NTU Journal of Engineering and Technology (2026) 5 (2) : 76-85
DOI: https://doi.org/10.56286/ntujet.v5i2

IRAQI Academic Scientific Journals

P-ISSN: 2788-9971 E-ISSN: 2788-998X
NTU Journal of Engineering and Technology
Available online at: https://journals.ntu.edu.iq/index.php/NTU-JET/index



Enhancing IoT Communication Efficiency Through a Vehicular Fog Resource Schedule

Raghad Ghalib Alsultan
Department of Electrical Techniques, Polytechnic College-Mosul,
Northern Technical University, Mosul, Iraq
raghad_alsultan@ntu.edu.iq

Article Information

Received: 23-02- 2026,
Revised: 27-03-2026,
Accepted: 01-04-2026,
Published online: 28-06-2026

Corresponding author:
Name:Raghad Ghalib Alsultan
Affiliation: Northern Technical University
Email:
raghad_alsultan@ntu.edu.iq

Key Words:
Internet of Things (IOT),
Fog computing,
Vehicular fog,
Cloud computing,
Grey Wolf Optimizer (GWO

A B S T R A C T

The growth of Internet of Things (IoT) devices has revealed significant issues with cloud-based infrastructures, including excessive latency, network congestion, and inefficient resource utilization. Fog computing helps with these challenges by deploying computing resources near IoT endpoints. However, the growing number of IoT workloads means that infrastructure needs to grow in ways that aren't sustainable. This study introduces an intelligent, hierarchical resource management framework that improves existing fog infrastructure without requiring additional hardware installation. A Master Fog (MF) layer handles the distribution of resources between cloud and Vehicular Fog Resources (VFRs) using the Comparative Attributes Algorithm (CAA) for multi-criteria task prioritization and the Grey Wolf Optimizer (GWO) for optimal VFR selection. the proposed framework enables sustainable, responsive, and cost-effective IoT-fog ecosystems, advancing practical solutions for intelligent resource scheduling in distributed computing environments.

THIS IS AN OPEN ACCESS ARTICLE UNDER THE CC BY LICENSE:
https://creativecommons.org/licenses/by/4.0/

1. Introduction

Cloud computing (CC) denotes a contemporary approach to computing that is increasingly recognized as a distributed architectural paradigm designed to facilitate efficient data processing. in the current internet-enabled landscape. By 2030, the technical and intellectual limitations that currently hinder the cloud from seamlessly handling all types of computation are expected to vanish. However, since the ability to perform every operation without limitation still depends on physical constraints such as the speed of light and energy, maintaining local computing capabilities will remain essential [1]. CC's supply of secure, reliable network storage services has economically benefited numerous sectors [2]. The escalating demand for a more dependable physical connection is attributed to the exponential proliferation of networked devices and the heightened dependence on rapid responses from numerous network edge devices [3]. The Cloud Data Center's (CDC) distant location and its management of numerous IoT applications result in long-distance transmission negatively impacting real-time performance in the cloud [4, 5]. Despite the potential of CC, it is not a panacea for the existing issues in the IoT [6]. Prolonged delays in IoT applications may compromise patient safety and lead to transportation network incidents. In 2012, Cisco developed an innovative technique called Fog Computing (FC) to address these challenges [7]. Hyper-virtualized FC links client devices to CDC servers by offering users computation, storage, and resources for communication [8]. IoT applications benefit from the fog's ability to position compute and infrastructure closer to the edge devices where they are necessary [9]. Thereby, being close to terminals and the IoT devices can be useful for services and computations that need real-time processing with very low latency [10].

The relation of the three levels of FC is obvious. The backbone of the network is the Internet of Things (IoT) layer. It connects a variety of devices including mobile phones, vehicles, personal computers and home automation systems [11, 12]. This layer gathers environmental data from the sensors and forwards it to the FC layer through wireless technologies like Wi-Fi, 3G, 4G, or 5G [13].

The second layer, called the FC layer, is composed of networking elements such as switches, routers, access points, gateways, and workstations. This layer is capable of calculations, connection and data storage. The highest level is the cloud infrastructure which consists of high performance cloud servers [14]. Fog devices need more processing power because of their limited capacity. Such devices have challenges in the organization of workflows, tracking of performance indicators, enforcement of policies and maintenance of data security [15]. But a well-designed resource allocation scheme can enhance network performance, reduce processing delays and improve the overall Quality of Service (QoS) [16]. Scheduling enhances the distribution of resources and the efficient use of application CPU time, thereby improving throughput, responsiveness, security, energy efficiency, and cost-effectiveness [17]. The order in which the task scheduler processes jobs is determined by the underlying software and hardware configuration , which can include factors such as priority levels assigned to tasks, the availability of system resources, and the specific algorithms used for scheduling [18]. To give a satisfactory answer to a request, FC looks at and collects data from a finished product, such as an IoT device. It also decides if the calculations should be done in the cloud [19]. FC provides its users storage, processing, memory, and application services, just like the cloud does [20].

