

Graphical Abstract

Highlights

- Research highlight 1
- Research highlight 2

Abstract

Abstract text.

Keywords:

1. Introduction*1.1. Background and motivation*

The Internet of Things (IoT) has become a foundational element of modern technology, facilitating the development of innovative applications in domains such as smart cities, healthcare, autonomous systems, and other fields. The proliferation of IoT devices has been rapid and significant, with Cisco reporting a global total exceeding 30 billion (Benaboura et al.). These devices generate a substantial volume of data, approximately 2 exabytes on a daily basis. Achieving maximum potential from these systems necessitates the implementation of efficient processing and analysis methodologies. This requirement presents a formidable challenge in the domains of data management, resource allocation, and system scalability.

The inherent limitations of IoT devices, such as their small batteries, limited processing power, and minimal storage capacity, render them ill-suited to manage the substantial volumes of data they generate. It is evident that tasks such as real-time processing of sensor data or computation-intensive applications frequently exceed the capabilities of local devices. Conventional approaches entail the delegation of these tasks to centralized cloud servers. However, network limitations and latency sensitivity frequently render this approach inefficient, particularly for real-time applications (Benaboura et al.).

In order to address these challenges, fog computing has emerged as a distributed computing paradigm that extends the capabilities of cloud computing to the edge of the network. By facilitating the execution of storage, computation, and data management operations in close proximity to the data

source, fog computing contributes to the reduction of latency, power consumption, and network traffic. It is evident that contemporary applications, including but not limited to smart homes, autonomous vehicles, smart agriculture, and healthcare, are contingent on this paradigm in order to satisfy their real-time and location-aware processing requirements as evoked by Das and Inuwa (2023). Fog computing, first introduced in 2012, provides a hierarchical architecture that serves to bridge the gap between cloud servers and IoT devices, thereby allowing for seamless data flow and operational efficiency (Fahimullah et al., 2022).

Notwithstanding the advantages inherent in task offloading in fog computing, this process is encumbered by numerous challenges. Optimizing resource utilization, reducing energy consumption, and maintaining Quality of Service (QoS) require robust strategies due to the heterogeneous and dynamic nature of fog networks. Factors such as fluctuating workloads, mobility, and task diversity have been identified as contributing to an increase in the complexity of the situation. In order to address these challenges, innovative resource allocation techniques are required, including Machine Learning-based (ML) methods, auction models, and heuristic optimization as Fahimullah et al. (2022) exhibits.

Conventional resource management techniques frequently employ static heuristic approaches, which prove ineffective when confronted with the diverse and dynamic workloads that are characteristic of fog environments. These methods, when configured offline for specific scenarios, lack the scalability and flexibility required for real-time task offloading and resource optimization. Consequently, a substantial decline in performance is experienced as system demands increase (Iftikhar et al., 2023).

To address this problem, metaheuristics such as GA have emerged as powerful optimization techniques capable of handling the multi-objective nature of fog computing task offloading. GAs excel in exploring complex solution spaces and finding near-optimal trade-offs between conflicting objectives such as latency minimization, energy efficiency, and resource utilization. Unlike traditional optimization methods that often focus on single objectives, multi-objective genetic algorithms can simultaneously optimize multiple performance criteria, making them particularly suitable for fog computing environments where various QoS parameters must be balanced. The evolutionary nature of GAs allows them to adapt to dynamic network conditions and heterogeneous resource availability, providing robust solutions for real-time task offloading decisions.

Recent advancements in the field of Artificial Intelligence (AI), particularly deep reinforcement learning (DRL), have demonstrated considerable potential in addressing the intricacies of task offloading in fog environments. Research has demonstrated the efficacy of AI-driven approaches in reducing latency, energy consumption, and operational costs. As Fahimullah et al. (2022) demonstrates in their review, techniques such as centralized Dueling Deep Q-Networks (DDQNs), decentralized learning models, and multi-agent reinforcement learning have been employed to optimize offloading policies and resource allocation. Furthermore, hybrid strategies that integrate artificial intelligence with conventional methods have demonstrated efficacy in heterogeneous fog environments (Mishra et al., 2023).

