Clustering Technique for Mobile Edge Computing To Detect Clumps in Transportation-Related Problems

https://doi.org/10.3991/ijim.v17i04.37801

J. Albert Mayan[1](✉), S.V. Manikanthan[2], Azham Hussain[3], S. Nithyaselvakumari[4], A. Vinnarasi[5]
[1] Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, India
[2] Melange Academic Research Associates, Puducherry, India
[3] School of Computing, Universiti Utara Malaysia, Sintok, Malaysia
[4] Department of Medical Instrumentation, Saveetha school of Engineering, Thandalam, India
[5] Department of ECE, SRMIST, Kattankulathur, Chennai, India
albert.cse@sathyabama.ac.in

Abstract—The daily functioning of civilization depends heavily on transportation. In most cities, a sizable section of the working class consistently attends both employment and education. There are many different things you can do when commuting, such as unwinding, eating out, and other things. The most popular means of transportation in North Cyprus, particularly in developing cities, is island transportation, which includes the usage of both private cars and commercial vehicles. The advent of edge computing, which offers the opportunity to connect potent processing servers next to the mobile device, is a significant step toward improving user experience and reducing resource use. Mobile Edge Computing is the next trustworthy approach for how mobile devices consume communications and computing. Offloading computation is a key component developing mobile edge computing, which enables devices to get around clustering techniques' limitations and get around computing, storage, and energy constraints. However, computation offloading is not always the best strategy to use; making choosing unloading is a critical step that requires consideration of numerous factors. For instance, shifting the high-resource node to an edge server and granting similar capabilities to the low-resource nodes would delegate heavy duties to the external unit inside the network. The evaluations' results were noteworthy and substantial. Problems involving the vehicles of institutions and organizations can be resolved using the suggested solution to the school bus routing issue. We also test the impact of network latency on the delivery of a particular result using an Edge Computing simulator.

Keywords—mobile edge computing (MEC), transportation problems, save time, clustering technique, energy efficiency, resource optimization

1 Introduction

Any city's growth is largely dependent on the state of its transportation infrastructure and the services it provides to its populace. With access to real-time information, an intelligent transportation system offers citizens quick and simple options for safe transport. These facilities influence how quickly people move through their daily lives and benefit society in many other ways, such as by reducing pollution, boosting the economy, hastening development, improving health, and much more. Numerous researchers have suggested various methods and schemes to hold the large data produced by smart metropolises to provide more clever, smart, and sustainable outcomes. However, these solutions don't just concentrate on the big data aspect of real-time smart freight forwarding. By 2050, 75% of the world's population, or more than 7 billion people, were expected to live in cities and nearby suburbs. As a result, the amount of the city's traffic will considerably grow. Elevated big data will be produced by the significant increase in city traffic intensity that will result from the overabundance of vehicular communication and road sensors in the CPS ecosystem. A proportionate spike in on-road traffic accidents may also result from this massive increase in traffic intensity. As a result, residents will have issues with traffic congestion and delayed travel times.

Due to the rapid advancement of technologies, consumers may desire to obtain any on-road city traffic information at any time and from any location. In order to prevent traffic congestion, the authorities may also need to distribute city traffic by skilfully rerouting it to fewer congested transportations in actual time. Additionally, this dispersion lessens air contamination, improving community health. In a nutshell, traffic authorities are required to manage the traffic system wisely with the least amount of human intervention and resources. This clever transport infrastructure makes it easier for residents to access real-time transportation facts and has a big impact on how people live [1].

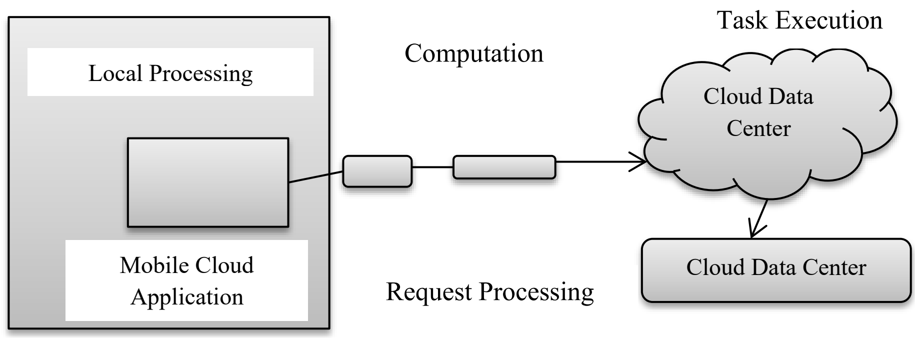

Fig. 1. Mobile Edge Computing

Edge computing often referred to as mobile edge computing has recently emerged to overcome the processing limitations of mobile devices in response to an increase in the number of computationally intensive apps. Edge computing relieves network capacity constraints by distributing the workload to dispersed computing clusters. Connections to servers are nevertheless slow, although cloud technology allows users to access a shared pool of computers and move expensive computations to the cloud. One of the key reasons why edge computing appears to have the ability to support portable virtual reality is because of this. Edge servers keep a local reference image database and use it to identify objects in image pixels. If the edge fails, the request goes to the cloud [2].