Despite the allure of the FC concept, the exponential data growth and corresponding infrastructure demands render its current implementation impractical. According to IDC, it is projected that 30% of all new data will be generated in real time [21]. In light of these limitations, there exists a pressing need for an alternative. A straightforward method to meet this computing demand is to augment the current fog devices or processors with more units [22]. The continued growth in the deployment of fog devices renders the overarching network increasingly complex and difficult to administer [23]. A systematic and precise evaluation of the network enables the identification of certain unoccupied or available fog nodes that may be allocated to fulfill a range of functions for other fog devices. Some fog nodes are in continuous operation [24] whereas others can perform secondary processing activities when the situation requires [25]. Based on prevailing conditions and the time of day, some fog regions remain underutilized, whereas others impose a heavy burden during different operations [26]. It is therefore strongly recommended that existing resource usage be better optimized [27]. Figure 1 depicts the three-tier fog computing (FC) architecture, where vehicles transfer their tasks to fog nodes, which are local computing resources, and overflow traffic is subsequently directed to the cloud, a centralized data storage and processing service.

In FC frameworks, scheduling is recognized as an NP-hard problem that calls for robust optimization strategies capable of minimizing processing delays while fully leveraging available resources. Notwithstanding the substantial strides made in research on scheduling resources in fog computing, existing solutions remain limited in several important respects: DRL methods require a lot of training and can be unstable; NSGA-II doesn't work well in dynamic and

changing environments; and fuzzy clustering doesn't always do a good job of balancing the load across the whole system. Along with that, the scheduling is generally modeled as a single-stage optimization problem in existing frameworks, and the hierarchical resource orchestration among vehicular fog nodes is not adequately considered. This paper aims to fill these gaps by:

1) Suggests a 3-level hierarchical architecture with a Master Fog (MF) layer in charge of resource allocation between cloud infrastructure and Vehicular Fog Resources (VFRs).

2) A Hybrid Scheduling method has been proposed which combines Comparative Attributes Algorithm (CAA) for multi-criteria task prioritization and Grey Wolf Optimizer (GWO) for optimal resources selection.

3) Extensive simulation to demonstrate that the proposed framework leads to better energy.

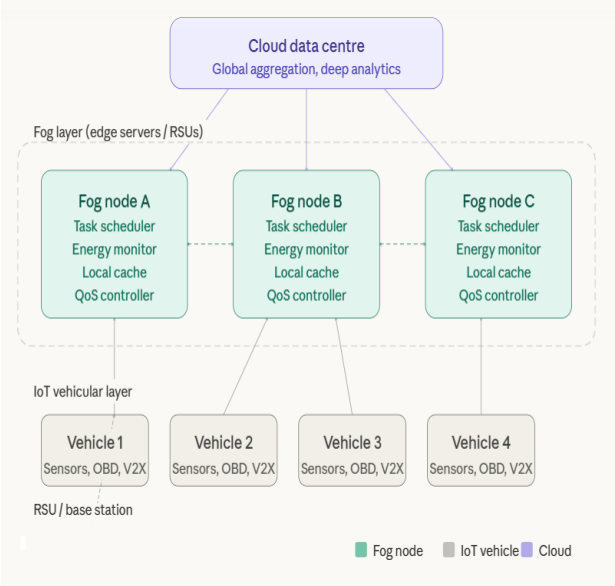


Fig. 1. Three-tier FC architecture: vehicles offload tasks to fog nodes; overflow escalates to cloud.

2. Literature Review

This section presents the methodologies used to handle the distribution of the limited computational resources of the fog layer. The ideas presented in this work are based on and developed from recent results in the relevant fields. However, each notable study either focused exclusively on a single comprehensive application or was grounded in only one of the available technologies. In cloud computing resource scheduling, a DRL-based methodology has been shown to outperform well-established conventional algorithms, namely FCFS, SJF, and RR, in terms of task execution time and resource utilization [32]. Nevertheless, DRL-based approaches are hindered by lengthy training requirements and prone to instability in highly dynamic, rapidly changing environments.

To meet QoS and energy consumption requirements while accounting for the heterogeneity and dynamicity of fog devices, a multi-objective optimization approach for resource scheduling in fog computing using an improved Non-dominated Sorting Genetic Algorithm II (NSGA-II) was proposed [33]. In [28], fog computing resource scheduling based on optimized fuzzy clustering was proposed, targeting service response time reduction and fog node energy efficiency improvement, though it may produce suboptimal global load balancing. In [29], a deep reinforcement learning method is suggested to optimize fog-based IoT system scheduling. The model is trained in a simulation environment and assessed using makespan and energy metrics. In [30], a scheduling method is proposed to reduce time and power required to restart a system following a fog incident, incorporating an Ant Mating Optimization (AMO) method. In [31], the aim is to achieve the best trade-off between system boot time and FC service energy usage. In [32], an improved firework algorithm variant (I-FASC) was developed for dynamic task scheduling.

Ghanavati et al. [33] proposed an energy-aware task scheduling model based on ant-mating optimization (AMO) with enhanced job delivery mechanisms. For time-critical healthcare tasks, a multi-agent fog computing framework (MAFC) was proposed by Mutlag et al. [34], which employed decision tables to manage and optimize task allocation according to priority, network load and resource availability. In [35], a DRL-based scheduling method for fog-based IoT applications was proposed, which shows significant improvement in task execution time and resource efficiency compared with traditional scheduling algorithms. An optimized resource scheduling scheme based on fuzzy clustering had been proposed in [36]. The effectiveness of the scheme in reducing service response time and improving fog node energy efficiency while handling the complexities originating from the uncertainty and heterogeneity had been demonstrated.