In order to maintain currency with the latest advancements in this domain, the present study investigates innovative methodologies that integrate the strengths of evolutionary computation and deep reinforcement learning for the purpose of optimizing IoT task offloading. The integration of these approaches addresses the limitations of individual techniques while leveraging their complementary advantages.

1.2. Contributions

To address these gaps, this work makes the following key contributions:

- The proposal of a two-transformers-based architecture, utilizing DQRL for the purpose of task offloading within a cloud-fog-edge environment, is hereby presented. The selection of the node (edge, fog or cloud) to execute each incoming task is determined by these models, representing different configurations. The NOTE system is oriented towards node-level features, such as CPU, buffer, and bandwidth. In contrast, the T-NOTE system incorporates additional task attributes, including size, deadline, and CPU-cycle requirements. This capability facilitates a more precise depiction of the interplay between node resources and task demands within a fog environment.
- A transformation of the conventional static genome to a dynamic one is achieved by adapting the Niched Pareto Genetic Algorithm (NPGA) and the Non-dominated Sorting Genetic Algorithm II (NSGA-II), two multi-objective genetic algorithms (GAs), to a MLP as the genome. Subsequently, a comparative analysis was conducted between the GAs and a Deep Q-Learning (DQRL) approach. The implementation of

these algorithms in the training of a MLP constitutes a pivotal element of the study. The GA approach is employed in a dynamic setting, thereby enabling the MLP to be optimized for multi-objective offloading in real time.

- This work also puts forth a model for offloading in hybrid environments that considers a substantial number of parameters for the QoS. To further expand upon the existing body of knowledge, an adaptation of a scenario with a real dataset was implemented within the framework of RayClousSim, which is available at <https://github.com/tutur90/Task-Offloading-Fog>.

1.3. Paper organization

The reminder of this paper is structured as follows: Section 2 reviews related work;.....

2. Related Work

Task offloading in Fog/Cloud environments is a relatively recent research area compared to well-studied domains like image classification or time series forecasting. However, the decision-making process for task offloading is inherently complex due to its combinatorial nature.

2.1. Deterministic Approaches

Deterministic algorithms provide a direct method for solving the task offloading problem. For instance, the optimal task assignment algorithm proposed by Yan *et al.* Yan et al. (2020) employs a three-step approach: (i) assuming the offloading decisions are pre-determined, (ii) deriving closed-form expressions for optimal offloading, and (iii) implementing bisection search and a one-climb policy. This approach effectively reduces energy consumption and execution time for IoT tasks in Mobile Edge Computing (MEC) systems.

Nevertheless, the task offloading problem is considered NP-hard, as demonstrated in Guo et al. (2024); Jin et al. (2024); Sarkar and Kumar (2022). Consequently, deterministic methods face scalability limitations, prompting the exploration of heuristic, metaheuristic, and AI-based approaches.

2.2. Heuristic and Metaheuristic Methods

Heuristic methods are well-suited for scaling in complex edge/cloud environments, as they avoid the exponential computational growth of deterministic algorithms Zhang et al. (2024). For example, the Deadline and Priority-aware Task Offloading (DPTO) algorithm Adhikari et al. (2020) schedules tasks by prioritizing delays and task priorities, selecting optimal devices to minimize overall offloading time.

Metaheuristic algorithms, known for their adaptability, have also shown strong performance. (Bernard et al., 2024) introduced the Drafting Niche Pareto Genetic Algorithm (D-NPGA), which optimizes task offloading decisions and improves makespan and cost efficiency for IoT tasks in fog/cloud systems.

