN selection model. The authors maximized the global potential, which is defined subject to the battery capacity, equivalent data processing rate, and historical average energy consumption, using the KM algorithm. The principal aim of this scheme was to decrease the delay in task processing while continuing to satisfy the constraints on the TNs' energy consumption. Simulation results indicate that the given method can achieve a satisfactory trade-off between the energy usage fairness among FNs and the task processing delay of TNs. To conclude, the optimization scheme algorithms considerably decrease the energy consumption while providing the required quality of service. However, the techniques are applicable to the specific settings with various tradeoffs. As each algorithm does not cover all efficiency parameters, further improvements can be done in the schemes.

Scheme	Year	Study	Strategy	Simulation Tool	Advantage/achievement	Drawback/limitation
Mathematical Scheme	2021	[14]	Multi-Objective Evolutionary Algorithm	MATLAB	Significantly reduced the energy consumption and latency	Transmission energy was not considered when computing the total energy consumption.
	2021	[15]	Genetic algorithm (GA) and particle swarm optimization (PSO)	Real simulation	Proposed algorithm achieved better offloading overhead comparing to others	Limited to similar capabilities of IoT devices.
	2019	[16]	Greedy-heuristic	POX2 SDN controller and the Mininet3 network emulator	Proposed algorithm achieved better average delay and energy consumption	Limited to only static topology where the access points and the fog nodes are considered fixed.
	2021	[17]	UAVs Trajectory Optimization	Real simulation	Reduce the total network overhead (network delay and energy consumption)	Cloud option is not considered
	2021	[18]	interior-point method and Dinkelbach's algorithm	MATLAB	Proposed method achieved better total energy consumption compared to the benchmark schemes	Not consider the scenario of multiple task nodes to multiple computing nodes
	2020	[19]	Lyapunov optimization	NA	Reduced the task execution delay and the energy consumption at the task node	Not considered task queues at node level
	2016	[20]	Lyapunov optimization	Opportunistic Network Environment (ONE) simulator	Proposed algorithm considered various of situations in terms of task type, user amount and task frequency.	Don't consider users' mobility
	2021	[21]	Kuhn-Munkres	NA	Proposed scheme achieved better balance between task processing delay of TNs and energy consumption fairness among fog nodes.	IoT devices have similar capabilities

Table II. Comparison of Task offloading based on optimization Scheme

5. Graph and Search Scheme

In this section, we discuss various schemes that model the fog paradigm network as a graph. Also, AI based search schemes used to find the best candidate node selection to offload the task to. The graph scheme maps the fog architecture to a graph by treating devices as nodes and the link between them as edges, and then using different graph algorithms like Dijkstra's to find the best mechanism to offload the tasks. Similarly Search based techniques utilizes AI based search algorithms to for improving different activities in the fog computing process. A method for joint task offloading and QoS aware resource allocation in fog enabled IoT networks are being proposed by the authors in Huang et al. [22]. The authors claimed that their proposed method can minimize the overhead of the computing networks that consist of Resource Block (RB) allocation, computing resource allocation, and bilateral matching games with task process delay and energy consumption. They have introduced an AHP-based QoS evaluation framework to analyze the several types of IDs with varying QoS requirements. The evaluation framework is show in fig. 9. Simulation results show that their proposed method is more efficient, which could ensure the improvement of RB utilization, and also reduce the network overhead.

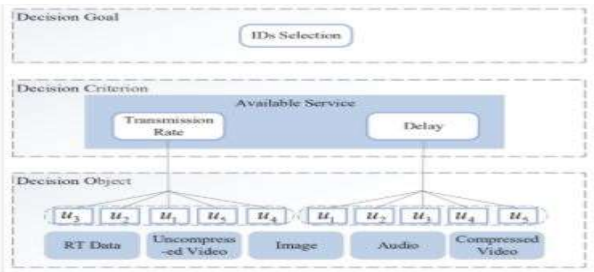

Figure: 9 QoS evaluation framework based on AHP [22]

Swain et al. [21] propose a framework called METO, for densely connected IoT fog networks. Considering a full-offloading scenario, METO aims to minimize the total energy usage and overall latency incurred by the network. Since the problem of offloading is proven to be NP-Hard, its complexity increases exponentially with an increase in the problem size. As a result, the authors consider the problem to be a one-to-many matching game between IoT devices and FNs in polynomial time to identify a sub-optimal solution. Based on simulation results, this scheme beats existing systems in terms of energy usage, completion time, and execution time, as well as fewer outages. Fig. 10 shows the graph comparing battery life and total energy consumption.

