Vehicular Traffic Congestion Detection System and Improved Energy-Aware Cost Effective Task Scheduling Approach for Multi-Objective Optimization on Cloud Fog Network

Praveen Kumar Mishra, Amit Kumar Chaturvedi
Computer Application Deptt., Govt. Engg. College, Ajmer, BTU, Bikaner, India

***Abstract*—A current research area called fog computing aims to extend the advantages of cloud computing to network edges. Task scheduling is a crucial problem for fog device data processing since a lot of data from the sensor or Internet of Things layer is generated at the fog layer. This research suggested a vehicular traffic congestion detection model and an energy-aware cost effective task scheduling (ECTS) method in a cloud fog scenario. This research proposes an ECTS approach to allocate jobs to the fog nodes effectively. The recommended scheduling approach minimizes energy consumption and decreases expenses for time-sensitive real-time applications. The ECTS algorithm is implemented, and results are analysed using the iFogSim simulator. The proposed method minimizes energy consumption and cost. The suggested ECTS method is tested with five sets of inputs in this paper. The experiment's results show that an ECTS minimizes energy consumption in comparison to alternative algorithms. It also reduces the execution cost. The suggested approach outperforms both the Round-Robin (RR) and Genetic Algorithm techniques. According to the simulation results, the suggested algorithm reduced overall costs by 13.38% and energy usage by 6.59% compared to the Genetic Algorithm (GA). Compared to RR, the proposed method minimizes energy use by 13.76% and total costs by 18.46%.**

***Keywords*—IoT; fog computing; task scheduling; multi objective Model; iFogSim tool**

I. Introduction

One of the most crucial improvements in the realm of technology in the last several years is the Internet of Things (IoT) devices for the computation and exchange of information. IoT devices allow a wide range of items and equipment like sensors, cameras, automobiles, and connectivity with the Internet with smart devices like cell phones and laptops. Numerous applications and service offerings, such as latency minimization, control of traffic, and response time improvement, can be carried out as a result. Large volumes of data are produced through end devices as a result, which require supervision, processing, and analysis to generate appropriate data that will meet the user's objectives and aims. Moreover, the volume of data and a variety of required services and apps are expanding very quickly, demanding more computing power than even the most advanced smart devices can no longer match. The well-known cloud environment is a vast repository of resources that permits the universal ability to share and dynamically provide users with resources through virtualization procedures, which is one potential platform to aid in IoT improvements. By shifting resource and service-intensive jobs to a trustworthy computer environment, like the cloud, allowing smart devices to undertake basic tasks, limitations of current smart devices, such as enhancements, might be made to processing speed, capacity of storage, and resources required at the network. However, combining the use of clouds with the IoT creates further problems. It is anticipated that 50 billion IoT devices has been deployed in 2023. This figure will be increased to 35.8 billion in 2030, with the exponential rise in connected devices and cloud architectures that rely on traditional centralized processing features where storage and computational resources combined won't be enough to handle the demands of the Internet of Things devices burden. The main reason is that IoT gadgets and the cloud's infrastructure are quite far apart. The enormous amount of data of IoT devices send via the Internet to the cloud will strain the network's capacity and bandwidth, causing congestion, particularly near bottlenecks [4]. IoT applications are latency sensitive; therefore, a transmission delay reduces the Quality of Services (QoS), negatively impacting the user experience.

Fog computing [1], an innovative strategy of cloud computing first introduced by Cisco, has the potential to transform connecting the network’s boundary to a distributed processing design that can accommodate the end devices services. By using fogging, clients may access computing and data storage power resources more conveniently by extending cloud computing to data-generating and data-receiving IoT gadgets; instead of moving all the processing to the CDS (Cloud Data Center), the Fog layer goals to process the maximum of the traffic load created by end devices nearby to the user’s ranger of the network, called fog computing devices. Anywhere there is network connectivity, such as factories, shopping malls, electricity poles, railroads, inside of cars, etc., may use end devices. A fog node is any device with networking, computing, and storage capabilities. These devices include embedded servers, switches, routers, controllers, and security cameras. Requests are optimized for transmission time by putting resources near the network’s edge, where the minimum amount of time needed for information to arrive at a point of processing larger-scale and delay-tolerant activities still be routed to the cloud layer. At the same time, smaller jobs or task requirements with low latency have to be given precedence to be handled by fog computing platforms deposited at fog nodes with limited processing capability. Ultimately, fogging and cloud computing combine to create the cloud fog scenario, a new paradigm for the computing environment. This innovative strategy has several benefits, which include latency minimization, minimize high network traffic, and minimizing power consumption. Balancing of load for jobs ensures that no resource remains idle while others are being used [1]. The security risks businesses face using cloud computing has decreased [2]. For enterprises deploying solution of big data on cloud infrastructure, is a crucial factor to consider.

The paper is presented in the following sequence. Specific literature review for task scheduling within the framework of cloud fog atmosphere is provided in section 2. The proposed Vehicular Traffic Congestion Detection (VTCD) System and ETS scheduling algorithm are presented in Section 3. Section 4 provides a simulation environment. Section 5 includes performance analysis and simulation outcomes. Section 6 present the conclusion and forthcoming scope at the end**.**

II. Literature Review

According to Jayasena et al. [3], scheduling of job necessitates optimizing two goals: minimizing cost and decreasing power utilization. The author designed a meta-heuristic Whale Optimization Algorithm (WOA) mapped procedure to explain the recommended system and calculate the outcomes in iFogSim against heuristic techniques like Particle Swarm Optimization (PSO) and RR and SJF. Xu et al. [4] explain the laxity-mapped precedence approach to build a scheduling of task order with a fair priority. According to the author, based on the ant colony system (ACO) algorithm, this strategy minimizes overall energy use.