More sophisticated methods use multiple approaches. For example, Energy-Efficient and Deadline-Aware Task Scheduling in Fog Computing (ETFC) Pakmehr et al. (2024) employs a Support Vector Machine (SVM) to predict traffic on fog nodes and classify them as low- or high-traffic. Then, they use reinforcement learning (RL) on the low-traffic group and a non-dominated sorting genetic algorithm III (NSGA-III) on the high-traffic group to make the offloading decision. This allows both algorithms to perform better on adequate tasks.

Metaheuristic approaches have gained attention for their ability to address the NP-hard nature of the task offloading problem with flexibility and adaptability. A recent and comprehensive survey by Rahmani et al. (2025) reviewed a wide range of metaheuristic algorithms—including Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Grey Wolf Optimizer (GWO)—applied to task offloading in IoT environments. The review identified key strengths of these methods in balancing latency, energy consumption, and cost efficiency across heterogeneous fog-edge-cloud infrastructures. Their taxonomy also distinguishes between evolutionary and swarm-based algorithms, showing that hybrid models are increasingly used to adapt to dynamic network conditions. Despite their strengths, the authors noted limitations in scalability and real-time adaptability—areas where AI and deep learning techniques, such as DRL or transformers, may provide an edge.

2.3. AI Approaches

Machine learning (ML) has recently demonstrated its efficacy in task offloading. For instance, decision tree classifiersSuryadevara (2021) determine

whether tasks should be offloaded to the fog or cloud, resulting in reduced latency and energy consumption. Logistic regression models Bukhari et al. (2022) have also been applied, estimating the probability of successful offloading by leveraging maximum likelihood estimation.

Deep learning (DL), a subset of ML, deserves special attention due to its notable advances across various fields.

Basic deep neural networks (DNNs) have been applied to IoT offloading tasks, such as in Sarkar and Kumar (2022), where parallel DNNs optimize cost, energy consumption, and latency. However, deep reinforcement learning (DRL) is often preferred for its decision-making capabilities in dynamic environments.

For example, Jiang *et al.* Jiang et al. (2021) utilized Deep Q-Learning (DQN) to identify optimal offloading policies and resource allocation strategies for user equipment in fog/cloud systems. Their approach combines a dueling deep Q-network for model pre-processing and a distributed deep Q-network for efficient task allocation.

Moreover, multi-agent DRL methods Ren et al. (2021) have been employed to offload IoT tasks. Each IoT device trains its DRL model to select fog access points, followed by a greedy algorithm to determine cloud offloading. This approach demonstrates competitive performance in energy efficiency compared to exhaustive search and genetic algorithms.

Long Short-Term Memory (LSTM) networks, a type of recurrent neural network, are particularly suitable for the temporal dimensions of task offloading. Tu *et al.* Tu et al. (2022) combined LSTM with DQN to predict task dynamics in real-time, leveraging observed edge network conditions and server load. This hybrid model significantly improved latency, offloading cost, and task throughput.

Transformers, known for their revolutionary impact on natural language processing (NLP), have also been adapted for task offloading. Gholipour *et al.* Gholipour et al. (2023) proposed TPTO, a transformer-based framework with Proximal Policy Optimization (PPO). Their model encodes task dependencies using a transformer encoder and employs an actor-critic framework trained with PPO to generate probability distributions for offloading actions, achieving state-of-the-art results in edge computing environments.

While many existing offloading solutions rely on synthetic datasets and focus on Multi-access or Vehicular Edge Computing (MEC/VEC) scenarios Fahimullah et al. (2022); Tu et al. (2022); Gholipour et al. (2023), research specifically targeting fog computing remains limited. In particular, models

such as Deep Neural Networks (DNNs) Sarkar and Kumar (2022) and Deep Q-Learning (DQL) Jiang et al. (2021) have rarely been evaluated in realistic fog environments using real-world data.

