

Latency-Aware and Energy-Efficient Workload Distribution in Hierarchical Vehicular Fog Computing

Salman Khan

drkhan@gachon.ac.kr

Gachon University

Woong-Kee Loh

Gachon University

Muhammad Babar

Middle East College

Research Article

Keywords:

Posted Date: October 13th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10552058/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Latency-Aware and Energy-Efficient Workload Distribution in Hierarchical Vehicular Fog Computing

Salman Khan*
School of Computing
Gachon University
Sujeong-gu, Seongnam-si,
13120, Republic of Korea
Corresponding author:
drkhan@gachon.ac.kr

Woong-Kee Loh
School of Computing
Gachon University
Sujeong-gu, Seongnam-si,
13120, Republic of Korea
Email: wkloh2@gachon.ac.kr

Muhammad Babar
Department of Computing
and Electronics Engineering
Middle East College,
Muscat 113, Oman
Email: babarm@mec.edu.om

Abstract—The rapid evolution and growing complexity of the modern automotive sector, along with increasing interconnectivity among vehicular systems, have sparked significant academic and industrial interest. Emerging intelligent transportation systems demand computational architectures capable of achieving ultra-low latency and high processing efficiency. However, these performance requirements are constrained by high energy consumption and system wide power limitations. To mitigate these challenges, intelligent workload distribution specifically, adaptive task allocation and optimal utilization of decentralized computing resources offers a scalable and energy efficient solution. This study proposes an advanced framework for resource provisioning and task offloading, embedded within a hierarchical tri-tier vehicular fog-cloud infrastructure. In this architecture, vehicles function as mobile IoT nodes, and fog computing acts as an intermediate service layer to reduce latency between edge devices and centralized cloud data centers. The proposed system supports reliable, real-time task processing and dynamic resource allocation, tailored for latency-critical applications in fast changing vehicular environments. To enhance the efficiency of the fog layer, a meta-heuristic optimization algorithm is introduced to maximize resource usage and reduce end-to-end communication delays. The framework is implemented and evaluated using iFogSim, a specialized simulation tool for fog computing research. Experimental results confirm the proposed MPSO reduces average latency by 23.8%, energy consumption by 22.1%, execution time by 32.4%, and convergence iterations by 34.9% compared with conventional PSO and other heuristic algorithms. It also improves resource utilization by 9.7% and task success rate by 4.5%. Overall, these results confirm that MPSO provides faster convergence, lower delay, reduced energy usage and more efficient fog resource utilization for large-scale vehicular fog-cloud computing.

I. INTRODUCTION

Modern real-time applications needs an efficient and quick response from the processing elements. The delay in response is unrealistic and unacceptable for end user applications. With the rapid growth of IoT applications in the field of transportation, autonomous vehicles is being considered as one of the major advancement in the field of information and communication technology. Adhoc vehicles connected through the network by means of communication protocols and acquire

the data from the surroundings through sensors. Onboard processing units installed in vehicles are used for object detection, parking allotment and path finding. However, the increase in the vehicles industry brings immense challenges in recent as well as for future aspect. How can the large bulk of data that needs quick computation and storage purpose if the onboard units are not capable to react and communicate further. For a quick response time means the computation time and less energy consumption of processing units are the potential problems to address. Cloud computing is one of the solution among many for the above mentioned problems which provides sufficient computation and massive storage for long term analytics. However, due to multi-hop distance from the end user IoT applications cloud provides significant delay, and is unacceptable for the delay-sensitive applications in the case of vehicular computation [1]. The best and the alternate solution to the cloud is the fog computing where computation, storage and communication over the network are performed very close to the end user. The basic aim of fog computing is to provide resources with minimal and low-cost latency for real-time applications. However, the increase in the number of IoT devices makes it more complicated. It has been estimated that due the increase of 'Things' connected to the Internet worldwide, which will produce 73 ZB of data by 2025 [2]. This increasing demand in request have forced the academia and industries to think and provide a possible solutions for real-time applications. Secondly, fog are resource constraint devices unlike cloud. The devices have limited capability to process and store the information with minimal delay. To fully optimize the performance of fog computing devices, there is a proper need of resource management as a premier layer in any computing paradigm [3]. Resource allocation, task offloading and load balancing are techniques which can maximize the efficient utilization of fog resources to facilitate the incoming requests and minimize the delay for critical vehicular applications [4].

It is important to underscore that the fundamental attributes characterizing Vehicular Fog Computing (VFC) exhibit sub-

TABLE I
DISTINCTION BETWEEN VCC AND VFC

Aspects	VCC	VFC
Location	Remotely	RSUs, OBU's
Latency	High latency	Low latency
Processing Speed	Slower	Faster
Network Usage	High	Low
Reliability	Low	High
Mobility Support	Low	High
Scalability	High	Low
Security & Privacy	Centralized	High privacy

stantial divergence from those inherent to Vehicular Cloud Computing (VCC), reflecting distinct architectural paradigms, operational constraints, and deployment contexts as shown in Table I.

In order to remediate the aforementioned challenges inherent to fog computing environments characterized by resource-constrained edge devices, we delineate a sophisticated resource orchestration and task offloading paradigm, underpinned by a Modified Particle Swarm Optimization (MPSO) algorithm, specifically tailored for latency-sensitive vehicular applications. The proposed framework is meticulously engineered to achieve near optimal resource utilization to mitigate computational overhead across fog nodes and concurrently minimize the latency, energy dissipation and execution time. The following contributions of this research work are enumerated as follows:

- A novel tri-layered architectural paradigm encompassing Cloud, Fog and IoT layers has been proposed designed to furnish robust services and comprehensive support for real-time vehicular applications.
- We put forward an advanced resource allocation and task offloading strategy aimed at optimizing the utilization of fog computing resources. This approach is underpinned by a Modified Particle Swarm Optimization (MPSO) algorithm tailored to achieve accelerated convergence, reduced latency, energy, execution cost and maximized task success rate thereby ensuring efficient and responsive system performance in dynamic computing environments.
- An exhaustive analytical and empirical evaluation of the proposed methodology was conducted through comprehensive simulation paradigms. The obtained results evidence unequivocally corroborates the preeminence of the proposed approach over extant comparative methodologies, particularly in terms of attenuated response latency and diminished energy expenditure, thereby underscoring its efficacy and operational superiority within the designated computational context.

The remainder of this manuscript is structured as follows: Section 2 offers relevant literature studies delineating foundational concepts while identifying prevailing research gaps in alignment with the scope of this investigation. Section 3 elaborates on the layering architectural framework model and explicates the constituent operational processes. Section 4 describe the proposed resource allocation and task offloading

challenges inherent to fog computing environments. Section 5 articulates the synthesis of the architectural framework and formalizes the multi-objective optimization model. Section 6 details the experimental configuration, encompassing both the simulation environment and implementation parameters. Section 7 presents the empirical findings. Finally, Section 8 encapsulates the study conclusions and outlines prospective future research directions.

II. RELATED WORKS

Resource management has persistently remained a focal point of scholarly inquiry due to its critical importance in the evolution of contemporary computing paradigms [5]. In any computational framework, the strategic orchestration of resources serves as a cornerstone for augmenting system wide performance and operational efficiency [6]. Within the realm of cloud computing, the abundance of computational power renders the challenges of resource allocation and task scheduling comparatively trivial, with the predominant bottleneck arising from communication latency between centralized data centers and peripheral user devices. This latency-induced inefficiency has catalyzed the emergence of an alternative architectural paradigm fog computing.