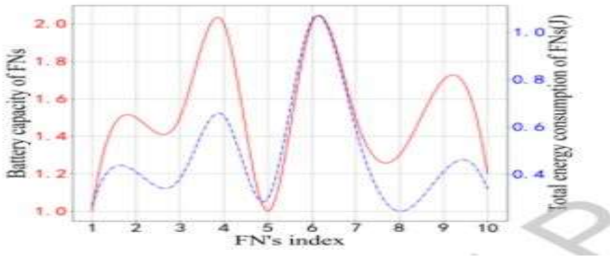

Figure: 10 Battery capacity and total energy consumption in of a fog node [21]

In Chiti et al. [22], the authors suggest a task offloading scheme that is suitable for an integrated edge-fog computing system. By designing an identical game with partial preferences lists and externalities between the task set and the computing sites, the given scheme tries to minimize system energy usage and the total worst completion time. The authors used the following criteria to assess the suggested scheme's performance: 1) mean/worst overall job completion time, 2) mean task communication time, 3) total system energy consumption, and 4) outage probability (i.e. the probability that the computation of a task does not get completed within its associated deadline).

Keshavarznejad et al. [25] used metaheuristic algorithms to investigate the probability of offloading and the amount of energy needed to transmit the data. The authors have used two metaheuristic algorithms: 1) NSGA II, and 2) Bees algorithm. The authors claim that their proposed approach is able to better establish a trade-off between offloading probability and consumed power. They utilized the iFogSim simulator for experimenting with their proposed approach and the obtained results show that their method has a superior response time and reduced energy usage. The topology used for the simulation is shown in fig. 11.

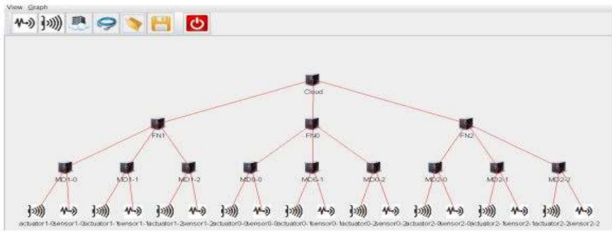

Figure: 11 Simulation topology [25]

A task offloading search method for fog computing networks is proposed by the authors in Li et al. [26]. The scheme leverages an improved contract net protocol and beetle antennae search algorithm to accomplish this task. In the proposed scheme, fog nodes and the task nodes have been uniformly distributed in a circular region with radius R.

Table III: Comparison of Task offloading based on graph and search Scheme

Scheme	Year	Study	Strategy	Simulation Tool	Advantage/achievement	Drawback/limitation
Graph Approach	2020	[22]	Bilateral matching game	NA	Improve resource block RB utilization, and reduce the network overhead.	Use similar task size, FN and IoT devices capabilities
	2020	[13]	Matching game	iFogSim	Improves energy consumption, completion time, execution time and reduces the number of outages	No partial offloading, as well no dynamic preferences.
	2019	[14]	Matching game	NA	Minimize both the system energy consumption and the longest task overall completion time	Limited mobility of edge devices (Eds) and heterogeneous in terms of computation capability and power consumption
Search Approach	2021	[25]	Bees search algorithm	iFogSim	Better trade-off offloading probability and the consumed power at the same time.	Did not investigate the possibility of node failure as well as the deadline for running tasks
	2020	[26]	Beetle antennae search	NA	Achieve high efficiency	Limited computing abilities for the fog nodes

The purpose of the task node is to divide any given task into several sub-tasks and allocate them to the fog nodes. This method takes advantage of the beetle antennae search algorithm and genetic algorithms to minimize the cost of the task node and optimize the process. A summary comparison presents in table III.