Genetic Algorithms (GAs) are often static in nature, such as in ?Pakmehr et al. (2024), meaning that they cannot be employed in real-time applications and can only converge through iterations on the same simulation environment where tasks and their order remain static. This characteristic makes them particularly complex to deploy in real-world applications where dynamic adaptation is required.

Transformer-based deep learning techniques remain poorly explored in this domain, despite being state-of-the-art in many fields. Some exploration efforts show promise; however, existing approaches often implement basic action spaces, such as TPTO Gholipour et al. (2023), which is designed with only two actions: offloading the task to the cloud or processing it on the edge.

Algorithms are referenced in the Table 2.3 with the method, the dataset type, the QOS evaluated, and the job type. Used QOS are referenced in the Table 2.3 with their description.

Paper	Method	Algorithm	Dataset	QoS	Type of Job	Environment	Tool
Yan et al. (2020)	Deterministic	Optimal Task Assignment	Real	E, ET	IoT tasks	MEC	NA
Adhikari et al. (2020)	Heuristic	DPTO	Synthetic	QWT, D	IoT tasks	Cloud & Fog	NA
?	Metaheuristic	D-NPGA	Synthetic	C, M	IoT tasks	Cloud & Fog	Python
This work	Metaheuristic	NPGA/NSGA-II+MLP	Real	C, L, TTR	IoT tasks	Cloud & Fog	Python, PyTorch
Pakmehr et al. (2024)	AI, Metaheuristics	ETFC	C, E, L, DLV	Synthetic	IoT tasks	Fog	NA
Bukhari et al. (2022)	AI	Logistic Regression (LR)	Real	C, E, L	IoT tasks	Cloud & Fog	Python, Matlab
Suryadevara (2021)	AI	Decision Tree (DT)	Synthetic	E, L	IoT tasks	Cloud & Fog	iFogSim
Sarkar and Kumar (2022)	AI	Deep Neural Network (DNN)	Synthetic	C, E, L	Mobile	Cloud & Fog	iFogSim
Jiang et al. (2021)	AI	Deep Q-Network (DQN)	Synthetic	E, L	Mobile	Cloud & Fog	Python, Adam optimizer
Ren et al. (2021)	AI	Multi-agent DRL	Synthetic	E	IoT tasks	Cloud & Fog	NA
Tu et al. (2022)	AI	DRL+LSTM	Real	C, L, TTR	Mobile	MEC	NA
Gholipour et al. (2023)	AI	Transformers PPO	Synthetic	L	IoT tasks	Edge	Python, TensorFlow
This work	AI	DQRL+MLP	Real	C, L, TTR	IoT tasks	Cloud & Fog	Python, PyTorch
This work	AI	Transformers	Real	C, L, TTR	IoT tasks	Cloud & Fog	Python, PyTorch

Table 1: Summary of Task Offloading Methods in Fog and Edge Computing

Metric	Description
C: Cost	Refers to the expenses incurred in task offloading, including computation, storage, and data transfer costs.
L: Latency	The time delay between the initiation of a request and the reception of the response. Crucial for real-time applications.
E: Energy	The total energy consumed during task offloading, including device and network-level energy usage.
ET: Execution Time	The total time taken to execute a task from start to finish, including computation and communication time.
D: Delay	The time difference between task submission and the start of its processing. Includes network and processing delays.
M: Makespan	The total time required to complete all tasks in a batch or workflow. Indicates overall system efficiency.
QWT: Queue Waiting Time	The duration a task spends waiting in the queue before being processed. Impacts response time and throughput.
TTR: Task Throw Rate	The rate at which tasks are successfully processed and completed in the system, indicating system throughput.
DLV: Dead Line Violation	The percentage or rate of tasks that succeed within their specified deadlines, indicating system reliability and quality of service.