Table 1 outlines the main approaches researched for resource scheduling in fog and IoT environments along with their main characteristics and areas of application. Importantly, most of existing work has either covered a single all-inclusive application or centered around specific technological approaches.

Table 1. Comparative Analysis: Methodologies in Fog Resource Scheduling.

Methodology	Core Approach	Key Limitations	How Proposed Method Improves/Differs
Deep Reinforcement Learning (DRL) [32, 34]	Uses trial-and-error learning with neural networks to develop adaptive scheduling policies.	High training overhead; unstable convergence in dynamic environments; requires extensive hyperparameter tuning.	Replaces the complex DRL training phase with a lightweight Grey Wolf Optimizer (GWO) that requires no pre-training, offers faster decision-making, and maintains stability in volatile VFR networks.
NSGA-II [33]	Evolutionary multi-objective optimization using Pareto sorting to handle conflicting objectives.	Scalability issues with large problem instances; slow convergence for real-time scheduling; computationally intensive.	Employs a hierarchical filtering strategy: CAA handles multi-criteria prioritization upfront, allowing the single-objective GWO to optimize efficiently without NSGA-II's computational burden.
Fuzzy Clustering-based Scheduling [10]	Groups fog nodes into clusters using fuzzy logic to manage heterogeneity and uncertainty.	Often leads to local suboptimal solutions; limited global load balancing; performance sensitive to membership function design.	Introduces a global Master Fog (MF) orchestrator that maintains system-wide resource visibility, enabling optimal VFR selection across the entire network rather than within isolated clusters.
Proposed MF-CAA-GWO Framework	Hierarchical 3-tier architecture with hybrid CAA prioritization + GWO-based resource selection.	The MF layer introduces a potential single point of failure; it requires reliable VFR state updates.	Novel integration of architectural orchestration (MF) with algorithmic hybridization (CAA+GWO) to simultaneously address energy efficiency, latency reduction, and global resource utilization.

3. Methods and Materials

3.1 Basic Model

To maximize the use of current fog sources without developing new ones, a Dynamic Scheduling in Fog Computing (SDFC) Architecture is proposed. The Master Fog (MF) layer is positioned between the cloud and the Vehicular Fog Resource (VFR). The Master Fog (MF) layer is situated between the cloud and the Vehicular Fog Resource (VFR). The MF has to control if the tasks are processed in VFRs or in the cloud. The jobs are evaluated using the Comparative Attributes Algorithm (CAA) and the most readily available VFR with best computational capabilities is selected using a meta-heuristic Grey Wolf Optimizer (GWO) approach. The outcome of the test can be exploited to assist in strategic planning between fog and cloud especially in optimizing the resource allocation and improving the overall system efficiency.

3.2 Proposed Methodology

3.2.1 System architecture

In this section, we describe the main components of the proposed paradigm that enable intelligent scheduling in fog computing networks.

The model is an extension of the basic SDFC architecture, by adding an extra layer of vehicular fog between the cloud and the required fog devices. The three-tiered design acknowledges and retains the complexities introduced by fog and cloud cover [35].

The context is monitored, and requests to use the MF are queued. They are isolated from each other and the other fog layers to encourage disjoint interaction between the end devices. The study found that better organizational practices between cloud and fog can lead to better resource utilization and faster response times, as well as lower request failure rates. Figure 2 illustrates the four-tier architecture of the intelligent scheduling paradigm and the communication links between them.

This paradigm not only simplifies the operations but also ensures smooth communication between the different elements of the system. The cloud tier at the top offers the policy backbone: long term storage, global optimization feedback, and an overflow compute sink for tasks that collectively outgrow VFR capacity.