Fog computing extends the capabilities of cloud infrastructures by decentralizing core services like computation, storage, and networking and provisioning them in closer proximity to the edge of the network [7]. This paradigm shift is particularly advantageous for latency-sensitive applications. However, unlike the cloud robust infrastructure, fog environments are inherently constrained in terms of computational, storage, and energy resources.

Consequently, the processes of resource allocation, scheduling, and load balancing in fog computing present significantly heightened complexities. These challenges underscore the necessity for the development and deployment of sophisticated optimization techniques aimed at maximizing resource utilization and minimizing overhead-induced latency, particularly in the context of real-time Internet of Things (IoT) applications [8].

Shojafar *et al.* [9] This paper presents an energy-efficient adaptive resource scheduler for Networked Fog Centers (Net-FCs) operating at the vehicular network edge. Connected to Vehicular Clients (VCs) via single-hop I2V TCP/IP links, the scheduler utilizes local TCP/IP state measurements to optimize communication and computation energy efficiency while satisfying strict QoS constraints on minimum transmission rates, maximum delays, and delay jitter. Husnain *et al.* [10] proposed the Workflow Scheduling and Resource Allocation (WSRA) problem is formulated as a multi-objective optimization task. A hybrid GWO-MFO algorithm is proposed to minimize latency and energy consumption while satisfying deadline and reliability constraints. Li *et al.* [11] proposed an adaptive wireless resource allocation strategy for computation offloading in a three-layer vehicular edge cloud framework. The process is modeled at the wireless resource block level to suit dynamic vehicular scenarios and align with 5G evolution. A

value density function is introduced to evaluate resource cost-effectiveness, achieving its maximum when offloading energy consumption equals half of local energy consumption. The authors proposed an optimized task offloading process in the VFC system in terms of latency and energy objectives while taking the deadline constraint into considerations by adopting a Multi-Objective Evolutionary Algorithm (MOEA). Four different execution/transmission models are proposed where vehicle resources are utilized for tasks execution and transmission, and the well-known Dijkstra's algorithm is adopted to find the minimum path between each two nodes [12]. Chiang *et al.* [13] proposed a Vehicular Ad Hoc Networks (VANETs) framework which enable vehicles to form on-board sensor networks and exchange traffic information. However, this increases storage and processing demands, and transmission latency between devices and data centers remains a key challenge. Peng *et al.* [14] proposed a multi-attribute double auction mechanism for VFC, considering both price and non-price factors for optimal matching. It is the first to apply such an approach in VFC, ensuring computational efficiency, individual rationality, budget balance, and truthfulness. Saleem *et al.* [15] proposed a jammed vehicular cloudlet (JVC) assisted task offloading framework that utilizes idle resources from congested vehicles and roadside units to handle resource-intensive tasks. An incentive mechanism is designed to encourage participation by rewarding JVCs and charging users. A joint task assignment and resource allocation problem is formulated to maximize JVCs total profit under data segmentation, deadline, and budget constraints. Cheng *et al.* [16] proposed a novel method for vehicular applications to optimize fog resource allocation and reduce service latency using parked vehicles. An improved heuristic, Revised Fitness-based Binary Battle Royale Optimizer (RF-BinBRO), is introduced, combined with Deep Adaptive Reinforcement Learning (DARL) to effectively handle resource allocation and vehicle parking dynamics. Behura *et al.* [17] proposed a comprehensive evaluation of 118 advanced VANET routing protocols, including topology-based, geographic, cluster based, and AI-enhanced methods, as well as secure routing via cryptography and blockchain. Despite many proposed techniques, a systematic real-world performance review has been lacking. The protocols are compared based on key QoS metrics like latency, reliability, scalability, and security. Salonikias *et al.* [18] analyzed the operational aspects of an ITS paradigm using fog computing and highlights related access control challenges. To address these, it proposes key elements of an attribute-based access control scheme tailored for fog environments, laying the groundwork for future refinement and validation. Jamil *et al.* [19] proposed IRATS, a PPO-based intelligent, priority and deadline-aware algorithm for online and distributed resource allocation in vehicular fog networks. It models the problem as a Markov decision process to reduce task waiting time and delay, using multi level queues to schedule tasks on vehicles sharing idle resources based on priority. Wu *et al.* [20] proposed an optimize joint task assignment and resource allocation using mobility prediction. Two methods are proposed: (1) split the

problem into subtask assignment and bandwidth allocation, handled by matching and convex optimization; (2) apply an iterative 0-1 ILP approach for higher accuracy.

It is evident from the preceding discourse that resource management constitutes a critical research avenue across computing paradigms. Core components such as resource allocation, task scheduling, load balancing, reliability, and scalability collectively serve to augment overall system performance when strategically addressed through advanced optimization methodologies. Nevertheless, a substantial portion of existing literature and implemented techniques often overlook the nuanced demands of real-time applications within these paradigms. In response to this gap, we introduce a sophisticated resource provisioning and task offloading framework tailored for mobility support autonomous vehicular networks, leveraging a modified Particle Swarm Optimization (PSO) algorithm. The proposed approach is meticulously designed to optimize performance by concurrently minimizing communication latency and energy consumption, thereby aligning system behavior with the stringent requirements of real-time, delay-sensitive vehicular environments. Table II presents a detailed comparative analysis of existing works in the domain of vehicular technology, categorized by key feature sets. It highlights how each approach addresses core functionalities, implementation strategies, and performance aspects, enabling a clear understanding of their relative advantages and limitations.

III. LAYERING ARCHITECTURAL FRAMEWORK

This section describes the layered architecture in IoV systems, which is primarily composed of three interconnected and hierarchical layers:

- 1) Topmost Layer-Cloud Layer
- 2) Middle Layer-Fog Layer
- 3) Bottom Layer-Internet of Vehicles (IoV) Layer

This architecture is designed to enhance data processing efficiency, real-time decision-making, and system scalability by distributing computational tasks across these layers.

A. VFC System Service

The cloud is the topmost layer in the system architectural and is considered the important layer as shown in Fig. 1. It mainly consists of powerful computing devices to perform large and complex operations and large storage devices for storing the data for long-term [24]. Cloud data centers are centralized and geographically distributed over a wide area of network. Due to geographically widespread and multi-hop distance it create communication latency for real-time applications which is unacceptable for critical delay-sensitive applications [25].

The fog is the second most important layer after cloud in the system architecture, where the computation, storage and network resources are provided very close to the IoT users [26]. Fog computing consist of computation nodes which can process the user request without any delay. For real-time applications, quick response time is one of the important

TABLE II
COMPARATIVE ANALYSIS OF RELATED WORKS BY FEATURE SET

Reference	Design Framework	Mobility Support	Resource Allocation	Task Offloading	Real Time	Evolutionary Approach	Empirical Simulation
[9]	Cloud-Fog	✓	✓	×	✓	×	Simulation
[10]	Cloud-Fog-IoT	×	✓	×	✓	✓	Simulation
[11]	Cloud-Fog-IoT	✓	✓	✓	✓	×	Simulation
[13]	Cloud-Fog-IoT	×	×	×	×	×	Empirical
[14]	Fog-IoT	×	✓	×	×	×	Simulation
[15]	Fog-IoT	✓	✓	✓	✓	✓	Simulation
[16]	Cloud-Fog-IoT	×	✓	×	✓	×	Simulation
[18]	Cloud-Fog-IoT	✓	×	×	×	×	Empirical
[19]	Cloud-Fog-IoT	✓	✓	×	✓	×	Simulation
[20]	Cloud-Fog-IoT	✓	✓	×	✓	×	Simulation
[21]	Cloud-Edge-Fog	×	×	×	✓	×	Empirical
[22]	Edge-Fog-IoT	×	✓	×	×	×	Simulation
[23]	Cloud-Edge-Fog	×	×	✓	×	×	Simulation
Proposed work	Cloud-Fog-IoT	✓	✓	✓	✓	✓	Simulation

characteristic for critical applications. For efficient utilization of fog computing, proper resource management is necessary and it also support the resource provisioning, task offloading, balancing the load and reliability, heterogeneity and better quality of services (QoS) [27][28]. Fig. 1 illustrates the importance of this layer that how it effectively and efficiently utilizes the resources to meet the smart vehicles requirements. The main ambition of fog computing is to minimize the latency and overhead response and ensure optimal utilization of resources. Fog computing to be the most promising computing paradigm for any real-time applications. In the case of vehicular computing, fog nodes decide whether to process the vehicle request suddenly or to send to the cloud for complex operation. Hence the feature of delay-sensitivity is minimized by the fog computing for autonomous vehicle system.