Table 2: Explanation of QoS Metrics in Edge-Fog-Cloud Computing

3. Problem Modeling

3.1. Scenario Modeling

This scenario is modeled according to foundational concepts presented in Aazam et al. (2022); Bukhari et al. (2022); Jazayeri et al. (2021), following a three-tier offloading approach that involves *edge*, *fog*, and *cloud* nodes. In Figure 1, a high-level cloud-enabled architecture is shown, where a **Global Gateway (GG)** collects tasks from various IoT devices before determining whether to process them locally or offload them to fog or cloud resources.

A real-world dataset of IoT-generated tasks, is employed to study and evaluate these offloading decisions. The tasks in this dataset vary in size, deadline constraints, and computational complexity, thereby reflecting the heterogeneous nature of practical IoT workloads.

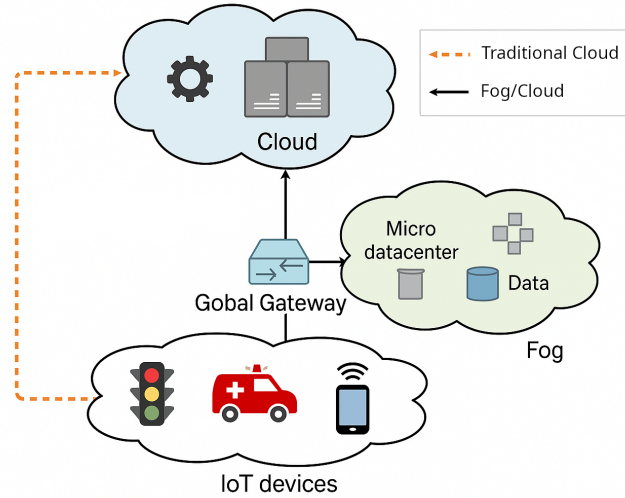


Figure 1: System Architecture

3.1.1. Architecture

In Figure 1, a schematic overview of the **proposed three-tier architecture** is illustrated. The system is composed of IoT devices (sensors, mobile nodes, and other edge devices) that generate tasks. These tasks subsequently reach the Global Gateway, which processes them locally if resources are available, or decides to offload them to the fog or the cloud based on resource availability and energy considerations.

Global Gateway. The GG layer is located at the edge and aggregates tasks from local IoT devices. In addition to routing capabilities, the GG has limited computational power, enabling it to handle *smaller tasks* locally and thus reduce network traffic. This approach is particularly advantageous for time-sensitive tasks that can be processed quickly on-site.

Upon receiving a task, several factors are evaluated:

- Node state: The current load on the GG, along with the availability of fog and cloud nodes.
- Network conditions: Bandwidth, potential congestion, and latency to fog or cloud nodes.
- Energy usage: A predictive assessment of energy costs if the task is executed locally versus offloaded.
- Task requirements: Size, computational complexity, and deadline or QoS constraints.

A subsequent decision is made to determine the optimal execution strategy for the task. The system evaluates three primary options: processing the task locally if it is computationally feasible and resource-efficient, offloading the task to a nearby Fog Node to leverage edge computing capabilities, or offloading the task directly to a Cloud Node for intensive computational requirements that exceed local and fog node capacities.

Fog Nodes. Fog nodes provide moderate computational resources that exceed those of edge devices but remain below the capacity of cloud data centers. These intermediate computing nodes offer lower latency than distant cloud nodes due to their closer physical proximity to the network edge. Fog nodes serve as an essential intermediate tier for tasks that cannot be processed locally at the Gateway (GG) yet do not require the extensive resources of the cloud infrastructure. This positioning makes them ideal for applications requiring balanced performance between computational capability and response time.

Cloud Nodes. Cloud nodes represent large-scale data centers with substantial computational and storage capabilities. Although they typically have higher baseline latencies because of greater network distances, they are well suited for computationally intensive applications. Cloud nodes excel in handling

large tasks that demand significant CPU and memory resources, making them optimal for complex analytical workloads. They are particularly effective for batch processing scenarios with relaxed real-time constraints, where throughput is prioritized over immediate response. Furthermore, cloud data centers can accommodate high concurrency situations through dynamic resource scaling, allowing them to adapt to varying computational demands efficiently. concurrency situations, as cloud data centers can scale resources dynamically.