To sum up, the graph and search schemes are both effective techniques to improve efficiency of task offloading, have their own setbacks and areas of improvement. The Matching game algorithm significantly improves energy consumption and execution time but has room for improvement to provide dynamic preferences. The Beetle antennae search algorithm provides high efficiency but has restricted computing resources to fog nod

6. Machine Learning Scheme

.In this section, we study different machine learning based schemes that have been proposed recently to implement task offloading and achieve better balance between task delay and energy consumption in fog computing architecture. These algorithms use the fact that machines are becoming intelligent enough to learning from data and patterns. Concepts like regression tree, classification tree, and machine learning are employed to improve performance in fog computing.

Zhang et al. [27] propose an algorithm called the Fair and Energy Minimized Task Offloading (FEMTO) algorithm for fog-enabled IoT networks. The FEMTO method uses a fairness scheduling measure and takes into account the following factors: 1) offloading energy usage, 2) FNs historical average energy, and 3) FN priority. In a fair and energy-efficient manner, the proposed approach obtains analytical conclusions of the optimal TN transmission power, optimal target FN, and optimal subtask size. According to the numerical results of extensive simulations, the suggested scheme offers greater FN feasibility and minimizes energy usage for work offloading. They also claimed that their suggested algorithm can provide a high and reliable level of justice for FN energy use. The simulation parameters used are shown in fig. 12.

Parameter	Value
The radius of the fog cluster	$10-90$ m
W	10 MHz
N	$10-100$
l	$2-8$ MBytes
$d_{\max}$	10 s
$p_{\max}$	1 W
η_{T}	1000 cycle/bit
η_i	$200-2000$ cycle/bit
f_{T}	2 GHz (cycle/s)
f_i	$1-15$ GHz (cycle/s)
θ_{T}	5×10^{-10} J/cycle
θ_i	$1-10\times10^{-10}$ J/cycle
ρ_i	$0.1-1$
Forgetting factor α	0.002

Figure: 12 Simulation parameters [27]

Rahbari and Nickray [28] solved the task offloading problem in mobile fog computing by using the regression and classification tree. The authors have developed an algorithm called MPCA (Module Placement method by Classification and regression tree Algorithm). They have used the MPCA to select the best fog modules. In the proposed approach, at first, the power consumption of MD is checked. Offloading will be done if the detected value is larger than the WiFi's power usage. The proposed approach used seven parameters for selecting the best FD that are: 1) Authentication, 2) Confidentiality, 3) Integrity, 4) Availability, 5) Capacity, 6) Speed and 7) Cost. The authors have also optimized the MPCA by analyzing and applying the probability of network resource utilization in module offloading. This proposed optimized approach is known as MPMCP. The fig. 13 shows the flow chart of the MPMCP approach. They have compared both the MPCA and MPMCP with the First Fit and local processing methods and claimed that their method is superior compared to these.

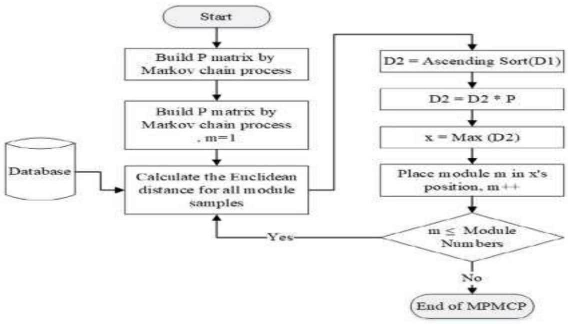

Figure: 28 MPMCP algorithm flow chart [28]