Tan, et al. [5] proposed an energy-efficient approach and looked at a task scheduling issue along with time limit restrictions in instances when could exist distributed throughout heterogeneous assets, such as fog computing, and an energy-conscious algorithm capable of finding the best solution in a polynomial amount of time. Nikoui et al. [6] developed a genetic-based (CAGB) planning method that improves efficiency at a lower cost for real-world applications with tight deadlines. Its effectiveness is evaluated regarding system overload, expenses, and delay. Fellir Z. et al. [7] presented multi-agent-based planning method, the most important tasks are handled first to ensure that when a packet with the highest importance goes into the waiting line, the job is dealt with, without interfering with the least significant task's implementation if it is presently being run. Madej et al. [8] presented four scheduling schemes named NFCFS (Naive First Come First Serve) technique. The other three schemes are client fair, prioritized fair, and hybridization.

Abdel Basset, et al. [9] suggested method improved the effectiveness of the best outcome and justified the workload across the accessible simulated engines by using a meta-heuristic method and a shift modification technique. Yang et al. [10] presented the superior value efficiency and scale outcome set to tackle the two-parameter collaboration to minimize fog computing scheduling task difficulties. The results demonstrate how the suggested strategy performs better than conventional strategies regarding resource cost, overall job execution time, etc. Hoseiny et al. [11] suggested cost aware scheduling method reduces latency, computation costs, and communication costs for IoT inquiries while increasing the proportion of tasks that are finished earlier than the deadline. This algorithm is compared with genetic algorithm.

The Task Priority Resource Allocation (TPRA) algorithm was presented by Dang et al. [12]. This algorithm's primary goal is to minimize the average latency in the fog network's diverse environment. Abdel-Basset et al. [13] presented the multiple objective task scheduling technique. This algorithm aims to minimize the rate of carbon emissions, make-span, and energy consumption. The resource-aware-cost-efficient (RACE) scheduler was introduced by Arshed et al. [14]. This method distributes the incoming jobs to fog nodes function. This strategy pursues minimizing bandwidth use, maximize Fog Node (FN) utilization at the fog layer, and shorten application make span.

In a fog context, Singh et al.'s [15] hybrid swarm optimization using genetic algorithms (GA) reduces execution time and cost. Compared to GA and PSO, the workflow scheduling experimental result that is being provided is superior. The MGWO multi-objective optimization approach was introduced by Saif, et al. [16] multiple goals, including make-span, throughput, energy, and delay. This approach aims to ascertain the optimal strategy for work scheduling at the fog layer. The MGWO algorithm's experimental result outperforms the equivalent methods regarding power minimization and delay reduction. Zhang et al. [17] introduced the Enhanced Whale Optimization Algorithm (EWOA). It's a technique for scheduling tasks with multiple objectives in a cloud computing environment—a search strategy known as Levy's struggle in EWOA. The results of the various heuristic and meta-heuristic algorithms match the results of the EWOA experiment. EWOA performs better in cost reduction and energy use minimization than these existing algorithms. Alwabel et al. [18] offered a deadline and power-efficient job scheduling method in a fog computing network. This method aims to determine which jobs are crucial and prioritize them so that they may be finished at the fog layer. The simulator iFogSim is used for outcome analysis. The recommended approach outperforms earlier algorithms regarding deadline and energy consumption reduction.

The PEWO (Parallel Enhanced Whale Optimization) approach was created by Khan et al. [19] for task scheduling at the cloud computing layer. This meta-heuristic method aims to minimize make-span and execution time. In a heterogeneous cloud environment, tasks are assigned using the PEWO approach. The experimental result of PEWO is better represented by the random matrix particle swarm optimization (RMPSO). Ali et al. [20] presented DNSG, a task scheduling system, by dynamically assigning the task to a fog node; the proposed approach aims to reduce make-span and cost compared to modified GA. Balancing tasks is also one of the issue in cloud fog environment. Within the Cloud-Fog system [24-25], job scheduling aims to maximize benefits for either service End users are concerned about minimizing make-span, power consumption and cost.

A thorough analysis reveals that there is a trade-off between minimizing costs and minimizing energy use. Consequently, this paper proposes ECTS to assign jobs to FN at the fog layer with the least amount of energy consumption and the best possible cost. WOA is inherited by the proposed ECTS method. This method determines the fitness function by adding up the expenses of RAM and CPU (central processing unit) for every fog node, together with the power used for task execution and FN's power when idle. The primary contribution of this study is the development of the ECTS algorithm. This suggested method for vehicular traffic congestion detection applications is simulated using the iFogSim simulator as the secondary contribution.

Table I presents a survey of the latest published study on cloud and fog computing task scheduling strategies, as well as main idea, improvement parameters and algorithms of previous studies.