On the other hand, IoV is the perception layer which consists of sensors, actuators and onboard units. The vehicles are equipped sensors and network communicating devices which makes it capable to communicate with the same vehicle, other vehicle and with the fixed infrastructure. These intelligent vehicles works as an information acquisition and execute the command. The information collected from the surrounding sensors and execute commands accordingly. The main components involved are sensors and actuators which comprises of lidar sensor, radar sensor, proximity sensor. All these sensors gathered information from the surrounding and forwarded to the nearest appropriated fog devices for prompt execution of service. However, there are many challenges associated with IoV which consists of Quality of Service, reliability and scalability of services. In this article, we proposed an effective and efficient combination of Fog-IoV and presents an optimal utilization of fog resources for the reliable and scalable needs of IoV [29].

IV. EFFICIENT RESOURCE ALLOCATION AND TASK OFFLOADING PROBLEM IN FOG-IOV SYSTEM

Efficient Resource allocation and task offloading for internet of vehicles is illustrated in this section. Autonomous vehicles act as a processing node for various activities like

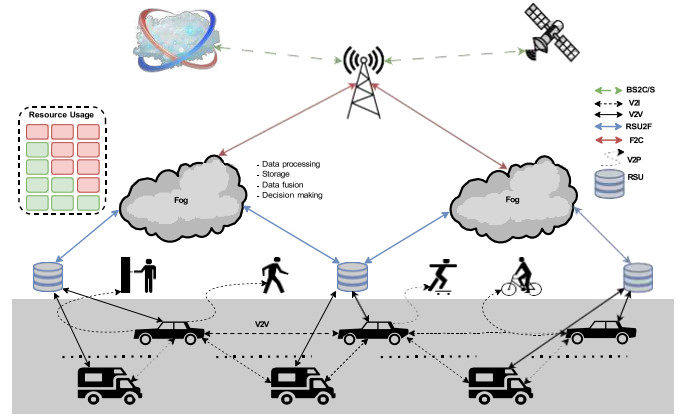


Fig. 1. VFC System Service

image detection, parking detection, and communication with other vehicles and infrastructure. However, the concerned autonomous platform become cumbersome when the massive increase in vehicles and response time become linger on due to the limited resource constraint on-board units installed in vehicles. In order to required a quick response time for real-time vehicle applications, fog resources are suitable solution. In this section, we proposed a novel and efficient resource allocation and task offloading strategy for IoV applications. The better illustration of workload assignment till execution is shown in Fig. 2. Autonomous vehicles often includes various sensors which gather information from the surrounding environment and forward it to further computation devices for quick response time. The above figure shows the RADAR sensor, LiDAR sensor and proximity sensor forward information through access control router. The access control router takes the information as an input flow and forward it to the process buffer which takes the input flows in considerable amount of size. The active workload that is of the task size is forwarded to the fog broker or a dispatcher which further offload these tasks to active fog nodes. These fog nodes consists of virtual machines which allocate resource to the concerned tasks.

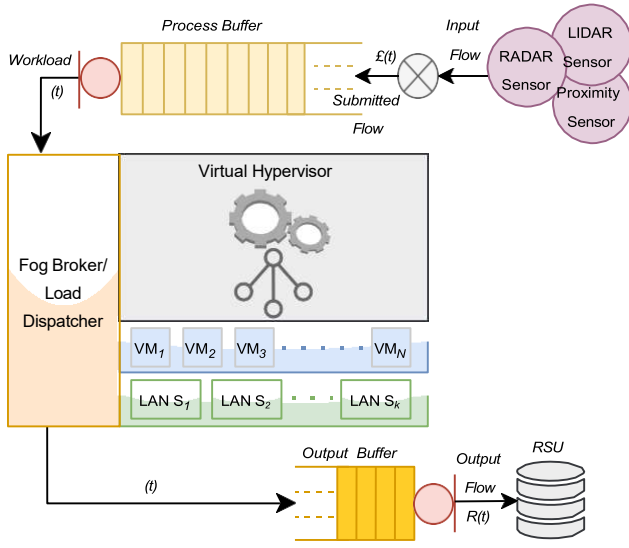


Fig. 2. Optimal resource allocation and task offloading technique

The tasks are performed by the fog nodes in terms of their computation costs and energy consumption. The computing nodes processed the tasks and forward it to the output buffer of finite size. The output information send to the road-side units with minimum response time and maximum throughput for autonomous vehicles to react on the spot. The integration of Fog-IoV structure makes the communication reliable, quick response and less energy execution for real-time autonomous vehicles.

This study introduces a computationally efficient strategy for dynamic resource provisioning and intelligent task offloading, underpinned by a meta heuristic optimization algorithm, specifically tailored to elevate the overall performance efficacy of fog enabled computing environments. The principal objective of the proposed technique is to orchestrate optimal resource utilization across the distributed fog infrastructure, thereby minimizing under utilization of computational assets. Concurrently, it aims to substantially mitigate end to end communication latency and cumulative energy consumption both serving as critical performance indicators through intelligent workload distribution and adaptive scheduling mechanisms, particularly in latency sensitive autonomous vehicular application domain.

V. ARCHITECTURAL SYNTHESIS AND MULTI-OBJECTIVE OPTIMIZATION FRAMEWORK FORMULATION

Fig. 2 represents a fog-assisted vehicular computing model in which heterogeneous sensing data generated by onboard vehicle sensors, such as radar, LiDAR, and proximity sensors, is transformed into computational workloads and managed through a fog broker. The incoming input flow is first placed in a process buffer, where tasks are queued before being submitted to the virtualized fog processing layer. A virtual hypervisor manages the available computing resources by organizing them into virtual machines and LAN segments,

enabling flexible task scheduling, isolation and resource sharing. The fog broker/load dispatcher continuously observes workload intensity, buffer status and resource availability to allocate tasks to suitable virtual machines for execution. After processing, completed tasks are transferred to the output buffer and forwarded as output flow toward the roadside unit or external service infrastructure. Overall, the architecture integrates sensing, buffering, virtualization, scheduling and RSU-based communication to support low-latency, scalable and resource-efficient task execution in vehicular fog computing environments.