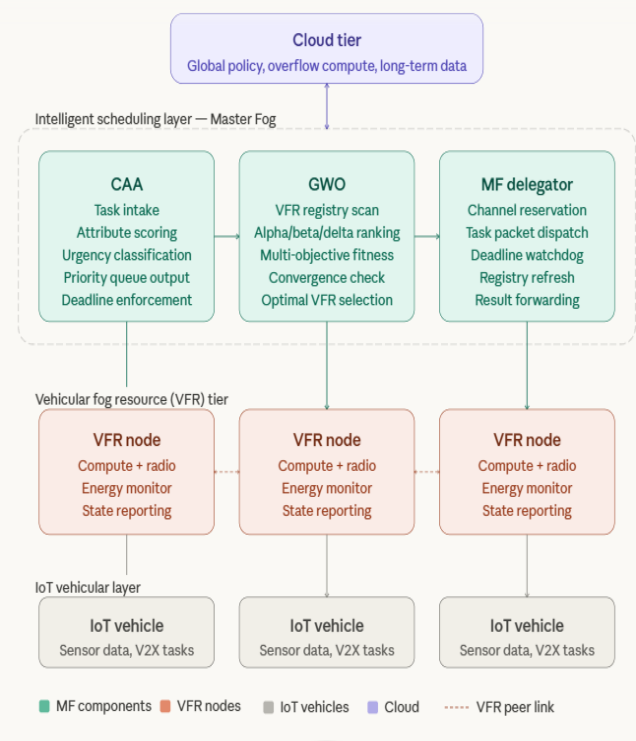


Fig 2: Intelligent Scheduling System architecture diagram

3.2.2 mathematical formulation of the scheduling problem

Several enhancements and modifications to the scheduler's structure were made to achieve the main goal of resource scheduling in FC. Its main duties include accepting and scheduling jobs from VFRs, monitoring VFR resource availability, choosing the optimal VFR or cloud to complete a task, and reporting back to the original VFR with the task's execution results. Sorting nodes and allocating appropriate devices to data using the GWO method was also greatly enhanced. The GWO algorithm represents one of the innovative swarm-intelligence-based approaches; it is a hierarchical metaheuristic technique that draws inspiration from the collaborative hunting tactics of grey wolves [36].

The GWO algorithm draws inspiration from the collective hunting behaviour of grey wolves, who typically gather in groups of 5–12 [37]. Wolves live in packs; the alpha (α) wolves are the leaders, overseeing hunting, shelter, planning, and movement. The beta wolves serve as second-in-command, assisting in decision-making. Wolves ranked as omega occupy the lowest position. Any wolf not categorized as alpha, beta, or omega is classified as a delta wolf. The wolf's location is represented by the continuous vector

The GWO algorithm enhances job allocation and resource assignment. Assume m VFRs and n tasks. The wolf's location is represented by the continuous vector $W_i(W_{i1}, W_{i2}, \ldots W_{im})$. The i-th wolf's values are transformed to the P_i permutation vector. Eq. (1) assigns tasks to resources:

$$p(i,k) = mod(s(i),k,m) \quad (1)$$

Equations (2) and (3) describe the encircling action [38]:

$$\vec{DS} = \text{Mod}(\vec{N} \cdot \vec{X}\text{pry}(t) - \vec{X}w(t)) \quad (2)$$

$$\vec{X}w(t+1) = \vec{X}\text{pry}(t) - \vec{M} \cdot \vec{DS} \quad (3)$$

where t = iteration number; $\vec{X}$ = vector of the grey wolf's positions; DS = distance between X-Pry(t) and Wolf $X_W(t)$. Multiplier vectors A and C are defined as:

$$\vec{A} = 2\vec{a} \cdot \vec{r_1} - \vec{a} \quad (4)$$

$$\vec{C} = 2 \cdot \vec{r_2} \quad (5)$$

where $\vec{a}$ decreases linearly from 2 to 0, and $\vec{r_1}$, $\vec{r_2}$ are random vectors with values between zero and one. The request evaluator at the VFR node uses the weight function defined in Eq. (6) to assess resource availability before scheduling:

$$W_j = \left|L_j^p - L_j^m\right| / \, U_j^p - U_j^m \quad (6)$$

where L_j^p = available CPU time for the j-th Physical Machine (PM); L_j^m = available memory (RAM) for the j-th PM; U_j^p = processor minutes used during the j-th PM; U_j^m = quantity of RAM used during the j-th PM.

4. Fog Computing And The Internet Of Things (Iot)

In this section, the function that fog infrastructure performs is illustrated across three application domains of significance: Autonomous Vehicle Networks, Intelligent Traffic Signal Infrastructure, and Wireless Sensing and Actuation Networks.

4.1 Autonomous Vehicle Networks

The Autonomous Vehicle implementation presents a complex landscape of connectivity: vehicle-to-vehicle exchanges, vehicle-to-infrastructure connections (Wi-Fi, 3 G, LTE, roadside units, and intelligent traffic signals), and infrastructure-to-infrastructure links. Fog computing exhibits several distinctive affordances that render it particularly conducive to the realization of smart connected vehicle (SCV) ecosystems encompassing in-vehicle entertainment, cybersecurity protocols, intelligent traffic governance, and large-scale vehicular data analytics.

These include wide geographic distribution across metropolitan and corridor networks, mobility awareness and location sensitivity, ultra-low latency, heterogeneous interoperability, and real-time communicative throughput. Collectively, such attributes facilitate the seamless embedding of intelligent communication infrastructures within Internet of Things-enabled transportation architectures, thereby substantively enhancing systemic operational efficiency.

4.2 Intelligent Traffic Signal Infrastructure

An intelligent traffic signal node capable of local communication with a set of detection devices capable of detecting pedestrians and cyclists and at the same time estimate the proximity and velocity of approaching vehicles. The system enables synchronized, continuous traffic flow at interconnected intersections via coordinated interaction with adjacent signal nodes. The smart signal node dynamically broadcasts alert messages to approaching vehicles by exploiting the data collected by the detection mechanisms and dynamically adjusts its timing sequences in real time, thereby effectively reducing the probability of collision and increasing the overall traffic safety.