The main contribution of this paper is, namely, as follows:

- We bring forth a problem for minimising the amount of time a task takes to complete by jointly optimising its computational and networking resources, such as its edge computing capabilities.
- In order to handle the original optimisation problem's non-linearity and non-convexity, we separated it into smaller issues and iteratively resolved each one. Additionally, we also provide a closed-form method at MEC for task splitting and computing allocation of resources.
- To show the effectiveness of the suggested approach, statistical solutions are contrasted with those from an extensive search and other benchmark schemes. The numerical findings demonstrate that the proposed method outperforms all existing schemes and performs epsilon identically to the extensive search when task computation energy and time consumption are taken into consideration as outstanding development [3].

The essay's remaining sections are organised as follows. Section 2 presents the research on the pertinent prior work. Section 3 describes the features of the proposed system, including the proposed system architecture, implementation model, characteristics of the graph-based technique, and data analysis. The implementation environment is described and the system's effectiveness is rated in Section 4. Section 5 provides the resolution.

2 Related works

Liu, L., Zhao et.al [4] additionally, some academics talked about the use of transfer culture in MEC. Reference used distributed knowledge to simultaneously decide which MEC networks should be offloaded. In order to address the trade-off between latency and energy consumption, Reference established a hierarchical computational intelligence task distribution framework. In this framework, the lower layers of the pertained CNN model are embedded in the autonomous aerial vehicles, while the higher layers are handled by the MEC server. In addition to demonstrating the potential of deep learning and edge computing, this study also emphasises the importance of generalisation and resource problems in useful uses. These aforementioned methods frequently require much iteration to arrive at a local optimum, making them inappropriate for real-world applications involving compute offloading. Additionally, when the MEC network scale increases, their computational cost tends to increase dramatically.

Sarfraz, M., Alshahrani et.al [5] this method of computing is referred to as MEC. Unfortunately, due to the compute offloading link's current shortcomings, the MEC model has not yet realised all of its promise. For example, when unloading computation rather than completing the task locally, devices at the cell's perimeter have a notoriously poor offloading success rate and/or may face increased delay. As a result, these devices are forced to rely on their computing capacity, which is frequently insufficient to execute resource-demanding applications. The efficiency of the MEC systems' communications must be improved because of this.

abd Al_kadum Hassan, H., Hasan et.al [6] Vehicle-to-Vehicle system removes the need for a main station to manage the network architecture and provides a connection between vehicles using an ad hoc wireless network. Because of their high speed, vehicular ad hoc networks (VANETs) are distinguished by the self-organization of the vehicles and quick changes in the network topology. As link connections failures frequently occur in VANETs, ensuring the security of information in VANETs is more challenging than in typical MANETs. Creating a cluster with a hierarchical system within the network is an effective and affordable way to reduce the mobility effect and improve VANET network access.

Ahn, J., Lee, J., Park et.al [7] For the MEC environment, we suggest a power-efficient clustering method (PECS) it keeps the MEC servers' processing of incoming delays within acceptable bounds. By analysing the impact of the size of clusters on the CPU workloads, the optimization problem for choosing the clustering to lower the energy consumption of MEC servers is established and addressed. A thorough examination of the suggested model shows that it effectively determines the ideal number of clusters under the predetermined circumstances that satisfy the convexity of the energy consumption model.

Zhang, X., Shen et.al [8] Applications like virtual reality, vr technology, and language processing are among the many new ones that have emerged with the growth of the mobile web and the popularity of smart terminals. These applications typically have asset resources and demand a lot of computing and storage resources to operate, which has an impact on the quality of the service. Although the performance of smart terminal processors is constantly improving, they are unable to quickly process high-performance programs, which have a significant negative influence on the user's service encounter.

Rathore, M. M., Attique Shah et.al [9] transportation guarantees to provide the correct evidence at the proper period, location, and device to assist the public in making any transportation-related decisions. The vehicular network and the system of allow are two examples of these two components. To collect data and determine traffic information, a small computer and road sensors are installed at each intersection of the road, such as the volume of cars, average vehicle speed, and traffic conditions intensity. In contrast, the vehicular network infrastructure is utilised to obtain details about each individual vehicle, including its location and speed. Relay nodes, coordinator, gates, integrators, and classifiers are all utilised by the suggested system to connect the two subsystems.

Potts, C. M. et.al [11] the second challenge, data prioritising, requires a researcher or user to effectively comb through a big dataset in search of crucial data. We assume that the dataset is too big to look through manually, necessitating the use of a computer engine. It has to do with how we order and prioritize analysis and search terms. The second objective is about deciding what data to offer a user, whereas the first task is about generating knowledge from data. This interactive component is crucial since experts may become quickly dissatisfied, possibly give up on the system, or make a vital error with potentially serious repercussions if important information is hidden or overshared.