Table IV shows summary review of above studies

Scheme	Year	Study	Strategy	Simulation Tool	Advantage/achievement	Drawback/limitation
Supervised learning Approach	2018	[27]	Fairness metric	NA	Achieved good balance between low energy consumption and fair task offloading between FNs	Not supported time sensitive applications
	2020	[28]	Classification and regression tree Algorithm	Cloudsim	Better energy consumption and task delay compare to first fit (FF) and local mobile	Vulnerable to failure because it use central decision controller.
Reinforcement learning Approach	2019	[29]	Bandit learning algorithm	NA	In online mode, algorithm can select the optimal node to offload task	Cloud option is not considered

Table IV Comparison of machine learning schemes for energy efficient task offloading

Zhu et al. [29] proposed an algorithm called BLOT (Bandit Learning-based Offloading of Tasks) in fog enabled networks. The main aim of the proposed algorithm is to reduce the latency factors, switching cost, long term cost, and energy consumption. In the proposed approach, the authors have also allowed changing abruptly at unknown time instants. They have considered the fact that after finishing the task, then only queried nodes are allowed to give feedback. The authors believed that their proposed algorithm is asymptotically optimal in a fog enabled network. Numerical results verify the performance of BLOT. The success ration versus time graph is shown in fig. 29

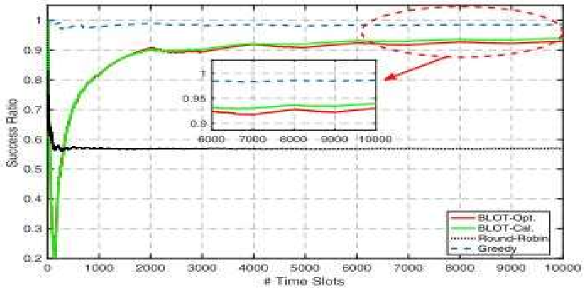

Figure: 29 cumulative successes versus time with breakpoints set to 150 [29]

The machine learning techniques are advanced and provides significant efficiency. As shown in table, fairness metric strategy lowers energy consumption and provide fair task offloading among the fog devices but is not efficient for time sensitive applications. Similarly, classification and regression tree algorithm improve energy consumption and task delay but has a vulnerable because of having a central decision controller.

7. Performance Enhancement Scheme

In this section, we present summary review for a collection of other scheme types, based on their performance in term of processing capability and workload. These schemes are mainly hybrid schemes that may implements one of the previously discussed scheme with the aim towards improving its performance.

Iqbal et al. [31] established a framework for job offloading in micro-level vehicular fog networks. In this framework, the vehicles are viewed as fog nodes with given tasks. At RSU, they maintain a distributed blockchain-based-social-reputation framework, in which a vehicle gets rewarded on the completion of every task. Thus, whenever the decision model encounters a new incoming task, it is able to choose among the trusted vehicles. The findings of the experiment reveal that task distribution based purely on social reputation causes fog resources to be overloaded. The overall performance improves greatly when paired with the typical queue time scheme. Fig. 30 shows the block chain based framework

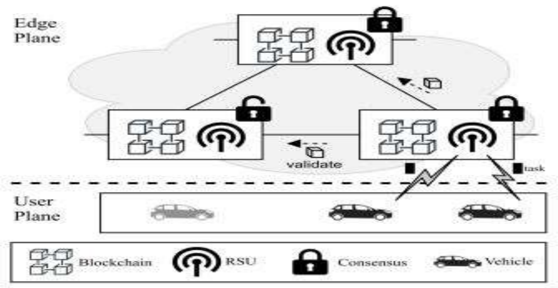

Figure: 30 Blockchain based framework [31]

Wang et al. [32] proposed a task-driven data offloading scheme for urban IoT services. The authors have used a three-layer fog networking architecture with IoT sensors, mobile gateways, and central service servers. The architecture is as shown in fig. 31. The Task-Driven Offloading (TDO) process is developed as a combinatorial optimization problem, considering the abilities of mobile gateways and the task deadlines, making it an NP-hard problem. To solve the formulated NP-hard problem, the authors have devised a G-TDO algorithm. Moreover, the authors have set priorities for the reorganized task and they reorganize the tasks according to each IoT sensor, which leads to the proposal of the RG-TDO algorithm. The authors have used real-world trace datasets and evaluation results show that both the G-TDO and RG-TDO algorithms outperform state of the art algorithms.