To maintain seamless and uninterrupted traffic management across the network, any adjustment made to the timing sequence is instantly followed by resynchronization with neighboring signals through the coordination tier of the fog infrastructure. The data gathered by the system is continuously processed and analyzed in real time, allowing for dynamic and precise adjustments to signal timings that reflect the latest traffic patterns and conditions on the ground. Moreover, data consolidated from groups of intelligent traffic signal nodes is routinely transmitted to the cloud layer, where it serves as the basis for extensive long-term analysis and well-informed strategic planning initiatives.

5. Simulation Environments

The proposed smart scheduling framework is organized in three different tiers, namely, the Vehicular Fog Resources (VFR), Master Fog (MF) and Cloud layers. The framework systematically evaluates the capacity of each VFR, and designs its operations to maximize the overall utilization of resources. The intelligent scheduling in FC environment is simulated using the i-FogSim simulator [38]. This platform enables users to generate Tuple tasks through the Sensor class, transmit them to Fog Devices for processing and execution, and subsequently retrieve the resulting outcomes via the Actuator class. The VFR, MF, and Cloud were implemented through the Fog Device class, an object-oriented programming technique.

The intelligent scheduling model's network architecture, which comprises three principal tiers, is illustrated in Figure 3. The architecture places the MF as a permanently available orchestrator with a global, continuously updated registry of every VFR's current CPU load, available memory, radio bandwidth, and energy budget. VFRs themselves are the actual compute and communication endpoints — they could be parked vehicles, roadside units, or dedicated edge servers. IoT vehicles generate tasks and submit them upward; the MF decides where each goes, and results flow back down. Cloud escalation is reserved for tasks that exceed collective VFR capacity or have relaxed deadlines. Each MF is connected to ten VFRs. Another advantage over the SDFC and Smart Scheduling paradigms [40] is that the intelligent scheduling allows MF to make connections with an unlimited number of VFRs, depending on the traffic load. The maximum number of linked nodes allowed by the SDFC and Smart Scheduling paradigms is five.

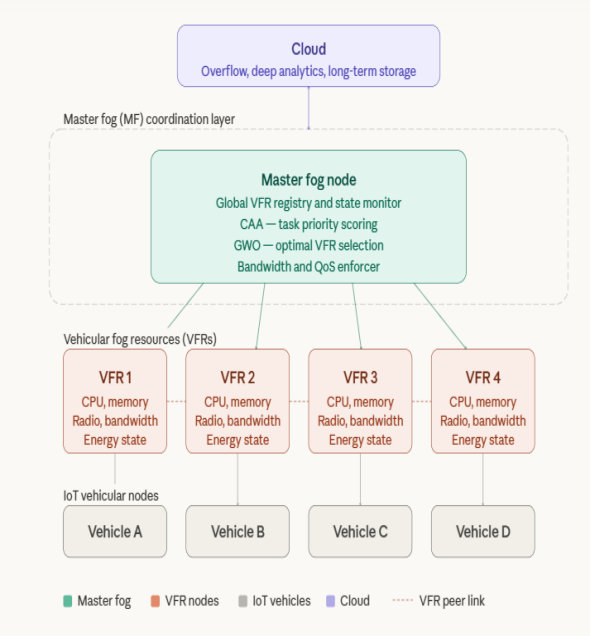


Fig. 3. Network Configuration.

The simulation configurations are presented in Table 2, while Table 3 outlines the system specifications and VFR and MF simulation parameters. A total of 500 distinct workloads were examined throughout the simulation process.

Table 2. Simulation environments.

System Properties	Specification
Operating System	Windows 10 (64-bit)
Processor	Intel® Core™ i7-10510U CPU @ 1.80 GHz / 2.30 GHz
Storage	RAM 8 GB, SSD 256 GB
System Architecture	64-bit, x64
Simulator	iFogSim
Programming Language	Java

Table 3. System requirements.

Specification	Attribute
No. of CF for each MF	5
MIPS	1024 × Random(8, 12)
RAM	512 × Random(1, 3)
UPBW	10000
DOWNBW	270
Busy Power (W)	107.339 – 87053
Idle Power (W)	82.4333 – 82.44
Storage	1000000
Bandwidth	1024 × Random(8, 12)
Architecture	x86
OS	Linux
VMM	Xen
Time-Zone	10.0
Cost	3.0
Cost per Memory	0.05
Cost per Storage	0.001

6. RESULTS AND DISCUSSION

A randomized collection of user workloads and assets is selected to evaluate the proposed approaches. Through the utilization of the intelligent allocation framework, assignments are ranked by priority, and VFRs are screened based on the accessibility of available assets. Upon reception of a new directive by the MF, it is promptly relayed to the most appropriately matched VFR as determined by the specifications.

The proposed methodology incorporates the CAA and GWO algorithms to ensure proper management of designated VFR assets, as shown in Figure 4. The CAA acts as a fast pre-filter: rather than sending every task through an expensive optimization loop, it scores each incoming task on multiple comparative attributes — deadline urgency, payload size, compute intensity, and priority class — and immediately classifies it. The highest-urgency tasks are fast-tracked directly into the GWO with maximum scheduling priority; lower-urgency tasks enter a short batch window to allow the GWO to optimize across a group of tasks simultaneously, reducing per-task overhead.