CPU capacity: Cloud servers can support a high volume of CPU cycles, facilitating efficient offloading for tasks with elevated computational demands.

3.2. QoS Modeling

Task offloading decisions are evaluated within the RayCloudSim framework developed by Zhang et al. (2022), a comprehensive simulation platform for modeling and assessing cloud-edge-IoT computing environments based on LEAF (Wiesner and Thamsen, 2021).

3.2.1. Task Throw Rate

In distributed environments, task execution failures can occur due to issues such as *network disconnections*, *node isolation*, or *buffer overflows*.

The *task throw rate* τ is defined as:

$$\tau = \frac{\text{Number of failed tasks}}{\text{Total tasks generated}}. \quad (1)$$

A lower throw rate is indicative of a more robust and efficient offloading strategy.

Appendix A. Example Appendix Section

Appendix text.

(Mishra et al., 2023) Example citation, See Mishra et al. (2023).

References

- Aazam, M., Islam, S.U., Lone, S.T., Abbas, A., 2022. Cloud of Things (CoT): Cloud-Fog-IoT Task Offloading for Sustainable Internet of Things. *IEEE Transactions on Sustainable Computing* 7, 87–98. URL: <https://ieeexplore.ieee.org/document/9214500/>, doi:10.1109/TSUSC.2020.3028615.