TABLE I. REVIEW OF THE EXISTING CLOUD FOG ENVIRONMENT'S SCHEDULING PROCEDURES

Year	Author	Improvement Parameter	Algorithm	Main ideas
2019	Jayasena, et al. [3] IEEE	Minimize energy consumption, Reduce cost	Whale Optimization task scheduling algorithm	A fog processing system job-planning strategy that optimizes two objectives: reducing power consumption and cutting expenses.
2019	Xu, et al. [4] IEEE	Energy consumption, Execution time	Laxity based Ant Colony algorithm [LBACA]	To effectively control the adaptability of work latency and energy consumption, implemented the ant colony systems algorithm and flexibility while accounting for the relevance of each task and when it will be finished.
2020	Tan, et al. [5] Elsevier	Energy, Deadline	Energy Efficient scheduling method	An energy-efficient task scheduling method finding the best solution in a polynomial time.
2020	Nikoui, et al. [6] IEEE	Deadline, Cost	Genetic algorithm	A genetic-based (CAGB) planning method that improves efficiency at a lower cost for real-world applications with tight deadlines. Its effectiveness is evaluated regarding system overload, expenses, and delay.
2020	Fellir Z, et al. [7] IEEE	Priority, Execution Time	Priority based task scheduling algorithm	Multi-agent-based planning method, the most important tasks are handled first to ensure that when a packet with the highest importance goes into the waiting line.
2020	Madej, et al. [8] IEEE	Priority, Job Execution	Priority based task scheduling algorithm	Four scheduling schemes named NFCFS (Naive First Come First Serve) technique. The other three schemes are client fair, prioritized fair, and hybridization are presented
2020	Abdel Basset, et al. [9] IEEE	Energy, Makespan	Energy aware task scheduling algorithm	Improved the effectiveness of the best outcome and justified the workload across the accessible simulated engines by using a meta-heuristic method and a shift modification technique.
2020	Yang, et al. [10] IEEE	Total task execution time, resource cost	Meta heuristic scheduling algorithm	Demonstrate how the suggested strategy performs better than conventional strategies regarding resource cost and execution time.
2021	Hoseiny, et al. [11] IEEE	Cost, deadline	Combined (QoS) quality of service and cost effective scheduling method	In contrast to a genetic algorithm, the suggested technique reduces latency, compute costs, and communication costs all at once for IoT inquiries while increasing the proportion of tasks that are finished earlier than the deadline.
2021	Dang, et al. [12] IEEE	Task priority	An algorithm for allocating resources based on task priorities	Resource allocation algorithm based on task priority reduces the average delay in the heterogeneous environment in the fog environment.
2021	Abdel-Basset, et al. [13] IEEE	Energy, makespan	Multi objective scheduling algorithm	The purpose of this algorithm is minimizing energy, make-span and carbon emission rate.
2021	Arshed, et al. [14] IEEE	Execution time, cost	RACE scheduler	Resource aware cost efficient scheduling algorithm at fog layer.
2023	Singh, et al. [15] IEEE	Makespan, cost	Hybrid particle swarm optimization with genetic algorithm (GA)	The purpose of this algorithm is to reduce execution time and cost in cloud fog environment.
2024	Saif, et al. [16] IEEE	Throughput, makespan	Multi-Objectives Grey Wolf Optimizer (MGWO) algorithm	MGWO is multi objective optimization technique for optimal solution of task scheduling.
2024	Zhang, et al.[17] Springer	Execution Cost, power consumption	Improved Whale Optimization Algorithm (EWOA)	EWOA is advanced task scheduling algorithm at cloud computing environment.
2024	Alwabel, et al. [18] IEEE	Deadline , energy consumption	Power-Aware Placement Mechanism (POAPM)	Deadline and energy consumption minimization scheduling at fog computing network
2024	Khan et al. [19] IEEE	Make-span, throughput	Parallel Improved Whale Optimization (PIWO) algorithm	PIWO algorithm is used for allocation of tasks at cloud computing layer. The purpose of this algorithm is minimizing make-span and execution time
2022	Ali et al. [20] IEEE	Execution time, cost	Non-dominated Sorting Genetic (NSG) algorithm II	DNSG is scheduling algorithm at cloud fog environment. Purpose of proposed algorithm is minimizing cost and execution time

III. PROPOSED SYSTEM ARCHITECTURE

This section describes the proposed three tire architecture of a vehicular congestion detection system, and the ECTS algorithm in cloud fog network.

A. Proposed Vehicular Traffic Congestion Detection Architecture

Vehicular Traffic Congestion Detection (VTCD) architecture in a cloud fog network is shown in Fig. 1. As shown in the diagram this is a three-layer model for detecting vehicular traffic congestion. Layer 1 represents the end devices layer, and at this layer, sensors detect vehicular traffic congestion, and forward requests at layer two, i.e., fog computing layer through IoT enabled devices. At the layer two, clusters of FN are available. Each cluster of FN is connected to a Master Fog Server (MFS) called a fog server. MFS is responsible for checking resource availability, scheduling tasks to FN, and assigning tasks (jobs) to the appropriate FN. These fog nodes process jobs and respond to MFS. Using an actuator, MFS responds to end devices and displays traffic congestion detection-related information. MFS also forwards task results to layer three, the Cloud Data Center (CDC), through a proxy server to store the results for future reference.

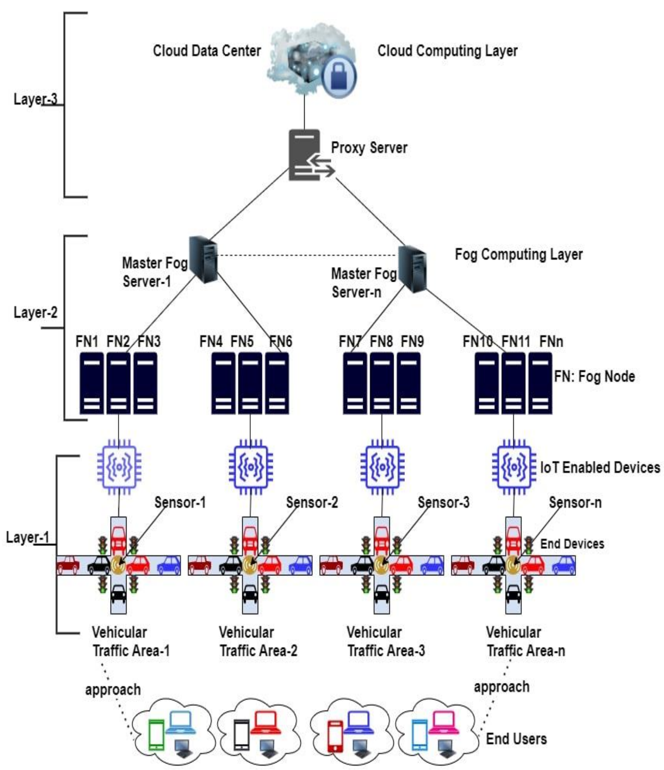

Fig. 1. Proposed three-tier model of vehicular traffic congestion detection.