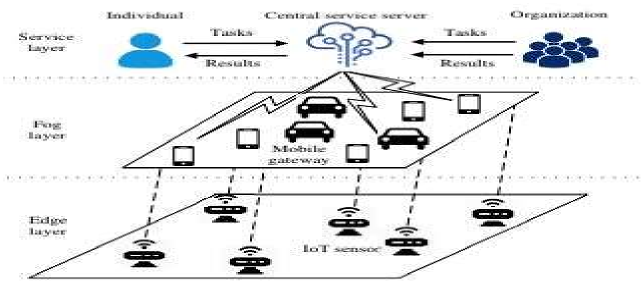

Figure: 31 Fog enabled task driven architecture [32]

The authors of Sun et al. [8] propose a general IoT fog-cloud architecture that can fully harness the Advantages of fog and cloud. The suggested architecture is based on transforming energy and time-efficient computation offloading and resource allocation into a problem of minimizing energy and time costs. To address this issue, the authors presented ETCORA (Energy and Time-Efficient Computation Offloading and Resource Allocation), a new scheme that can reduce energy usage while also improving application request completion times. Simulation results verified that ETCORA was able to outperform the alternate Scheme to solving the given problem.

Wu and Lee [33] proposes an energy-efficient scheduling algorithm for heterogeneous fog computing architectures. In the given work, the authors attempt to maximize the lifespan of a fog given device by minimizing its energy consumption. The developed scheme uses a battery-lifetime and schedule delay aware offloading scheme that is capable of ensuring QoS in real-time. Given that offloading tasks are usually made up of several subtasks, each having an end to end deadline, the authors have also discussed a run-time scheduler with an end to end latency. The authors claim that evaluation results coupled with a real platform study confirm that considerable energy can be saved by using the proposed framework. The offloading framework is shown in fig. 32.

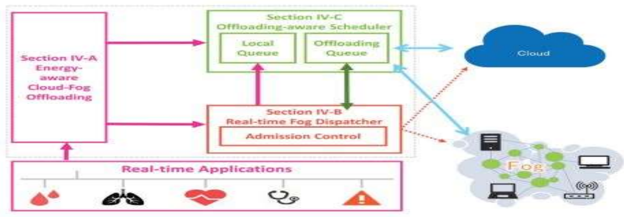

Figure: 32 Cloud fog computing offloading framework [33]

Zhu et al. [9] propose a task offloading decision model for fog computing devices that aims to make fog computing power more accessible to mobile users. The proposed task offloading policy accounts for several factors, including energy usage and execution, and is accompanied by an FCM consisting of remote and local cloud nodes. The authors claim that the experimental results show the efficacy of their proposed approach and that it outperforms other methods in the literature when comparing energy usage and execution time.

A data and task offloading decision scheme were proposed in Ciobanu et al. [30] for collaborative mobile fog-based networks. To tackle the difficulty of mobile data offloading in Drop Computing, the authors seek to shift computation and data from mobile devices to fog nodes or other devices. The authors discover that a crowd computing layer beneath the fog nodes is acceptable for constrained mobile networks by simulating real-world and artificial scenarios. Furthermore, they claim that the proposed scheme is efficient as it reduces cloud usage, decreases total computation time and improves battery consumption. Table V shows summary review of above studies.

All the performance enhancement schemes utilize different technologies to achieve the target efficiency. Block chain based solution brings an innovative concept of social reputation in the fog computing paradigm and also provide further study opportunity to scale the solution. Drop computing improve battery consumption but is limited to tasks with same size of MB.