A. Optimization Model

The proposed technique underwent rigorous optimization to quantitatively evaluate its efficacy with respect to critical performance metrics, namely latency minimization, energy efficiency, execution cost and maximized resource utilization and task success rate. The underlying algorithm demonstrated a high degree of operational intelligence by strategically orchestrating task scheduling across appropriate fog nodes, thereby enabling optimal execution pathways. This, in turn facilitated the minimization of end-to-end latency, energy consumption and execution cost as performance metrics. The formalized representation of these performance dimensions is encapsulated in the following mathematical formulations.

$$\min_{\mathbf{x} \in \Omega} \mathbf{F}(\mathbf{x}) = L(\mathbf{x}), E(\mathbf{x}), C(\mathbf{x}) \quad (1)$$

$$L(\mathbf{x}) = \sum_{n=1}^N l_n(\mathbf{x}), \quad E(\mathbf{x}) = \sum_{n=1}^N e_n(\mathbf{x}) \quad (2)$$

$$\mathbf{P}^* = \{\mathbf{x}^* \in \Omega \mid \nexists \mathbf{x} \in \Omega : \mathbf{F}(\mathbf{x}) \leq \mathbf{F}(\mathbf{x}^*), \mathbf{F}(\mathbf{x}) \neq \mathbf{F}(\mathbf{x}^*)\} \quad (3)$$

$$\mathbf{x}^{best} = \arg \min_{\mathbf{x} \in \mathbf{P}^*} \alpha \hat{L}(\mathbf{x}) + \beta \hat{E}(\mathbf{x}) + \gamma \hat{C}(\mathbf{x}) \quad (4)$$

$$\alpha + \beta + \gamma = 1, \quad \alpha, \beta, \gamma \geq 0 \quad (5)$$

The above equations define the optimization foundation of the proposed MPSO-based task offloading model. The first equation formulates the problem as a true multi-objective optimization problem, where the aim is to minimize latency, energy consumption and computational cost simultaneously instead of using only a simple weighted sum. This is important because vehicular fog environments involve conflicting objectives; for example, selecting a nearby fog node may reduce latency but may also increase energy use or overload the node. The second equation calculates the total latency and total energy consumption for all generated tasks in the system. These values help MPSO evaluate how efficiently each candidate fog node or offloading decision performs during the search process. The third equation defines the Pareto-optimal solution set, which ensures that the selected solutions are non-dominated, meaning no other solution is better in all

objectives at the same time. This is important for fair trade-off handling among latency, energy and computation cost. The fourth equation selects the final best solution from the Pareto set by using normalized objective values and preference weights. In the proposed MPSO, this step helps choose the most balanced resource allocation decision after local Pbest and global Gbest evaluation. The fifth equation ensures that the weights assigned to latency, energy and computational cost remain valid and balanced. Overall, these equations strengthen the proposed MPSO model by showing that task offloading is performed using Pareto-aware, resource-efficient and trade-off-based optimization rather than simple single-objective minimization. Table III present the possible notations with their relevant meanings.

B. Algorithmic Process

Fig. 3 represents a Modified Particle Swarm Optimization (MPSO)-based task offloading and resource allocation process for a vehicular fog-cloud computing environment. It begins with the generation of application tasks or tuples by vehicular IoT nodes, which are then forwarded to a broker or decision-making entity responsible for selecting the most suitable execution resource. The broker first evaluates whether the combined current load and incoming task load exceed the available processing capacity of the fog host measured in Million Instructions Per Second (MIPS). If the fog host is overloaded, the task is directly forwarded for task execution through an alternative offloading route. Otherwise, the system performs a Best-Fit node search within the neighborhood to identify a suitable nearby fog node. If a feasible node is found, the task is assigned for execution; if not, the optimization process enters the MPSO search phase. In this phase, the algorithm first calculates the personal best position or Pbest of candidate resources within the local vicinity and checks whether the selected resource satisfies the task requirements. If the resource is not suitable, the local search is repeated until a valid Pbest is obtained. Once local feasibility is confirmed, the algorithm calculates the global best (Gbest) position across the wider resource vicinity. The selected Gbest candidate is again verified against resource availability, processing capacity and task constraints. If the global resource verification phase fails, the Gbest search procedure is re-executed; otherwise, the algorithm advances to the fitness evaluation stage in which the candidate allocation is assessed with respect to the defined optimization objectives, including latency reduction, energy efficiency, load balancing and computational capacity. Upon successful completion of the fitness evaluation, the task is deployed to the selected fog resource for execution, and the overall process is subsequently terminated. Overall, the flow diagram illustrates how MPSO combines local neighborhood search, personal best evaluation, global best refinement and resource-aware decision-making to achieve efficient task offloading in a dynamic vehicular fog computing environment.

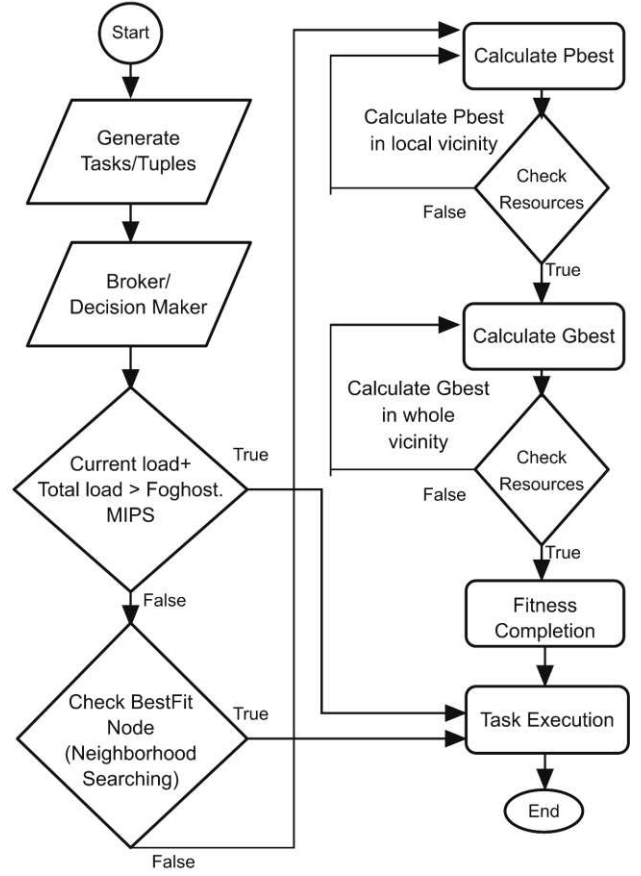


Fig. 3. Flow diagram of proposed algorithm

C. Asymptotic Complexity Characterization of the Proposed Algorithm: A Statistical Perspective

The proposed MPSO algorithm has a per-iteration complexity of $O(P(TN_k + F))$, where P is the swarm size, T is the number of tasks, N_k is the local neighborhood size, and F is the number of fog nodes. Therefore, the total computational complexity over I iterations is expressed as

$$O(I \cdot P(TN_k + F)). \quad (6)$$

In the worst case, when all fog nodes are explored, the complexity becomes

$$O(I \cdot P \cdot T \cdot F), \quad (7)$$

whereas in the best case, when a suitable node is found during the early Best-Fit searching phase, the complexity reduces to

$$O(I \cdot P \cdot T). \quad (8)$$

The memory complexity is given by

$$O(PT + F), \quad (9)$$

Which mainly arises from storing particle positions, velocities, personal best values, global best value, task states and fog-node resource information.

This statistical asymptotic analysis confirms that while the algorithm maintains linear efficiency in the best-case context, its practical execution time gravitates toward a quadratic regime under typical and adverse operating conditions underscoring the algorithms sensitivity to input-dependent branching and iterative constructs.