The GWO then takes over for VFR selection. Inspired by the social hierarchy of grey wolf hunting, it models the search for the best VFR as a convergent pursuit: the alpha wolf represents the current best candidate VFR (highest combined score of available bandwidth, remaining energy, low CPU load, and physical proximity to the requesting vehicle), while beta and delta wolves represent the second and third best candidates. The pack iterates, progressively narrowing the search space until the alpha solution is stable, at which point that VFR receives the task assignment from the MF.

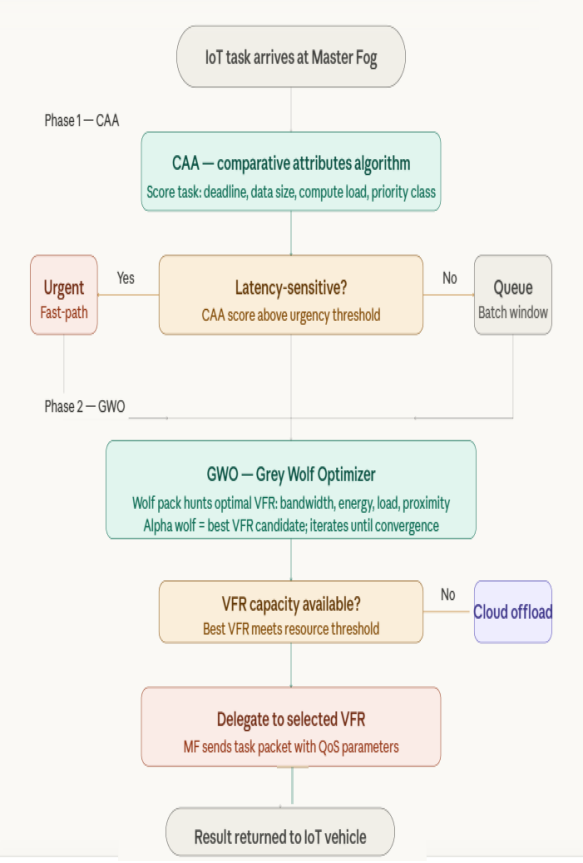


Fig. 4. Combined operation of CAA and GWO algorithms within the MF for scheduling decisions

As shown in Fig. 5, the total power consumption of all approaches increases with workload. Total energy cost is measured from the time the request is registered till the procedure is finished. As shown in the chart, the proposed scheduling technique achieves the lowest energy consumption among SDFC, Smart Scheduling,

and DRL-based models. Figure 6 shows a comparison of the simulation times of the models and shows the degree of improvement of the results of the proposed system, thanks to a number of structural alterations and modifications in the classes of the simulator. Figure 7 compares the tasks completed at the fog layer in the proposed system with the previous system; the proposed model completes more tasks in the fog layer than the SDFC model. Furthermore, the proposed model outperforms the SDFC and Smart Scheduling models in terms of bandwidth utilization.