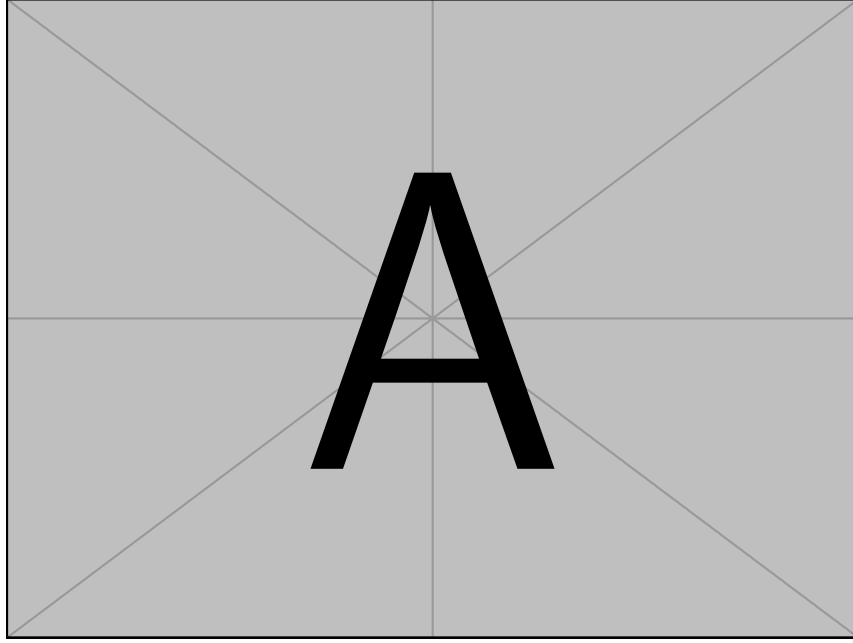


Figure 2: Figure Caption

- Adhikari, M., Mukherjee, M., Srirama, S.N., 2020. DPTO: A Deadline and Priority-Aware Task Offloading in Fog Computing Framework Leveraging Multilevel Feedback Queueing. *IEEE Internet of Things Journal* 7, 5773–5782. URL: <https://ieeexplore.ieee.org/document/8863944/>, doi:10.1109/JIOT.2019.2946426.
- Benaboura, A., Bechar, R., Kadri, W., . A comprehensive survey of task offloading techniques in IoT-Fog-Cloud computing .
- Bernard, L., Yassa, S., Alouache, L., 2024. D-NPGA : a new approach for tasks offloading in fog/cloud environment. URL: <https://ieeexplore.ieee.org/abstract/document/10708605/authors>, doi:10.1109/CoDIT62066.2024.10708605. ISSN: 2576-3555.
- Bukhari, M.M., Ghazal, T.M., Abbas, S., Khan, M.A., Farooq, U., Wahbah, H., Ahmad, M., Adnan, K.M., 2022. An Intelligent Proposed Model for Task Offloading in Fog-Cloud Collaboration Using Logistics Regression. *Computational Intelligence and Neuroscience* 2022, 1–

25. URL: <https://www.hindawi.com/journals/cin/2022/3606068/>, doi:10.1155/2022/3606068.
- Das, R., Inuwa, M.M., 2023. A review on fog computing: Issues, characteristics, challenges, and potential applications. *Telematics and Informatics Reports* 10, 100049. URL: <https://linkinghub.elsevier.com/retrieve/pii/S2772503023000099>, doi:10.1016/j.teler.2023.100049.
- Fahimullah, M., Ahvar, S., Trocan, M., 2022. A Review of Resource Management in Fog Computing: Machine Learning Perspective. URL: <http://arxiv.org/abs/2209.03066>, doi:10.48550/arXiv.2209.03066. arXiv:2209.03066 [cs].
- Gholipour, N., Assuncao, M.D.d., Agarwal, P., gascon samson, j., Buyya, R., 2023. TPTO: A Transformer-PPO based Task Offloading Solution for Edge Computing Environments. URL: <http://arxiv.org/abs/2312.11739>, doi:10.48550/arXiv.2312.11739. arXiv:2312.11739 [cs].
- Guo, L., Lin, J., Xu, X., Li, P., 2024. Algorithmics and Complexity of Cost-Driven Task Offloading with Submodular Optimization in Edge-Cloud Environments. URL: <http://arxiv.org/abs/2411.15687>, doi:10.48550/arXiv.2411.15687. arXiv:2411.15687 [cs] version: 1.
- Iftikhar, S., Gill, S.S., Song, C., Xu, M., Aslanpour, M.S., Toosi, A.N., Du, J., Wu, H., Ghosh, S., Chowdhury, D., Golec, M., Kumar, M., Abdelmoniem, A.M., Cuadrado, F., Varghese, B., Rana, O., Dustdar, S., Uhlig, S., 2023. AI-based Fog and Edge Computing: A Systematic Review, Taxonomy and Future Directions. *Internet of Things* 21, 100674. URL: <http://arxiv.org/abs/2212.04645>, doi:10.1016/j.iot.2022.100674. arXiv:2212.04645 [cs].
- Jazayeri, F., Shahidinejad, A., Ghobaei-Arani, M., 2021. Autonomous computation offloading and auto-scaling the in the mobile fog computing: a deep reinforcement learning-based approach. *Journal of Ambient Intelligence and Humanized Computing* 12, 8265–8284. URL: <https://link.springer.com/10.1007/s12652-020-02561-3>, doi:10.1007/s12652-020-02561-3.