1) Vehicular traffic area and end devices: As shown in Fig. 1, end users approach end devices using gadgets like tablets, smartphones, desktops, notebook computers, wearable devices, etc. In this paper, end devices are vehicular traffic on roads in different city areas. As we can see in this diagram, vehicular traffic area 1 to vehicular traffic area n are shown where sensor 1 to sensor n detect traffic on the road and forward this information to layer 2 using IoT-enabled devices.

2) The proxy server and fog computing layer: Fog computing is the middle layer in the cloud fog scenario with three-tier architecture called fogging or fog networking. FNs are near-end devices for computing, storage, and communication with end users locally with reduced latency, low bandwidth and lower cost compared to cloud computing environments. An enhanced version of cloud computing, i.e., fogging, reduces stress in the layer of clouds [22-23]. A proxy server is router that communicates with and prevents cyber-attacks, reduces latency between the CDS and layer 2. Data can be retrieved by fog nodes from cloud storage whenever further processing is required.

B. Proposed Task Scheduling Model

The suggested task scheduling technique for assigning jobs to the FN in the cloud fog network is covered in this part. A recommended job scheduling plan is based on the WOA [21] and the multiple-objective model [26]. Fig. 2 depicts the system of the suggested ECTS method of this paper for assigning tasks to the FN. The memory, CPU, and energy consumption functions are all computed using the multiple-objective computation. The cost function and energy consumption are added to get the fitness value. By the fitness rating, the tasks are allocated to the Fog nodes. ECTS first considers the current solution is the best solution. This process is repeated until the optimum solution is identified. In this paper task scheduling aims to minimize energy and total cost while assigning task to the fog node as effectively as feasible.

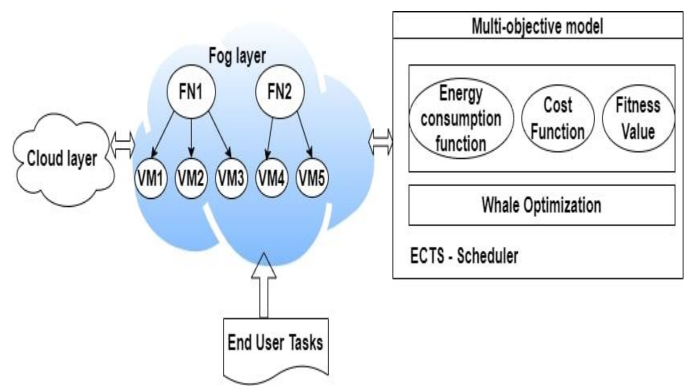

Fig. 2. Model of the proposed ECTS scheduler.

The fog layer which contains of n numbers of fog nodes. Where FL represents fog layer and $\{FN1, FN2, FN3, FN4 \ldots\ldots\ldots FN_n\}$ represent fog nodes presented at fog layer. This can be represented as,

$$Fog,\ FL = \{FN1,\ FN2,\ \ldots FN_n\} \tag{1}$$

The fog node FN1 can be represented by the equation that follows. Where, $\{FN1, FN2, \ldots FN_n\}$ represents the fog nodes. Master Fog Node (MFS) connected with cluster of fog nodes at layer 2.

$$MFS = \{FN1,\ FN2,\ \ldots FN_n\} \tag{2}$$

Each fog node has the CPU and the memory. Tk represents task and MFS represent master fog server.

$$Tk = \{ T_1,\ T_2 \ldots\ldots T_{500} \} \tag{3}$$

Where, T_1 is the task first and T_{450} shows 500^{th} task. Assume that one IoT device forward 10 Tk to MFS.

In this paper proposed algorithm: ECTS, where tasks and fog nodes are inputs. The suggested task scheduler aims to best distribute the job across the fog nodes. The WOA is the basis for this scheduler. Initially, the search agent population is initialized. The model of proposed ECTS scheduler computes the fitness value. Equation number 15 updates the search agent's position if the e is greater than or equal to 0.5. The 13^{th} and 14^{th} equation are used to update the search agents' positions if the e is less than 0.5. Until the ideal solution is found, this procedure is carried out repeatedly.

Proposed Algorithm : ECTS
Input: Task T, Fog Node FN
Result: Fog nodes are assigned the jobs.
Control Parameters S^*, α, N, W, t, p
Begin
Set up the population initially. S_i (i = 1, 2, …, n)
Fitness value obtained from the sub function
Initialize the current best agent S^*
Update α, N, W, t, p
if (e < 0.5)
if (|N| < 1)
Change the search agent's location with equation 13.
Else if (|N| ≥ 1)
Change the search agent's location with equation 14.
End if
End if
if (e ≥ 0.5)
Change the search agent's location with equation 15.
End if
In the event that (any search agent leaves the search space)
Update S^*
Assign y to y + 1.
End if
End
Sub function:
Input: Tasks T, Fog Nodes FN
Output: Fitness value
For (all the Fog nodes)
Find active energy consumption function by equation 4
Find idle energy consumption function by equation 5
Find total energy consumption function by equation 6
Find the fitness value by equation 12
End for
End

1) Energy Consumption: Energy consumption formula is similar like [15].

$$ER_{usage}(y) = \sum_{i=1}^{n} \mu\, F_i\, V_i^2\ \ (BT_{T_i} - ET_{T_i}) \tag{4}$$

$$ER_{idle}(y) = \sum_{j=1}^{m} \sum_{idle_{jk}\ \varepsilon\ IDLE_{jk}} \mu\ RF_{\min i}\ \ VL^2_{\min i} \tag{5}$$

The power consumption during job i, when the resource is operating at peak efficiency and is about to enter sleep mode, and added together to get the total energy consumption.