TABLE III
NOTATION AND MEANING OF THE PROPOSED MULTI-OBJECTIVE MPPO FORMULATION

Symbol	Meaning
$\mathbf{x}$	Candidate offloading/resource allocation solution
Ω	Feasible search space
$\mathbf{F}(\mathbf{x})$	Multi-objective fitness vector
$L(\mathbf{x})$	Total latency
$E(\mathbf{x})$	Total energy consumption
$C(\mathbf{x})$	Computational cost/load
$l_n(\mathbf{x})$	Latency of task n
$e_n(\mathbf{x})$	Energy of task n
N	Total number of tasks
P^*	Pareto-optimal solution set
$\mathbf{x}^*$	Pareto-optimal candidate solution
$\leq$	Pareto dominance relation
$\mathbf{x}^{best}$	Final selected best solution
$\hat{L}(\mathbf{x})$	Normalized latency
$\hat{E}(\mathbf{x})$	Normalized energy consumption
$\hat{C}(\mathbf{x})$	Normalized computational cost
α, β, γ	Objective preference weights
$\alpha + \beta + \gamma = 1$	Weight normalization condition

VI. EXPERIMENTAL SETTINGS AND SIMULATION SETUP

This section presents the simulation environment used to evaluate the proposed MPPO-based resource provisioning and task offloading framework. The evaluation focuses on latency-sensitive vehicular fog-cloud scenarios where efficient task allocation, reduced communication delay and lower energy consumption are essential. The proposed framework is implemented using the iFogSim simulator and its performance is assessed under different workload and fog-node configurations.

A. Simulation Environment

The experimental configuration underpinning the evaluation of the proposed algorithm has been meticulously implemented utilizing the iFogSim simulation toolkit, a specialized environment designed for modeling and analyzing resource management strategies in fog and edge computing infrastructures. This simulation framework is strategically employed to facilitate performance optimization with particular emphasis on minimizing system-wide communication latency and cumulative energy consumption, both of which are designated as the primary quantitative performance metrics.

B. Experimental Settings

Table IV presents the configurations of experimental settings for a latency-critical intelligent transportation scenarios. Vehicles were considered as mobile IoT nodes that generate computational tasks or tuples, while roadside units and fog servers acted as intermediate processing resources. The cloud layer was used for tasks that could not be efficiently processed

Algorithm 1 An Optimized Resource Allocation and Task Offloading algorithm Leveraging a Modified Particle Swarm Optimization

Input Parameters: Task Set T_n where $T_n | n = 1, 2, \dots, m$: A finite collection of computational tasks awaiting scheduling. Fog Server Set F_j where $F_j | j = 1, 2, \dots, k$: A distributed set of fog computing nodes equipped with constrained computational resources.

Output Objective: Latency, Energy consumption and Execution cost.

Initialization Phase:

- 1: $CU_r \leftarrow$ Current devices resource availability vector
- 2: $MU_r \leftarrow$ Maximum permissible resource utilization threshold
- 3: $UT_i^m \leftarrow$ Desired MIPS for each unscheduled task $i \in [1, m]$
- 4: $R_m \leftarrow$ Required MIPS for task execution
- 5: $R_j \leftarrow$ Resource vector set across all fog nodes
- 6: $Local_{best} \leftarrow 0$: Initialization of individual agent's historical best
- 7: $Global_{best} \leftarrow 0$: Initialization of global optimal solution reference
- 8: **Iterative Allocation and Scheduling Logic:**
- 9: **for** each task $i = 1$ to m , **do**
- 10: **if** $CU_r + T_{load} \leq MU_r$ **then**
- 11: Proceed to execute task T_i locally on the originating device
- 12: **else**
- 13: **if** $UT_i^m \leq R_m$ **then**
- 14: Designate fog node $j \leftarrow UT_i$ as a potential offloading candidate
- 15: Migrate and execute task T_i on node j
- 16: **for** each server $j = 1$ to k , **do**
- 17: Select $F_j \leftarrow j$ (least loaded fog node) based on current resource usage
- 18: Update assign $Local_{best} \leftarrow F_j$
- 19: **end for**
- 20: Update assign $Global_{best} \leftarrow Local_{best}$
- 21: Determine next task-to-node mapping: Allocate $task_i \leftarrow j(Global_{best})$
- 22: Execute task T_i on selected fog node
- 23: **end if**
- 24: **end if**
- 25: **end for**

at the fog layer. The proposed MPPO algorithm was applied to determine the most suitable fog node for task execution based on resource availability, processing capacity, latency, and energy cost. The simulation parameters included the number of fog nodes, number of tasks, processing capacity in MIPS, bandwidth, RAM, energy consumption rate, and execution delay. Communication latency and energy consumption were selected as the main performance metrics because they directly represent the responsiveness and efficiency of the proposed offloading framework.

TABLE IV
SIMULATION SETUP

System Parameters	Values
Cloud servers	0–1
Fog servers	1000
No. of vehicles	250
Fog servers processing capability	5600 MIPS
Fog servers memory usage	4000 MB
Cloud server memory	40000 MB
WAN bandwidth	100–10000 MB
LAN bandwidth	10000–10000 MB
Idle & busy power of cloud servers	1648–1332 W
Idle & busy power of fog devices	107–83 W

C. Performance Metrics

The effectiveness of the proposed MPSO algorithm was evaluated using key performance indicators including average latency, energy consumption, execution time, convergence iterations, resource utilization and task success rate. Average latency measures the time required for task transmission, processing and response generation. Energy consumption represents the total energy used by fog and cloud resources during task execution. Execution time indicates the computational time required by each algorithm to produce an offloading decision. Convergence iterations measure how quickly the algorithm reaches an optimal or near-optimal solution. Resource utilization evaluates how efficiently available fog resources are used, while task success rate indicates the percentage of tasks successfully executed within the required constraints.

D. Simulation Objective

The main objective of the simulation was to validate the performance of the proposed MPSO algorithm against conventional heuristic and meta-heuristic algorithms. The simulation aimed to demonstrate that the proposed approach can achieve lower latency, reduced energy consumption, faster convergence and improved resource utilization in large-scale vehicular fog-cloud environments. By combining Best-Fit neighborhood searching, local Pbest evaluation, global Gbest refinement and resource availability checking, the proposed MPSO framework was designed to provide efficient and reliable task offloading for real-time intelligent transportation applications.

VII. RESULT AND DISCUSSION

The proposed algorithmic framework has been subjected to an in-depth analytical and empirical evaluation, wherein its operational efficacy is quantitatively assessed against critical performance metrics, specifically communication latency and energy consumption and execution time analysis. The resultant performance indices, derived from extensive simulation scenarios and are systematically obtained and rigorously placed aside with those produced by a suite of alternative meta-heuristic optimization algorithms. This comparative analysis enables a thorough assessment of the relative advantages of the proposed method in terms of computational efficiency and energy-aware task orchestration within resource-constrained vehicular–fog–cloud environments. A detailed performance

evaluation of the proposed approach is conducted against recent state-of-the-art methodologies as enumerated in the referenced Table V.

A. Average Latency Analysis

Latency represents a critical performance indicator for quantitatively evaluating the operational efficiency of distributed computing paradigms, particularly in the domain of latency-sensitive, real-time Internet of Things (IoT) applications [30]. Owing to the architectural proximity of fog computing resources to end-user devices, such infrastructures are, in principle, capable of delivering significantly lower latency relative to conventional cloud-centric systems.