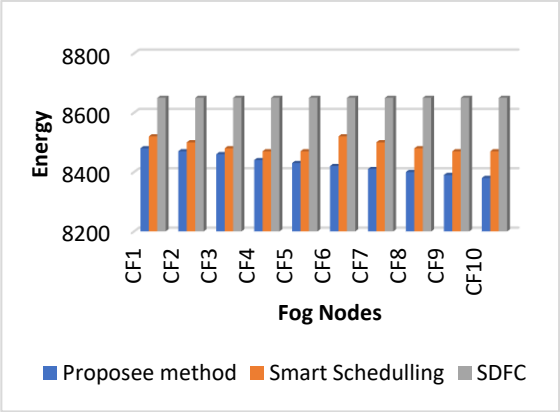


Fig. 5. Comparing energy consumption across scheduling methods

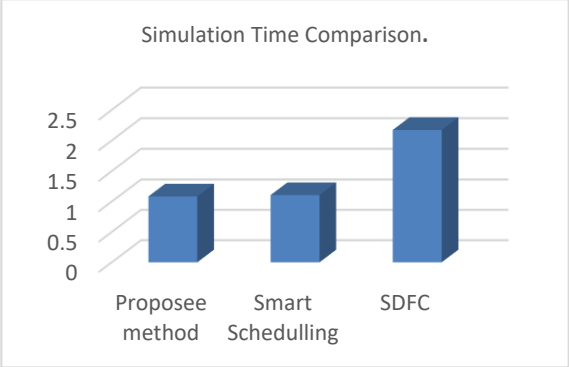


Fig. 6. Quantitative analysis of simulated times

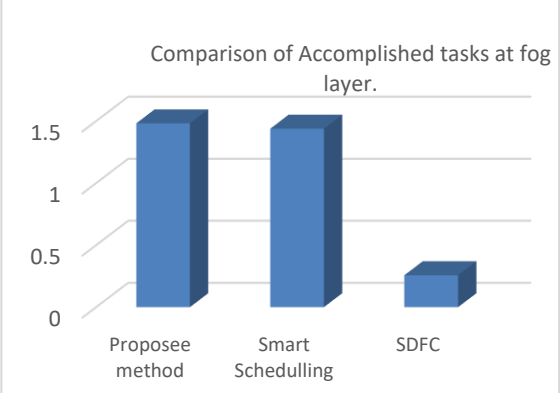


Fig. 7. Tasks completed at similar layers of fog comparison with previous system

7. Conclusion

This study presents a more advanced FC architecture that aims to improve communication and data processing in IoT-driven vehicular networks. The architecture presented in this work is built around three well-defined actors IoT vehicles, VFRs, and a Master Fog coordination layer supported by two principal algorithms, CAA and GWO, which are responsible for managing the greater part of the system's workload. The key novelty that distinguishes this framework from existing approaches lies in the carefully conceived role of the MF layer, which transcends the conventional boundaries of a typical fog node to function instead as a specialized orchestration tier nestled between the VFR mesh and the cloud infrastructure. Acting as the nerve center of the entire system, the MF layer takes charge of resource discovery, intelligent task routing, and the active reduction of scheduling overhead. Without this critical layer in place, each VFR would face the burdensome necessity of broadcasting task requests across the whole fog mesh, an inherently costly and time-consuming process. The MF layer elegantly addresses this shortcoming by offering VFRs a singular and reliable reporting point endowed with a broad global perspective on all available resources, thereby streamlining the scheduling process and enhancing overall system efficiency. This renders the proposed framework an exceptionally well-suited solution for bandwidth-hungry and latency-sensitive IoT applications in the fast-evolving landscape of smart transportation and urban mobility networks.

Acknowledgments

The author declares no specific funding was received for this research.

Author Contributions

R.G. Alsultan: Conceptualization, methodology, software development, simulation, formal analysis, writing original draft preparation, writing review and editing.

Competing Interests

The author declares that there are no competing interests.

References

[1] H. Parekh, H. Cheng, et al., "Understanding the Developments in the Business Perspective of Cloud Computing: A Multidimensional Scaling Analysis," Journal of Organizational and End User Computing, vol. 35, no. 1, pp. 1–36, 2023.

[2] S. A. A. Gillani et al., "Deep Learning in Fog Computing: A Comprehensive Review," IEEE Access, vol. 12, pp. 44567–44592, 2024.

[3] G. Li et al., "Service popularity-based competent resources partitioning for fog computing-enabled industrial Internet of Things," IEEE Internet of Things Journal, vol. 14, no. 10, pp. 4702–4711, Oct. 2018.

[4] R. Basir et al., "Fog computing enabling industrial internet of things: State-of-the-art and research challenges," Sensors, vol. 19, no. 21, p. 4807, Nov. 2019.

[5] S. W. Kareem and H. A. Rebeen, "An Evaluation of Big Data Reduction Approaches," Journal of Applied Computer Science & Mathematics, vol. 15, no. 2, pp. 34–41, 2021.

[6] D. Liu, Z. Yan, W. Ding, and M. Atiquzzaman, "A survey on secure data analytics in edge computing," IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4945–4967, June 2019.

[7] F. U. Khan et al., "Machine Learning-Based Resource Management in Fog Computing: A Systematic Literature Review," Sensors, vol. 25, no. 3, p. 687, 2025.

[8] S. J. C. S. F. Antonio et al., "Cisco delivers vision of fog computing to accelerate value from billions of connected devices," Cisco Systems, San Jose, CA, USA, White Paper, 2014.

[9] S. Ijaz et al., "Energy-efficient time and cost constraint scheduling algorithm using improved multi-objective differential evolution in fog computing," The Journal of Supercomputing, vol. 81, no. 1, p. 81, 2025.

[10] A. V. Dastjerdi and R. Buyya, "Fog Computing: Helping the Internet of Things Realize Its Potential," Computer, vol. 49, no. 8, pp. 112–116, Aug. 2016.

[11] J. Wang and D. J. S. Li, "Task scheduling based on a hybrid heuristic algorithm for smart production line with fog computing," Sensors, vol. 19, no. 5, p. 1023, Mar. 2019.

[12] M. Aazam, S. Zeadally, and K. A. Harras, "Fog Computing Architecture, Evaluation, and Future Research Directions," IEEE Communications Magazine, vol. 56, no. 5, pp. 46–52, May 2018.

[13] T. Choudhari, M. Moh, and T.-S. Moh, "Prioritized task scheduling in fog computing," in Proc. ACMSE Conference, Richmond, KY, USA, 2018, pp. 1–6.

[14] F. Samann, A. Abdulazeez, and S. Askar, "Fog Computing Based on Machine Learning: A Review," Int. J. of Interactive Mobile Technologies, vol. 15, no. 12, pp. 21–35, 2021.

[15] M. M. Mon and M. A. Khine, "Scheduling and load balancing in cloud-fog computing using swarm optimization techniques: A survey," in Proc. Seventeenth Int. Conf. on Computer Applications (ICCA 2019), Yangon, Myanmar, 2019, pp. 78–85.

[16] S. W. Kareem, S. M. Jihad Abdalwahid et al., "A New Efficient Method for Information Security in Hadoop," Qalaai Zanist Scientific Journal, vol. 7, no. 2, pp. 112–128, 2022.