ER_{usage} is active energy usage and ER_{idle} is idle energy consumed by system at sleep mode. F_i is frequency, V_i is voltage supply fog node where task executes. BT_{T_i}, represent beginning time and ET_{T_i} is end time for task T_i and μ is constant.

$$TER_{con}\ (\mathrm{y}) = ER_{usage} + ER_{idle} \tag{6}$$

2) Total cost: The fog node total cost is calculated as,

$$\mathrm{C}\ (\mathrm{y}) = \sum_{k=1}^{|FN|} C_{cost}\ (k) \tag{7}$$

$$C_{cost}\ (k) = C_{basic} * C_k * t_{ik} * C_{tr} \tag{8}$$

$$\mathrm{M}\ (\mathrm{y}) = \sum_{k=1}^{|FN|} M_{cost}\ (k) \tag{9}$$

$$M_{cost}\ (k) = M_{basic} * M_k * t_{ik} * M_{tr} \tag{10}$$

Where $C_{cost}\ (k)$ is the cost of CPU of fog node FN_k and t_{ik} is the amount of time in which task T_i is executed at node S_k. C_{tr} is the communication cost of the CPU of fog node. Here , C_{basic} and C_{tr} are constant where C_{basic} is 0.16 per hours and C_{tr} is 0.004, much like in [26]. |FN| is the total no of fog nodes. M_{basic} is 0.04 GB per hour and M_{tr} is 0.4, C(y) is the total no of cost of CPU of FN and M(y) is the memory cost of FN . $TC\ (y)$ denote the total cost, it can be calculated as,

$$TC(y) = \mathrm{C(y) + M(y)} \tag{11}$$

3) Fitness value determining: To determine the best solutions, the fitness value is computed; the solution must have the lowest possible energy consumption and lowest possible cost function. The following formula is used to calculate fitness.

$$FV(y) = TER_{con}\ (\mathrm{y}) + TC(y) \tag{12}$$

4) Whale optimization algorithm: For distributing the jobs to the fog nodes as efficiently as possible, the whale optimization method [21] is explained. The collection of random solutions is where the whale optimization process starts. It moves forward with the process under the presumption that the present answer is optimal. Repeating this procedure keeps on until the best solution is found.

$$S\ (y+1) = \overrightarrow{S*}(y) - \vec{N} * \vec{D} \tag{13}$$

$$\vec{S}(y+1) = \overrightarrow{S_{rand}} - \vec{N} * \vec{D} \tag{14}$$

$$\vec{S}(y+1) = D' * b^{vt} * \cos(2\,\Pi\, t\,) + S^*\,(y) \tag{15}$$

Where, y denotes current iteration, $\vec{S}$ is position vector and $\overrightarrow{S*}$ represents optimal solution. $\overrightarrow{S_{rand}}$ is random location vector, $\vec{N}$ represents the coefficient vector. $\vec{W}$ represents the coefficient vector. In equation 11, v is constant and t shows the value in [-1, 1] interval. | | denotes the absolute value, while * denotes multiplication of elements by elements. D' is calculated as follows,

$$D' = |\overrightarrow{S*}(y) - \vec{S}(y)| \quad (16)$$

$$\vec{D} = |\vec{W} * \overrightarrow{S*}(y) - \vec{S}(y)| \quad (17)$$

Where $\vec{N}$ and $\vec{W}$ can be computed by following formula,

$$\vec{N} = 2\ \vec{\propto} * \vec{\beta} - \vec{\propto} \quad (18)$$

$$\vec{W} = 2 * \vec{\beta} \quad (19)$$

The value of $\vec{\propto}$ is move between 2 to 0 and $\vec{\beta}$ denote the arbitrary vector in [0, 1].

IV. Simulation Environment

The simulation environment used to perform calculations is explained in this section. Simulation scenario of Vehicular Traffic Congestion Detection (VTCD) system in fog computing environment has shown in Fig. 3. where S1, S2, S3, S4 are sensors to collect cross-road vehicular traffic information and A1, A2, A3, A4…A8 are actuators to display the results and total four terminals for VTCD system IoT Devices (T_IoT_D1, T_IoT_D2, T_IoT_D3, T_IoT_D4) are used to collect VTCD information from sensors and forward to Fog Node (FN) where four fog nodes (FN1, FN2, FN3, FN4) have used in this simulation at fog computing layer. Controller Fog Server (MFS) collects information from FN and stores it at CDC (Cloud Data Center) via a proxy server. This topology was created and simulated by the iFogSim simulator.

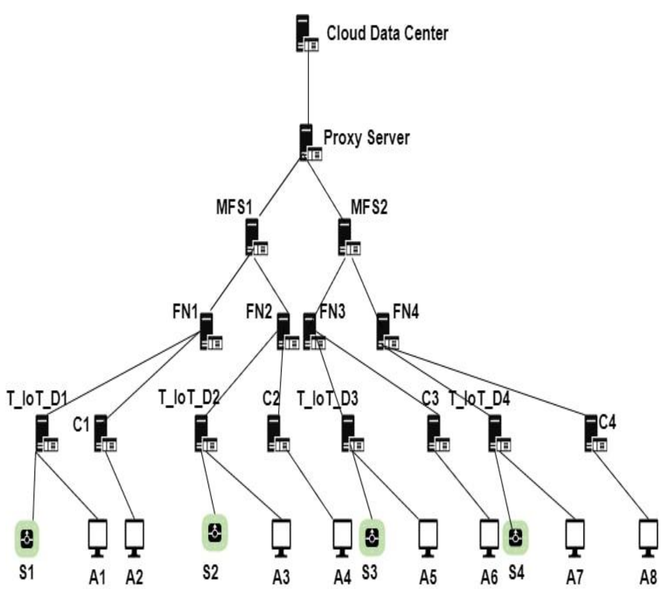

Fig. 3. Scenario of ECTS with VTCD for simulation in cloud fog environment.