In this study, we perform a rigorous end-to-end latency characterization of IoT application workflows executed within a fog computing environment. The analysis accounts for the aggregate temporal overhead incurred from the initiation of a task at the IoT node through to the retrieval of the final computational outcome. post-processing at the fog layer. The comparative performance profiling was systematically undertaken to clarify the latency differentials and responsiveness of the proposed model relative to alternative computational frameworks. The latency quantification is formally computed through the following methodological formulation. L_{total}^{ij} =

$$L_{total}^{ij} = \frac{D_i}{B_{ij}} + Q_{ij} + T_{ij}^{prop} + \frac{C_i}{F_j \cdot U_j} + W_j + T_{ij}^{ack} + R_j \cdot \delta$$

Where:

- L_{total}^{ij} : Total communication latency for task i offloaded to fog node j .
- **Offloading Latency Components:**
 - D_i : Data size of task i .
 - B_{ij} : Available bandwidth between IoT device i and fog node j .
 - Q_{ij} : Queuing delay at the network interface.
 - T_{ij}^{prop} : Propagation delay from IoT device i to fog node j .
- **Computation Latency Components:**
 - C_i : Computational complexity of task i .
 - F_j : Processing frequency of fog node j .
 - U_j : Utilization factor of fog node j (ranges from 0 to 1).
 - W_j : Waiting time in fog nodes task execution queue.
- **Transition Latency Components:**
 - T_{ij}^{ack} : Acknowledgment delay in the transmission of execution results.
 - R_j : Result data size generated by fog node j .
 - δ : Transmission delay per unit of result data based on current network conditions.

This formulation yields a multi-dimensional characterization of latency, explicitly incorporating transmission, queuing, propagation, computation and result-return delays. As such, it is well-suited for the simulation and quantitative analysis of performance in real-time, resource-constrained fog computing

TABLE V
EMPIRICAL PERFORMANCE COMPARISON FOR 1000 FOG NODES

Algorithm	Latency (ms)	Energy (J)	Execution (s)	Convergence Iterations	Resource Utilization (%)	Task Success Rate (%)
GA	312.8	684.5	18.7	82	76.4	89.2
ABC	294.6	651.3	16.9	75	78.9	90.6
ACO	281.4	627.8	15.8	71	80.2	91.4
Simulated Annealing	336.9	719.6	21.3	94	73.8	86.7
PSO	257.5	592.4	13.6	63	83.5	93.1
Proposed MPSO	196.2	461.7	9.2	41	91.6	97.3

environments.

To enable a rigorous assessment of the proposed MPSO-based algorithm, an extensive performance evaluation was carried out under three distinct operational regimes: best-case, average-case and worst-case conditions. These regimes were instantiated by exploring a wide range of task-to-fog-device allocation permutations, thereby emulating heterogeneous levels of resource availability and task–resource matching efficiency. In the worst-case regime—representative of an adverse yet practically relevant system behavior—the algorithm repeatedly fails to derive suitable task-to-resource mappings or incurs substantial delays in discovering compatible fog nodes. This situation results in pronounced communication latency, primarily due to extended task scheduling times and the overhead associated with exhaustive resource searches. In contrast, under both the average-case (nominal) and best-case conditions, where resource matching is achieved with higher probability and reduced overhead, the system demonstrates significantly lower response times. These findings underscore the effectiveness of the proposed optimization strategy under favorable or typical operating scenarios.

The communication latency patterns observed across these three regimes are graphically depicted in the corresponding figure, which encapsulates the algorithm performance behavior under varying levels of resource allocation complexity. Fig. 4 presents the latency outcomes for the different framework approaches.

B. Energy Consumption Analysis

To quantitatively evaluate the effectiveness of the proposed technique, energy consumption is adopted as the primary performance metric, as it constitutes a key indicator of operational efficiency and sustainability in fog computing environments for IoT-driven applications [31]. The assessment specifically targets the cumulative energy expenditure of fog devices during the execution of offloaded tasks, thereby enabling a detailed characterization of the system energy-aware behavior under heterogeneous workload conditions [32]. Fig. 5 presents a comparative analysis of the energy consumption associated with different approaches. The computational modeling and estimation of energy consumption are systematically carried out using the analytical formulations introduced in the subsequent equations, which capture the dynamic relationships among task execution, resource utilization and power dissipation within the fog layer.

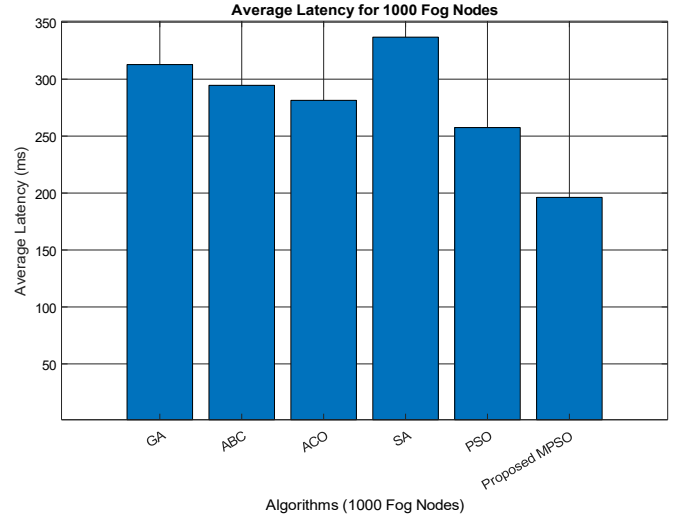


Fig. 4. Average latency analysis for increasing computational nodes

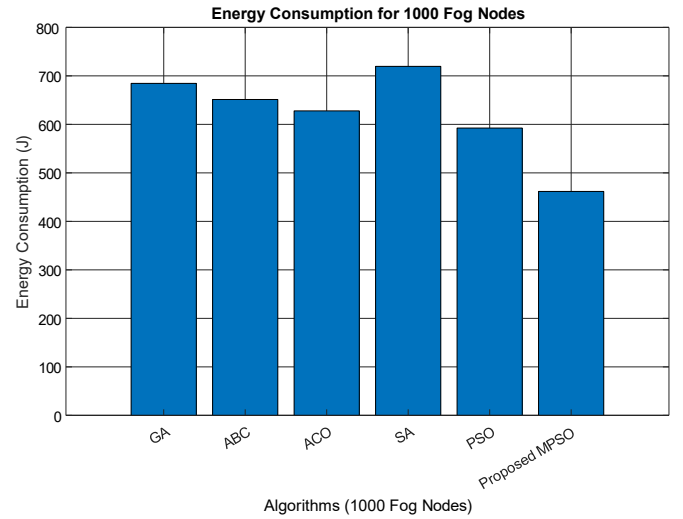


Fig. 5. Energy Efficiency Performance Under Increasing Vehicle Load

$$E_{total}^{i,j} = \sum_{t=1}^T \left(\frac{U_{cpu}^i(t)}{C_{cpu}^j} (P_{max}^j - P_{idle}^j) + P_{idle}^j \right) \Delta t + (\alpha E_{comm}^{i,j} + \beta E_{mem}^{i,j} + \gamma E_{net}^{i,j})$$

Where:

- $E_{total}(i, j)$: Total energy consumed by fog device j for executing task i .
- T : Total execution time of the task in time steps.
- $U_{cpu}^i(t)$: CPU utilization of task i at time step t .
- C_{cpu}^j : Maximum CPU capacity of fog device j .
- P_{max}^j : Maximum power consumption of fog device j under full load.
- P_{idle}^j : Power consumption of fog device j in idle state.
- Δt : Time step duration.
- $E_{comm}^{i,j}$: Communication energy consumed to transmit data between IoT node and fog device j for task i .
- $E_{mem}^{i,j}$: Energy consumed by memory read/write operations for task i .
- $E_{net}^{i,j}$: Energy consumed by networking operations during task offloading and result return.
- α, β, γ : Weighting coefficients representing the relative influence of communication, memory, and networking energy overheads, respectively.

This equation integrates dynamic CPU usage, communication overhead, and auxiliary system operations, offering a comprehensive and fine grained energy consumption model well suited for complex fog-IoT environments.