Table V Comparison of Performance Enhancement schemes for energy efficient task offloading

Scheme	Year	Study	Strategy	Simulation Tool	Advantage/achievement	Drawback/limitation
	2020	[31]	Blockchain	AnyLogic 8	Utilized combination of social reputation with the traditional queuing time approach significantly improves the overall performance	Used similar tasks size, roadside units (RSUs) and fog vehicles devices capabilities
	2020	[32]	Greedy algorithm	Real-world trace	Proposed algorithm achieved better value of successful ratio, average task cost, average task completion ratio, and system total cost comparing to others.	Number of offloading tasks limited to available mobile gateways.
	2019	[8]	Energy and time efficient computation offloading and resource allocation (ETCORA) algorithm	iFogSim	Reduced energy consumption and completion time of IoT application requests	Not considered security and reliability of services, because they would also have direct impact on the performance of the IoT applications.
	2019	[33]	Energy-aware cloud fog offloading (ECFO)	real platform	Proposed approach achieved better tradeoff between task response and device energy consumption.	Limited to assumption that the network links are supported by reliability schemes based on retransmission and redundancy schemes and assume no data loss.
	2017	[9]	offloading policy	MATLAB	The proposed policy considered multiple expenses.	Don't consider others energy consumption factors such as in information collection
	2019	[30]	Drop Computing	MobEmu mobile network simulator	Proposed solution achieved better battery consumption	Consider only tasks with same size 5 MB

Table V Comparison of Performance Enhancement schemes for energy efficient task offloading

8. Conclusion

Task offloading is critical in fog computing to enable internet of things to process intensive requests by executing them remotely on fog or cloud node. Task offloading required additional data communication which may increase energy consumption. Thus, to determine whether task offloading is beneficial or not, the IoT node should check whether the time and energy consumption executing remotely the task on fog or cloud is less than executing it locally. In this review article, we present summaries for most recent proposed schemes that used in task offloading to achieve better energy consumption. We classified those studies into seven categories based on their algorithmic approach and provided comparison between them from different perspectives, advantage, disadvantages, achievements, and limitations. In general AI-based schemes dominated this research area but limited studies utilized Machine learning approach to optimize task offloading. Considerable attention in future research may therefore concentrate in implementing various ML techniques for development of an optimized energy efficient task offloading scheme in IOT/Fog Computing environment.

References

[1] A. Wasicek, "The future of 5G smart home network security is micro-segmentation," Netw. Secur., vol. 2020, no. 11, pp. 11–13, 2020, doi: 10.1016/S1353-4858(20)30129-X.

[2] M. Mukherjee, L. Shu, and D. Wang, "Survey of fog computing: Fundamental, network applications, and research challenges," IEEE Commun. Surv. Tutorials, vol. 20, no. 3, pp. 1826–1857, 2018, doi: 10.1109/COMST.2018.2814571.

[3] P. Hu, S. Dhelim, H. Ning, and T. Qiu, "Survey on fog computing: architecture, key technologies, applications and open issues," J. Netw. Comput. Appl., vol. 98, no. September, pp. 27–42, 2017, doi: 10.1016/j.jnca.2017.09.002.

[4] C. Puliafito, E. Mingozzi, F. Longo, A. Puliafito, and O. Rana, "Fog computing for the Internet of Things: A survey," ACM Trans. Internet Technol., vol. 19, no. 2, 2019, doi: 10.1145/3301443.

[5] R. K. Naha, S. Garg, and A. Chan, "Fog-computing architecture: survey and challenges," Big Data-Enabled Internet Things, pp. 199–223, 2019, doi: 10.1049/pbpc025e_ch10.

[6] R. K. Naha et al., "Fog computing: Survey of trends, architectures, requirements, and research directions," IEEE Access, vol. 6, pp. 47980–48009, 2018, doi: 10.1109/ACCESS.2018.2866491.

[7] A. A. Alli and M. M. Alam, "The fog cloud of things: A survey on concepts, architecture, standards, tools, and applications," Internet of Things, vol. 9, p. 100177, 2020, doi: 10.1016/j.iot.2020.100177.