Table II shows configuration parameters and values at the Cloud Server, Proxy Server, and Fog Computing layer, which are used to simulate the cloud fog environment. Table III represents the description, various notations used, values assumed in this paper, and system configuration.

TABLE II. Configuration Parameter

Requirement	at Cloud	Proxy server	at Fog
Processing unit (MIPS)	44700	2900	2900
Main memory in MB	9900	3900	3900
Up_bps in MB	100	9900	9900
Dn_bps in MB	9900	9900	9900
Layer	3	2	1
Rate in MIPS	0.01	0	0
power_b in WATT	17*104	107.349	107.349
power_ID in WATT	17*83.24	85.5333	85.5333

TABLE III. Notation, Values and Description

Description	Notation and values
Max no of IoT device	50
p	[-1 , 1]
i	1, 2, 3…
Max_{itr}	100
FN	fog node
EDN	edge device node
System	Intel ® Core(TM) i3 CPU
Tool for simulation	iFogSim
OS(operation system)	Window 7 Ultimate, 64 bit

V. Experimental Results and Performance Analysis for Proposed ECTS Algorithm

Performance evaluation of the proposed algorithm with VTCD application in a cloud fog network is shown in this section. The measurements for energy consumption performance and the simulation result of overall cost are shown in Tables IV and V, respectively. The corresponding bar chart of the parameters shows that energy consumption is minimized when number of IoT devices have increased as shown in Fig. 4, and cost is also minimized when no of IoT devices have increased as shown in Fig. 5. Assume that 10 IoT devices are equal to 100 tasks.

TABLE IV. Simulation Results for the Energy Consumption

No. of IoT Devices	Consumption of energy (in WATTS)
10	188040.91
20	185103.87
30	185103.39
40	182487.18
50	176916.18

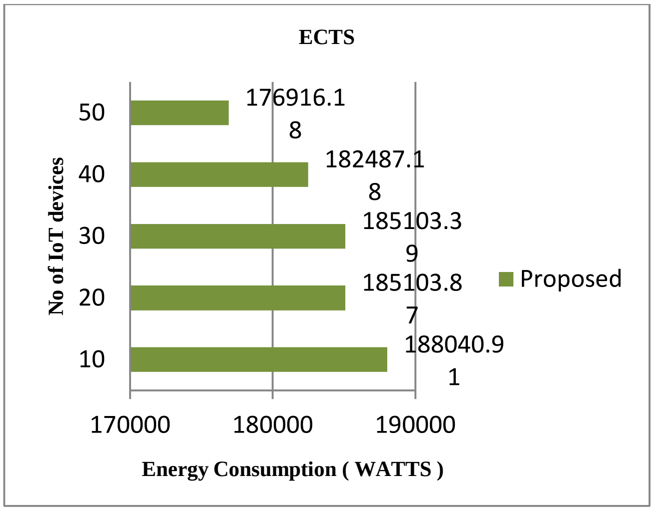

Fig. 4. Energy consumption.

TABLE V. SIMULATION RESULTS FOR THE COST

No. of IoT Devices	Cost ($)
10	395905.15
20	392444.41
30	614328.21
40	761434.06
50	810188.88

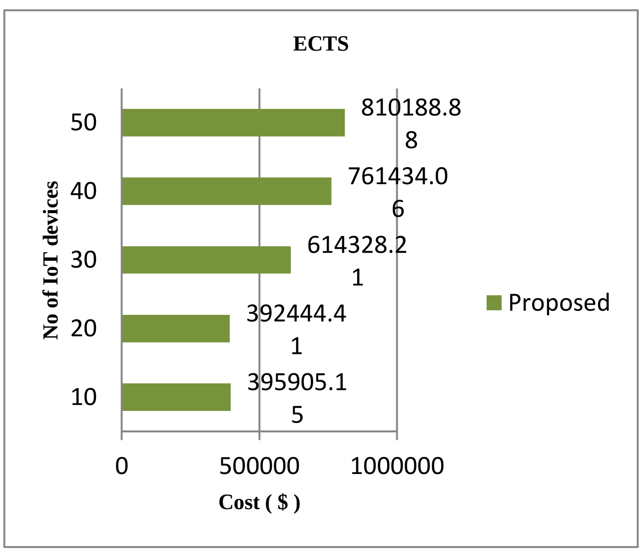

Fig. 5. Cost.

Table VI shows the comparative study of the proposed ECTS method with the existing methods, such as (Round Robin) RR [3] and (Cost aware genetic algorithm) GA [6]. The proposed ECTS method has the minimum energy consumption 176916.18. The energy consumption of RR is 201259.8464 and the energy consumption of GA is 188574.9563. The total cost of RR and GA are 959749.7472 and 918592.1521 respectively while the total cost of proposed ECTS method is 810188.88 which is smaller than the other existing methods.

TABLE VI. COMPARATIVE STUDY OF THE ECTS METHOD WITH THE RR AND GA

	ECTS	RR	GA
Energy consumption	176916.18	201259.8464	188574.9563
Cost	810188.88	959749.7472	918592.1521

This is the analysis of the proposed algorithm in this paper. A range of IoT devices as input of 10-50 have been used for the simulation at the iFogSim simulator. Assume that 10 IoT devices are equal to 100 tasks and 50 IoT devices are equal to 500 tasks. As shown in Fig. 6 energy consumptions are minimized in the proposed algorithm when no of IoT devices are increased as compared to RR and GA. As shown in Fig. 7. The proposed algorithm minimizes the overall cost when the number of IoT devices increases compared to RR and GA.