C. Execution Time Analysis

Execution time measures the computational time required by each algorithm to complete the task allocation and offloading decision process. The proposed MPSO records the lowest execution time of 9.2s, while conventional PSO requires 13.6s, ACO 15.8s, ABC 16.9s, GA 18.7s and Simulated Annealing 21.3s. The reduced execution time of MPSO results from its efficient local neighborhood search, faster convergence and resource validation strategy, which reduce unnecessary exploration of unsuitable fog nodes. This confirms that MPSO is more suitable for real-time vehicular fog computing applications where rapid decision-making is essential. Fig. 6 demonstrate the comparison of various methodologies in terms of execution time analysis.

D. Resource Utilization Analysis

Resource utilization constitutes a pivotal parameter in assessing the ramifications of resource consumption within a fog computing paradigm. Continuous surveillance of resource utilization metrics is imperative given their profound influence on the overarching system performance [33]. Fig. 7 shows the performance of resource utilization for different architectural frameworks. The principal aim of the proposed methodology is to alleviate the computational and communicational load imposed upon fog nodes. This technique achieves a decline in network traffic intensity by ensuring the prompt execution of submitted computational tasks, thereby obviating unnecessary

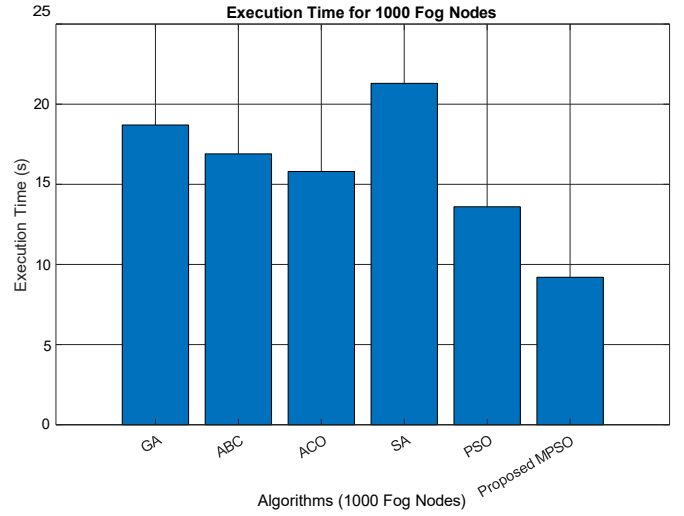


Fig. 6. Execution time analysis for increasing computational nodes

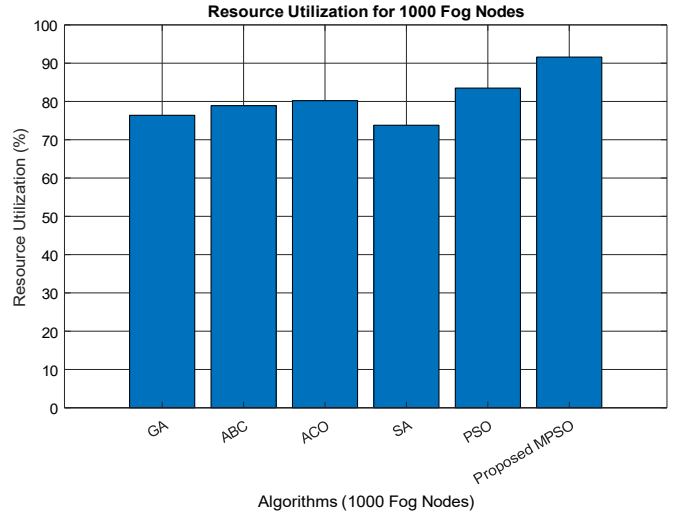


Fig. 7. Resource utilization analysis for increasing fog nodes

transmission delays. The quantification of network bandwidth utilization is rigorously formulated through a corresponding mathematical expression.

$$RU = \frac{\sum_{j=1}^F R_j^{used}}{\sum_{j=1}^F R_j^{total}} \times 100\% \quad (10)$$

where RU represents the resource utilization percentage, R_j^{used} denotes the amount of computational resources used by fog node j , R_j^{total} represents the total available resources of fog node j and F is the total number of fog nodes.

In the proposed MPSO-based framework, this metric measures how efficiently the available fog resources are used during task offloading. Higher resource utilization indicates better allocation of tasks across fog nodes and reduced resource wastage.

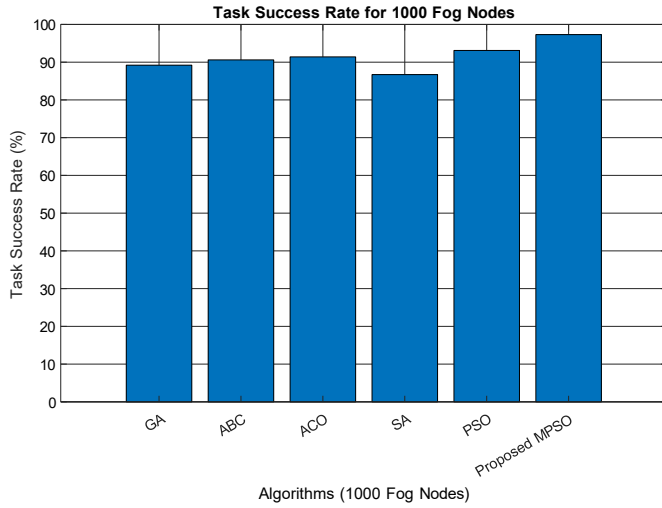


Fig. 8. Task success rate analysis for increasing fog nodes

E. Task Success Rate

Fig. 8 present the task success rate indication wit the percentage of tasks successfully completed within the required latency, resource and execution time constraints. In the 1000 fog-node scenario, the proposed MPSO achieves the highest task success rate of 97.3%, compared with conventional PSO at 93.1%, ACO at 91.4%, ABC at 90.6%, GA at 89.2% and Simulated Annealing at 86.7%. This improvement is mainly due to the resource-aware Best-Fit search and Pbest-Gbest refinement mechanism, which help avoid overloaded fog nodes and assign tasks to more suitable computing resources. Therefore, MPSO provides more reliable task completion in large-scale vehicular fog-cloud environments.

VIII. CONCLUSION

Fog computing acts as a crucial intermediary between cloud systems and IoT endpoints, offering cloud like capabilities closer to the data source to reduce latency and enable real-time responsiveness. Unlike the cloud centralized and resource rich architecture, fog nodes operate under limited computational and energy capacities, making them better suited for latency-sensitive IoT applications. However, the dynamic and heterogeneous nature of fog environments demands adaptive resource management to avoid inefficiencies and service delays. To address this, the study presents an intelligent task offloading and resource provisioning framework, driven by a meta-heuristic optimization algorithm. Designed for autonomous vehicular systems, the proposed solution aims to enhance throughput, balance resource usage and reduce energy consumption and response times.

Furthermore, fog computing presents a fertile ground for the integration of advanced algorithmic paradigms, including but not limited to heuristic and evolutionary techniques, to further enhance the operational efficacy of the hierarchical cloud/fog/IoT computing continuum. As a prospective extension, this research envisions the deployment of machine

learning-driven methodologies within the fog layer to enable real-time, adaptive resource management, thereby unlocking new frontiers in performance optimization and intelligent system orchestration.

Funding Statement: This research received no specific grant from any funding agency in the public, commercial or not for profit sectors.