[17] M. M. Mon and M. A. Khine, "Scheduling and load balancing in cloud-fog computing using swarm optimization techniques: A survey," in Proc. Seventeenth Int. Conf. on Computer Applications (ICCA 2019), Yangon, Myanmar, 2019, pp. 78–85.

[18] M. A. Al-Garadi et al., "A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security," IEEE Communications Surveys & Tutorials, vol. 22, no. 3, pp. 1646–1685, 2020.

[19] L. U. Khan, I. Yaqoob, N. H. Tran, Z. Han, and C. S. Hong, "Edge-Computing-Enabled Smart Cities: A Comprehensive Survey," IEEE Internet of Things Journal, vol. 7, no. 10, pp. 10200–10232, Oct. 2020.

[20] M. Cosovic et al., "5G mobile cellular networks: Enabling distributed state estimation for smart grids," IEEE Communications Magazine, vol. 55, no. 10, pp. 62–69, Oct. 2017.

[21] S. Yi et al., "Fog computing: Platform and applications," in Proc. Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb), Washington, DC, USA, 2015, pp. 73–78.

[22] J. K. Samriya et al., "Enhancing IIoT security through blockchain-enabled workload analysis in fog computing environments," Scientific Reports, vol. 15, p. 16542, 2025.

[23] F. Al-Doghman et al., "A review on Fog Computing technology," in Proc. 2016 IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, 2016, pp. 1525–1530.

[24] F. E. F. Samann et al., "IoT provisioning QoS based on cloud and fog computing," Journal of Applied Science and Technology Trends, vol. 2, no. 1, pp. 29–40, 2021.

[25] F. A. Khan et al., "Swarm intelligence-based resource management in fog computing: A comprehensive survey," IEEE Access, vol. 11, pp. 45678–45695, 2023.

[26] Z. Hao et al., "Challenges and software architecture for fog computing," IEEE Internet Computing, vol. 21, no. 2, pp. 44–53, Mar. 2017.

[27] S. Askar, "SDN Based Fog Computing: A Review," Academic Journal of Nawroz University, vol. 10, no. 2, pp. 88–96, 2021.

[28] M. Aazam, S. Zeadally, and K. A. Harras, "Fog computing architecture, evaluation, and future research directions," IEEE Communications Magazine, vol. 56, no. 5, pp. 46–52, May 2018.

[29] H. A. M. Name, F. O. Oladipo, and E. Ariwa, "User mobility and resource scheduling and management in fog computing to support IoT devices," in Proc. 2017 Seventh Int. Conf. on Innovative Computing Technology (INTECH), Luton, UK, 2017, pp. 98–103.

[30] I. Martinez, A. S. Hafid, and A. Jarray, "Design, Resource Management, and Evaluation of Fog Computing Systems: A Survey," IEEE Internet of Things Journal, vol. 8, no. 4, pp. 2494–2516, Feb. 2021.

[31] M. Ghobaei-Arani, A. Souri, and A. A. Rahmanian, "Resource management approaches in fog computing: a comprehensive review," Journal of Grid Computing, vol. 18, no. 1, pp. 1–42, Mar. 2020.

[32] P. Gazori, D. Rahbari, and M. Nickray, "Saving time and cost on the scheduling of fog-based IoT applications using deep reinforcement learning approach," Future Generation Computer Systems, vol. 110, pp. 1098–1115, Sep. 2020.

[33] S. Li, H. Liu, W. Li, and W. Sun, "Non-convex optimization of resource allocation in fog computing using successive approximation," Journal of Systems Science and Complexity, vol. 37, no. 2, pp. 805–840, 2024.

[34] E. M. Ali et al., "Analysis of Deep Reinforcement Learning Algorithms for Task Offloading and Resource Allocation in Fog Computing Environments," Sensors, vol. 25, no. 17, p. 5286, 2025.

[35] S. Ghanavati, J. H. Abawajy, and D. Izadi, "An energy-aware task scheduling model using ant-mating optimization in fog computing environment," IEEE Transactions on Services Computing, vol. 15, no. 3, pp. 1425–1438, May 2022.

[36] A. A. Mutlag et al., "MAFC: Multi-agent fog computing model for healthcare critical tasks management," Sensors, vol. 20, no. 7, p. 1853, Apr. 2020.

[37] E. L. Pimentel, C. A. de Souza, and C. B. Westphall, "Detecting attacks in Fog and cloud computing environments using Deep Learning: A systematic literature review," Computer Networks, vol. 264, p. 111269, 2025.

[38] H. Gupta et al., "iFogSim: A toolkit for modeling and simulating resource management techniques in the Internet of Things, Edge and Fog computing environments," Software: Practice and Experience, vol. 47, no. 9, pp. 1275–1296, Sep. 2017.

[39] B. H. Husain and S. Askar, "Smart Resource Scheduling Model in Fog Computing," in Proc. 2022 8th Int. Engineering Conf. on Sustainable Technology and Development (IEC), Erbil, Iraq, 2022, pp. 96–101. doi: 10.1109/IEC54822.2022.9807469.

[40] L. Chen and M. Liu, "Adaptive DRL for Dynamic Task Offloading in Mobile Edge Computing," IEEE Transactions on Mobile