[8] H. Sun, H. Yu, G. Fan, and L. Chen, "Energy and time efficient task offloading and resource allocation on the generic IoT-fog-cloud architecture," Peer-to-Peer Netw. Appl., vol. 13, no. 2, pp. 548–563, 2020, doi: 10.1007/s12083-019-00783-7.

[9] Q. Zhu, B. Si, F. Yang, and Y. Ma, "Task offloading decision in fog computing system," China Commun., vol. 14, no. 11, pp. 59–68, 2017, doi: 10.1109/CC.2017.8233651.

[10] J. Kim, T. Ha, W. Yoo, and J. M. Chung, "Task Popularity-Based Energy Minimized Computation Offloading for Fog Computing Wireless Networks,"

IEEE Wirel. Commun. Lett., vol. 8, no. 4, pp. 1200–1203, 2019, doi: 10.1109/LWC.2019.2911521.

[11] D. Wang, Z. Liu, X. Wang, and Y. Lan, "Mobility-Aware Task Offloading and Migration Schemes in Fog Computing Networks," IEEE Access, vol. 7, pp. 43356–43368, 2019, doi: 10.1109/ACCESS.2019.2908263.

[12] H. Mahini, A. M. Rahmani, and S. M. Mousavirad, "An evolutionary game approach to IoT task offloading in fog-cloud computing," J. Supercomput., vol. 77, no. 6, pp. 5398–5425, 2021, doi: 10.1007/s11227-020-03484-8.

[13] M. Aazam, S. U. Islam, S. T. Lone, and A. Abbas, "Cloud of Things (CoT): Cloud-Fog-IoT Task Offloading for Sustainable Internet of Things," IEEE Trans. Sustain. Comput., vol. 3782, no. c, pp. 1–13, 2020, doi: 10.1109/TSUSC.2020.3028615.

[14] S. Abdullah and A. Jabir, "A Light Weight Multi-Objective Task Offloading Optimization for Vehicular Fog Computing," Iraqi J. Electr. Electron. Eng., vol. 17, no. 1, pp. 1–10, 2021, doi: 10.37917/ijeee.17.1.8.

[15] O. K. Shahryari, H. Pedram, V. Khajehvand, and M. D. TakhtFooladi, "Energy and task completion time trade-off for task offloading in fog-enabled IoT networks," Pervasive Mob. Comput., vol. 74, p. 101395, 2021, doi: 10.1016/j.pmcj.2021.101395.

[16] S. Misra and N. Saha, "Detour: Dynamic Task Offloading in Software-Defined Fog for IoT Applications," IEEE J. Sel. Areas Commun., vol. 37, no. 5, pp. 1159–1166, 2019, doi: 10.1109/JSAC.2019.2906793.

[17] X. Huang, X. Yang, Q. Chen, and J. Zhang, "Task Offloading Optimization for UAV-assisted Fog-enabled Internet of Things Networks," IEEE Internet Things J., vol. 4662, no. c, 2021, doi: 10.1109/JIOT.2021.3078904.

[18] K. Wang, Y. Zhou, J. Li, L. Shi, W. Chen, and L. Hanzo, "Energy-Efficient Task Offloading in Massive MIMO-Aided Multi-Pair Fog-Computing Networks," IEEE Trans. Commun., vol. 69, no. 4, pp. 2123–2137, 2021, doi: 10.1109/TCOMM.2020.3046265.

[19] P. Cai, F. Yang, J. Wang, X. Wu, Y. Yang, and X. Luo, "JOTE: Joint Offloading of Tasks and Energy in Fog-Enabled IoT Networks," IEEE Internet Things J., vol. 7, no. 4, pp. 3067–3082, 2020, doi: 10.1109/JIOT.2020.2964951.

[20] L. Pu, X. Chen, J. Xu, and X. Fu, "D2D Fogging: An Energy-Efficient and Incentive-Aware Task Offloading Framework via Network-Assisted D2D Collaboration," IEEE J. Sel. Areas Commun., vol. 34, no. 12, pp. 3887–39014, 2016, doi: 10.1109/JSAC.2016.2624118.