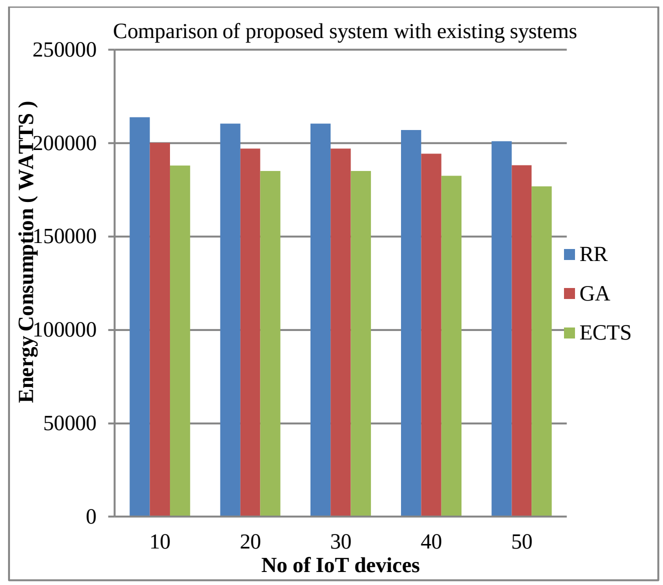

Fig. 6. Energy consumption comparison.

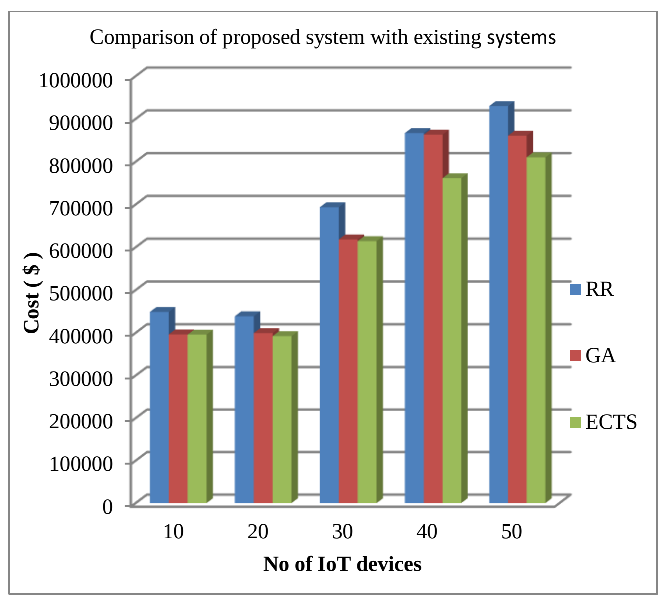

Fig. 7. Cost comparison.

Column chart for comparison of energy consumption, overall cost with GA, RR, and proposed algorithm show that the proposed result is better compared to others, especially in energy and cost parameters.

VI. Conclusion

The design of the task scheduling technique is the primary purpose of this study. The secondary objective is designing a Vehicular Traffic Congestion Detection (VTCD) system. The proposed Energy-aware Cost effective Task Scheduling (ECTS) scheduling algorithm performance has been analyzed using various inputs. Two other approaches, particularly the Genetic Algorithm (GA) and Round-Robin (RR) in a cloud-fog network, were compared with the proposed algorithm using the iFogSim simulator. Especially for energy usage and cost parameters, our proposed algorithm ECTS performed better than the others at five different sets of inputs. The simulation result shows that energy consumption is minimized by 6.59%, and the overall cost is minimized by 13.38% compared to GA. In comparison, energy consumption is minimized by 13.75%, and the overall cost is minimized by 18.46% compared to RR. Here, multi-objective means task's cost, energy consumption and deadline for scheduling the user's request at the fog computing layer. Furthermore, the suggested algorithm may adapt to the end user's requirement for higher processing performance for other applications.

In the future, improvements may be made in ECTS algorithm to address other issues like reducing make-span, response time, security issues, etc. improvement for other real time applications.

Acknowledgment

I sincerely thank Dr. A. K. Chaturvedi, Head, Computer Application Dept., at Govt. Engineering College, Ajmer, Rajasthan, for truthful academic backing throughout the work.

References

[1] A. I. Abueid, "Big Data and Cloud Computing Opportunities and Application Areas", Eng. Technol. Appl. Sci. Res., vol. 14, no. 3, pp. 14509–14516, Jun. 2024

[2] M. Ramzan, M. S. Farooq, A. Zamir, W. Akhtar, M. Ilyas, and H. U. Khan, "An Analysis of Issues for Adoption of Cloud Computing in Telecom Industries", Eng. Technol. Appl. Sci. Res., vol. 8, no. 4, pp. 3157–3161, Aug. 2018.

[3] N. Jayasena, K. P., & Thisarasinghe, B. S. (2019). Optimized task scheduling on fog computing environment using meta heuristic algorithms. 2019 IEEE International Conference on Smart Cloud. doi:10.1109/smartcloud.2019.00019.

[4] Xu, J., Hao, Z., Zhang, R., & Sun, X. (2019). A Method Based on the Combination of Laxity and Ant Colony System for Cloud-Fog Task Scheduling. IEEE Access, 7, 116218–116226. doi:10.1109/access.2019.2936116.

[5] Tan, H., Chen, W., Qin, L., Zhu, J., & Huang, H., Energy-aware and Deadline-constrained Task Scheduling in Fog Computing Systems. 2020, 15th International Conference on Computer Science & Education (ICCSE). doi:10.1109/iccse49874.2020.92017.

[6] Nikoui, T. S., Balador, A., Rahmani, A. M., & Bakhshi, Z. (2020). Cost-Aware Task Scheduling in Fog-Cloud Environment. 2020 CSI/CPSSI International Symposium on Real-Time and Embedded Systems and Technologies (RTEST). doi:10.1109/rtest49666.2020.9140118.