- Jiang, F., Ma, R., Gao, Y., Gu, Z., 2021. A reinforcement learning-based computing offloading and resource allocation scheme in F-RAN. *EURASIP Journal on Advances in Signal Processing* 2021, 91. URL: <https://asp-urasipjournals.springeropen.com/articles/10.1186/s13634-021-00802-x>, doi:10.1186/s13634-021-00802-x.
- Jin, X., Zhang, S., Ding, Y., Wang, Z., 2024. Task offloading for multi-server edge computing in industrial Internet with joint load balance and fuzzy security. *Scientific Reports* 14, 27813. URL: <https://www.nature.com/articles/s41598-024-79464-2>, doi:10.1038/s41598-024-79464-2. publisher: Nature Publishing Group.
- Mishra, K., Rajareddy, G.N.V., Ghugar, U., Chhabra, G.S., Gandomi, A.H., 2023. A Collaborative Computation and Offloading for Compute-Intensive and Latency-Sensitive Dependency-Aware Tasks in Dew-Enabled Vehicular Fog Computing: A Federated Deep Q-Learning Approach. *IEEE Transactions on Network and Service Management* 20, 4600–4614. URL: <https://ieeexplore.ieee.org/document/10143987/>, doi:10.1109/TNSM.2023.3282795.
- Pakmehr, A., Gholipour, M., Zeinali, E., 2024. ETFC: Energy-efficient and deadline-aware task scheduling in fog computing. *Sustainable Computing: Informatics and Systems* 43, 100988. URL: <https://linkinghub.elsevier.com/retrieve/pii/S2210537924000337>, doi:10.1016/j.suscom.2024.100988.
- Rahmani, A.M., Haider, A., Khoshvaght, P., Gharehchopogh, F.S., Moghaddasi, K., Rajabi, S., Hosseinzadeh, M., 2025. Optimizing task offloading with metaheuristic algorithms across cloud, fog, and edge computing networks: A comprehensive survey and state-of-the-art schemes. *Sustainable Computing: Informatics and Systems* 45, 101080. URL: <https://linkinghub.elsevier.com/retrieve/pii/S2210537924001252>, doi:10.1016/j.suscom.2024.101080.
- Ren, Y., Sun, Y., Peng, M., 2021. Deep Reinforcement Learning Based Computation Offloading in Fog Enabled Industrial Internet of Things. *IEEE Transactions on Industrial Informatics* 17, 4978–4987. URL: <https://ieeexplore.ieee.org/document/9183960/>, doi:10.1109/TII.2020.3021024.

- Sarkar, I., Kumar, S., 2022. Deep learning-based energy-efficient computational offloading strategy in heterogeneous fog computing networks. *The Journal of Supercomputing* 78, 15089–15106. URL: <https://doi.org/10.1007/s11227-022-04461-z>, doi:10.1007/s11227-022-04461-z.
- Suryadevara, N.K., 2021. Energy and latency reductions at the fog gateway using a machine learning classifier. *Sustainable Computing: Informatics and Systems* 31, 100582. URL: <https://linkinghub.elsevier.com/retrieve/pii/S2210537921000731>, doi:10.1016/j.suscom.2021.100582.
- Tu, Y., Chen, H., Yan, L., Zhou, X., 2022. Task Offloading Based on LSTM Prediction and Deep Reinforcement Learning for Efficient Edge Computing in IoT. *Future Internet* 14, 30. URL: <https://www.mdpi.com/1999-5903/14/2/30>, doi:10.3390/fi14020030.
- Wiesner, P., Thamsen, L., 2021. LEAF: Simulating large energy-aware fog computing environments, in: 2021 IEEE 5th International Conference on Fog and Edge Computing (ICFEC), pp. 29–36. doi:10.1109/ICFEC51620.2021.00012.
- Yan, J., Bi, S., Zhang, Y.J., Tao, M., 2020. Optimal Task Offloading and Resource Allocation in Mobile-Edge Computing With Inter-User Task Dependency. *IEEE Transactions on Wireless Communications* 19, 235–250. URL: <https://ieeexplore.ieee.org/document/8854339/>, doi:10.1109/TWC.2019.2943563.
- Zhang, R., Chu, X., Ma, R., Zhang, M., Lin, L., Gao, H., Guan, H., 2022. Osttd: Offloading of splittable tasks with topological dependence in multi-tier computing networks. *IEEE Journal on Selected Areas in Communications* .
- Zhang, S., Yi, N., Ma, Y., 2024. A Survey of Computation Offloading with Task Type. URL: <http://arxiv.org/abs/2401.01017>, doi:10.48550/arXiv.2401.01017. arXiv:2401.01017 [cs] version: 3.