[21] Y. Yao et al., "KFTO: Kuhn-Munkres based fair task offloading in fog networks," Comput. Networks, vol. 195, p. 108131, 2021, doi: 10.1016/j.comnet.2021.108131.

[22] X. Huang, Y. Cui, Q. Chen, and J. Zhang, "Joint Task Offloading and QoS-Aware Resource Allocation in Fog-Enabled Internet-of-Things Networks," IEEE Internet Things J., vol. 7, no. 8, pp. 7194–7206, 2020, doi: 10.1109/JIOT.2020.2982670.

[23] C. Swain et al., "METO: Matching Theory Based Efficient Task Offloading in IoT-Fog Interconnection Networks," IEEE Internet Things J., vol. 4662, no. c, pp. 1–1, 2020, doi: 10.1109/jiot.2020.3025631.

[24] F. Chiti, R. Fantacci, and B. Picano, "A matching game for tasks offloading in integrated edge-fog computing systems," Trans. Emerg. Telecommun. Technol., vol. 31, no. 2, pp. 1–14, 2020, doi: 10.1002/ett.3718.

[25] M. Keshavarznejad, M. H. Rezvani, and S. Adabi, "Delay-aware optimization of energy consumption for task offloading in fog environments using metaheuristic algorithms," Cluster Comput., vol. 0123456789, 2021, doi: 10.1007/s10586-020-03230-y.

[26] X. Li, Z. Zang, F. Shen, and Y. Sun, "Task Offloading Scheme Based on Improved Contract Net Protocol and Beetle Antennae Search Algorithm in Fog Computing Networks," Mob. Networks Appl., vol. 25, no. 6, pp. 2517–2526, 2020, doi: 10.1007/s11036-020-01593-5.

[27] G. Zhang, F. Shen, Z. Liu, Y. Yang, K. Wang, and M. T. Zhou, "FEMTO: Fair and energy-minimized task offloading for fog-enabled IoT networks," IEEE Internet Things J., vol. 6, no. 3, pp. 4388–4400, 2019, doi: 10.1109/JIOT.2018.2887229.

[28] D. Rahbari and M. Nickray, "Task offloading in mobile fog computing by classification and regression tree," Peer-to-Peer Netw. Appl., vol. 13, no. 1, pp. 104–122, 2020, doi: 10.1007/s12083-019-00721-7.

[29] Z. Zhu, T. Liu, Y. Yang, and X. Luo, "BLOT: Bandit Learning-Based Offloading of Tasks in Fog-Enabled Networks," IEEE Trans. Parallel Distrib. Syst., vol. 30, no. 12, pp. 2636–2649, 2019, doi: 10.1109/TPDS.2019.2927978.

[30] R. I. Ciobanu, C. Dobre, M. Bălănescu, and G. Suciu, "Data and task offloading in collaborative mobile fog-based networks," IEEE Access, vol. 7, pp. 104405–104422, 2019, doi: 10.1109/ACCESS.2019.2929683.

[31] S. Iqbal, A. W. Malik, A. U. Rahman, and R. M. Noor, "Blockchain-based reputation management for task offloading in micro-level vehicular fog network," IEEE Access, vol. 8, pp. 52968–52980, 2020, doi: 10.1109/ACCESS.2020.2979248.

[32] P. Wang, R. Yu, N. W. Gao, C. Lin, and Y. Liu, "Task-driven Data Offloading for Fog-enabled Urban IoT Services," IEEE Internet Things J., vol. XX, no. XX, pp. 1–13, 2020, doi: 10.1109/JIOT.2020.3039467.

[33] Y. L. Jiang, Y. S. Chen, S. W. Yang, and C. H. Wu, "Energy-Efficient Task Offloading for Time-Sensitive Applications in Fog Computing," IEEE Syst. J., vol. 13, no. 3, pp. 2930–2941, 2019, doi: 10.1109/JSYST.2018.2877850.