[7] Fellir, F., El Attar, A., Nafil, K., & Chung, L. (2020). A multi-Agent based model for task scheduling in cloud-fog computing platform. 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT). doi:10.1109/iciot48696.2020.9089652.

[8] Madej, A., Wang, N., Athanasopoulos, N., Ranjan, R., & Varghese, B. (2020). Priority-based Fair Scheduling in Edge Computing. 2020 IEEE 4th International Conference on Fog and Edge Computing ICFEC). doi:10.1109/icfec50348.2020.00012.

[9] Abdel-Basset, M., El-shahat, D., Elhoseny, M., & Song, H. (2020). Energy-Aware Metaheuristic algorithm for Industrial Internet of Things task scheduling problems in fog computing applications. IEEE Internet of Things Journal, 1–1. doi:10.1109/jiot.2020.3012617.

[10] Yang, M., Ma, H., Wei, S., Zeng, Y., Chen, Y., & Hu, Y. (2020). A Multi-Objective Task Scheduling Method for Fog Computing in Cyber-Physical-Social Services. IEEE Access, 8, 65085–65095. doi:10.1109/access.2020.2983742.

[11] Hoseiny, F., Azizi, S., Shojafar, M., & Tafazolli, R. (2021). Joint QoS-aware and Cost-efficient Task Scheduling for Fog-cloud Resources in a Volunteer Computing System. ACM Transactions on Internet Technology, 21(4), 1–21. doi:10.1145/3418501.

[12] Tran-Dang, H., & Kim, D.-S. (2021). Task Priority-based Resource Allocation Algorithm for Task Offloading in Fog-enabled IoT Systems. 2021 International Conference on Information Networking. (ICOIN). doi:10.1109/icoin50884.2021.9333992.

[13] M. Abdel-Basset, N. Moustafa, R. Mohamed, O. M. Elkomy and M. Abouhawwash, "Multi-Objective Task Scheduling Approach for Fog Computing," in IEEE Access, vol. 9, pp. 126988-127009, 2021, doi: 10.1109/ACCESS.2021.3111130.

[14] J. U. Arshed and M. Ahmed, "RACE: Resource Aware Cost-Efficient Scheduler for Cloud Fog Environment," in IEEE Access, vol. 9, pp. 65688-65701, 2021, doi: 10.1109/ACCESS.2021.3068817.

[15] Singh, G., Chaturvedi, A.K. Hybrid modified particle swarm optimization with genetic algorithm (GA) based workflow scheduling in cloud-fog environment for multi-objective optimization. Cluster Comput 27, 1947–1964 (2024). https://doi.org/10.1007/s10586-023-04071-1.

[16] F. A. Saif, R. Latip, Z. M. Hanapi and K. Shafinah, "Multi-Objective Grey Wolf Optimizer Algorithm for Task Scheduling in Cloud-Fog Computing," in IEEE Access, vol. 11, pp. 20635-20646, 2023, doi: 10.1109/ACCESS.2023.3241240.

[17] Zhang, Y., Wang, J. Enhanced Whale Optimization Algorithm for task scheduling in cloud computing environments. J. Eng. Appl. Sci. 71, 121 (2024). https://doi.org/10.1186/s44147-024-00445-3.

[18] A. Alwabel and C. K. Swain, "Deadline and Energy-Aware Application Module Placement in Fog-Cloud Systems," in IEEE Access, vol. 12, pp. 5284-5294, 2024, doi: 10.1109/ACCESS.2024.3350171.

[19] Z. A. Khan, I. A. Aziz, N. A. B. Osman and S. Nabi, "Parallel Enhanced Whale Optimization Algorithm for Independent Tasks Scheduling on Cloud Computing," in IEEE Access, vol. 12, pp. 23529-23548, 2024, doi: 10.1109/ACCESS.2024.3364700.

[20] I. M. Ali, K. M. Sallam, N. Moustafa, R. Chakraborty, M. Ryan and K. -K. R. Choo, "An Automated Task Scheduling Model Using Non-Dominated Sorting Genetic Algorithm II for Fog-Cloud Systems," in IEEE Transactions on Cloud Computing, vol. 10, no. 4, pp. 2294-2308, 1 Oct.-Dec. 2022, doi: 10.1109/TCC.2020.3032386.

[21] Mirjalili S. and Lewis A., "The whale optimization algorithm," Advances in engineering software, vol. 95, pp. 51–67, Elsevier, 2016.

[22] Celso A. R. L. Brennand, Daniel Ludovico Guidoni (2021). Fog Computing-based Traffic Management Support forIntelligent Transportation Systems. 17165-217-13756-1-10-20210911.

[23] Ning, Z., Huang, J., & Wang, X. (2019). Vehicular Fog Computing: Enabling Real-Time Traffic Management for Smart Cities. IEEE Wireless Communications, 26(1), 87–93. doi:10.1109/mwc.2019.1700441.

[24] Mishra P. K., A. K. Chaturvedi, "State-Of- The-Art and Research Challenges in Task Scheduling and Resource Allocation Methods for Cloud-Fog Environment," 3rd International Conference on Intelligent Communication and Computational Techniques (ICCT), Jaipur, India, 2023, IEEE, doi: 10.1109/ICCT56969.2023.

[25] Mishra P. K., Chaturvedi A. K., "Research Challenges in Job Scheduling and Resource Distribution Methodology for Cloud Fog Network: An

Organized Analysis. " International Conference on Computational Intelligence, Communication Technology and Networking, 2023, IEEE, doi: 10.1109/CICTN57981.2023.

[26] Sreenu, K., Sreelatha, M. W-Scheduler: whale optimization for task scheduling in cloud computing. Cluster Comput 22 (Suppl 1), 1087–1098 (2019). https://doi.org/10.1007/s10586-017-1055-5