REFERENCES

- [1] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3860–3873, 2016.
- [2] A. Kishor and C. Chakarbarty, "Task offloading in fog computing for using smart ant colony optimization," *Wireless personal communications*, vol. 127, no. 2, pp. 1683–1704, 2022.
- [3] H. Zhao, S. Wang, and H. Shi, "Fog-computing based mobility and resource management for resilient mobile networks," *High-Confidence Computing*, vol. 4, no. 2, p. 100193, 2024.
- [4] A. R. Hameed, S. ul Islam, I. Ahmad, and K. Munir, "Energy-and performance-aware load-balancing in vehicular fog computing," *Sustainable Computing: Informatics and Systems*, vol. 30, p. 100454, 2021.
- [5] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE communications surveys & tutorials*, vol. 20, no. 1, pp. 416–464, 2017.
- [6] S. Ahmad, S. Khan, F. Jamil, F. Qayyum, A. Ali, and D. Kim, "Design of a general complex problem-solving architecture based on task management and predictive optimization," *International Journal of Distributed Sensor Networks*, vol. 18, no. 6, p. 15501329221107868, 2022.
- [7] R. K. Naha, S. Garg, D. Georgakopoulos, P. P. Jayaraman, L. Gao, Y. Xiang, and R. Ranjan, "Fog computing: Survey of trends, architectures, requirements, and research directions," *IEEE access*, vol. 6, pp. 47 980–48 009, 2018.
- [8] M. Mukherjee, L. Shu, and D. Wang, "Survey of fog computing: Fundamental, network applications, and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1826–1857, 2018.
- [9] M. Shojafar, N. Cordeschi, and E. Baccarelli, "Energy-efficient adaptive resource management for real-time vehicular cloud services," *IEEE Transactions on Cloud computing*, vol. 7, no. 1, pp. 196–209, 2016.
- [10] M. H. Husain, M. Ahmadi, and F. Mardukhi, "A gwo-mfo-based resource allocation in vehicular fog computing with latency constraints and energy reduction," *IEEE Access*, 2024.
- [11] X. Li, Y. Dang, M. Aazam, X. Peng, T. Chen, and C. Chen, "Energy-efficient computation offloading in vehicular edge cloud computing," *IEEE Access*, vol. 8, pp. 37 632–37 644, 2020.

- [12] S. K. Abdullah and A. J. Jabir, "A multi-objective task offloading optimization for vehicular fog computing," *Iraqi Journal of Science*, pp. 785–800, 2022.
- [13] M.-L. Chiang, Y.-a. Lin, H.-C. Hsieh, and W.-C. Tsai, "Improving latency and reliability for vehicle system under fog computing networks," in *Recent Advances in Intelligent Information Hiding and Multimedia Signal Processing: Proceeding of the Fourteenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, November, 26-28, 2018, Sendai, Japan, Volume 1 14*. Springer, 2019, pp. 225–232.
- [14] X. Peng, K. Ota, and M. Dong, "Multiattribute-based double auction toward resource allocation in vehicular fog computing," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3094–3103, 2020.
- [15] U. Saleem, S. Jangsher, T. Li, and Y. Li, "Profit optimized task scheduling for vehicular fog computing," *Wireless Networks*, vol. 31, no. 1, pp. 759–777, 2025.
- [16] Y. Cheng, A. Vijayaraj, K. S. Pokkuluri, T. Salehnia, A. Montazerolghaem, and R. Rateb, "Vehicular fog resource allocation approach for vanets based on deep adaptive reinforcement learning combined with heuristic information," *IEEE Access*, 2024.
- [17] A. Behura, A. Kumar, and P. K. Jain, "A comparative performance analysis of vehicular routing protocols in intelligent transportation systems," *Telecommunication Systems*, vol. 88, no. 1, p. 26, 2025.
- [18] S. Salonikias, I. Mavridis, and D. Gritzalis, "Access control issues in utilizing fog computing for transport infrastructure," in *International Conference on Critical Information Infrastructures Security*. Springer, 2015, pp. 15–26.
- [19] B. Jamil, H. Ijaz, M. Shojafar, and K. Munir, "Irats: A drl-based intelligent priority and deadline-aware online resource allocation and task scheduling algorithm in a vehicular fog network," *Ad hoc networks*, vol. 141, p. 103090, 2023.
- [20] X. Wu, S. Zhao, and H. Deng, "Joint task assignment and resource allocation in vfc based on mobility prediction information," *Computer Communications*, vol. 205, pp. 24–34, 2023.
- [21] M. Goudarzi, M. Palaniswami, and R. Buyya, "A distributed deep reinforcement learning technique for application placement in edge and fog computing environments," *IEEE Transactions on Mobile Computing*, vol. 22, no. 5, pp. 2491–2505, 2021.
- [22] S. Liao, J. Wu, S. Mumtaz, J. Li, R. Morello, and M. Guizani, "Cognitive balance for fog computing resource in internet of things: An edge learning approach," *IEEE Transactions on Mobile Computing*, vol. 21, no. 5, pp. 1596–1608, 2020.
- [23] S. D. Okegbile, B. T. Maharaj, and A. S. Alfa, "A multi-user tasks offloading scheme for integrated edge-fog-cloud computing environments," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 7, pp. 7487–7502, 2022.
- [24] S. Khan, I. A. Shah, N. Tairan, H. Shah, and M. F. Nadeem, "Optimal resource allocation in fog computing for healthcare applications," *Comput. Mater. Contin.*, vol. 71, no. 2, pp. 6147–6163, 2022.
- [25] S. Khan, I. A. Shah, S. Ahmad, J. A. Khan, M. S. Anwar, and K. Aurangzeb, "A comprehensive survey on multifacet fog-computing resource management techniques, trends, applications and future directions," *Expert Systems*, vol. 42, no. 4, p. e70019, 2025.
- [26] O. Nazih, N. Benamar, H. Lamaazi, and H. Chaoui, "Toward secure and trustworthy vehicular fog computing: A survey," *IEEE Access*, vol. 12, pp. 35 154–35 171, 2024.
- [27] M. Babar, M. S. Khan, A. Din, F. Ali, U. Habib, and K. S. Kwak, "Intelligent computation offloading for iot applications in scalable edge computing using artificial bee colony optimization," *Complexity*, vol. 2021, no. 1, p. 5563531, 2021.
- [28] K. Gasmi, S. Dilek, S. Tosun, and S. Ozdemir, "A survey on computation offloading and service placement in fog computing-based iot," *the Journal of Supercomputing*, vol. 78, no. 2, pp. 1983–2014, 2022.
- [29] M. Babar, M. S. Khan, U. Habib, B. Shah, F. Ali, and D. Song, "Scalable edge computing for iot and multimedia applications using machine learning," *Human-centric Computing and Information Sciences*, vol. 11, p. 41, 2021.
- [30] B. Jamil, M. Shojafar, I. Ahmed, A. Ullah, K. Munir, and H. Ijaz, "A job scheduling algorithm for delay and performance optimization in fog computing," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 7, p. e5581, 2020.
- [31] S. Khan, I. A. Shah, M. F. Nadeem, S. Jan, T. Whangbo, and S. Ahmad, "Optimal resource allocation and task scheduling in fog computing for internet of medical things applications," *Hum.-Centric Comput. Inf. Sci.*, vol. 13, p. 56, 2023.
- [32] M. Babar, M. S. Khan, F. Ali, M. Imran, and M. Shoaib, "Cloudlet computing: recent advances, taxonomy, and challenges," *IEEE access*, vol. 9, pp. 29 609–29 622, 2021.
- [33] S. Khan, I. A. Shah, K. Aurangzeb, S. Ahmad, J. A. Khan, M. S. Anwar, and M. Babar, "Energy efficient task scheduling using fault tolerance technique for iot applications in fog computing environment," *IEEE Internet of Things Journal*, 2024.