3 Methods and materials

Many other approaches can be used to explain cluster stability; our method computes stability from arguments of distance travelled, velocity variability, and possibility. We carefully consider value stability when choosing a cluster head. In order to communicate with other vehicles, we assume that all vehicles have IEEE 803.12p radio transceivers and GPS systems, which individually allow for the acquisition of position-related data. Each car can use the clustering process in the suggested algorithm. A car notifies its neighbours when it needs help locating a cluster head. If there is no response, the process of creating groups is initiated. One cluster contains all the vehicles that are travelling in the same direction. Each car sends a message to the other members of the group, which includes the relocation, Identification, and velocity. After that, it calculations by arranging the inputs in a neighbourhood array:

- The separation between the actual vehicle Q and its Neighbour P is:

$$Distance_{Q,Y} = \sqrt{\left(\left(M_q - M_p\right) + \left(Z_q - N_p\right)\right)} \tag{1}$$

Where the location of the car is (m, n)

- The difference in speed between vehicle Q and its neighbour U.

$$\Delta U_{q,p} = U_q - U_p \tag{2}$$

- The likelihood that the car is the cluster head:

$$probablity = (F + 3 * density\ of\ the\ node + U)/Q_{max} \tag{3}$$

Where F is the node's power consumption when transmitting or getting a package (here, energy is the electricity a node uses throughout the data communication), and U is the vehicle's speed. Likelihood will be in the range of 1 and 3. The transport density, speed, and range of F have all been normalised (1–3). The greatest value of the calculated probability will be applied to all collected information. The likelihood that a car will act as the cluster head will therefore be calculated each time by dividing the result by the number 3 (Q_{best}); the resulting ranging value of likelihood will have a scale of 1-3. Remember that the value approaches 2 to suggest a poor choice, while the value approaches 1 to indicate the ideal candidate because it would be a problem of reduction; the best is the minimal.

Clusters are built after the cluster heads are chosen, and then communication between the cluster heads begins. It would be more likely for one cluster head to replace another if it used the least amount of energy throughout transmission. Energy in this context refers to the electrical power used by a node during transfer.

$$F = L * F_{com} \tag{4}$$

Where F_{com} describes the wireless send's energy consumption.

- The transmitting chain's RQ,P stability factor calculated for each neighbouring car P is:

$$R_{q,p} = \propto 1, \propto 2 * distance + \propto 3 \nabla Uq, p * Q + \propto 4 \tag{5}$$

Where the numbers are $\propto 1, \propto 2, \propto 3, and \propto 4$ referred to as the real constituents of the model.

$$|\propto 1..\propto 4| = (m^j * m^l) * (R * Y) \tag{6}$$

Where the effects of the matrix are R, output, and Y are representing a cluster's stability. Cluster stability is a significant objective that clustering techniques work to achieve and is a valid metric for assessing the effectiveness of a clustering method. There are several ways to explain cluster durability, and our method computes it based on arguments about distance, velocity variation, and chance. Cluster head is chosen carefully, keeping value consistency in mind. The experiment's findings were utilized to modify each of the three variables (vehicle speed differential, distance U t, and likelihood) within the following ranges:

Table 1. Variable findings

	Small Grade: +2	**Tall Degree: -2**
Expanses	Close	distant
Speediness	Short	great
Probability	Short	great

- Stability R component:

All cars in a cluster exchange stability measurement with their nearby counterparts. The vehicle with the greater value of steadiness is the cluster head.

$$R = \sum R_{p,q} \tag{7}$$

Procedure:

The suggested methodology's first algorithm follows the following steps: The storage space, transmission range, and energy requirements will be used to compile a list of backup mobile edge nodes that could be used. A backup mobile edge-node notifies the cluster through email before moving head alerting it to start a fresh selection process for a new backup node if it left its clusters (took another intersection). Another backup node with a high score will be chosen by the cluster leader from the list of candidates. After a minute, if the handoff is still not working, the backup mobile edge-node welcomes the user. The area has been improved, and the BSI list is handled and kept current.

```
Algorithm 1:
Input:
Vehicle group Yd1 = "Y11, Y12, Y13,..., Y1n"
Output:
The groups of vehicles to be chosen from set Yd1 as
cluster midpoints are Hd1 = "g11, g12, g13,..., g1l"
1. The vehicles travelling in a fixed direction (e1)
   while inside the range of a certain
2. GAM is a member of the set Yd1
3. The cluster midpoints are chosen at random from set
   Yd1 by 'M' cars, and
4. set ld1 sort order
5. Recap
6. Do for each k from 2 to M
7. The cluster's midway is one of the cars ck from set
   Ld1
8. The separation, distance(di, yj), between each yj
   vehicle in set yd1 and di is
9. Calculated
10. In the event when |dj| = 40 && max (distance(di,
    yj)) = d do
11. The vehicles dj with the closest separation from xi
    are connected to
12. Set xi has dj and arranged in it.
13. Complete
14. Complete
15. Till
16. The created clusters' new cluster midpoints are
    determined using to find b)
17. better answer, saved as xi (unique)
18. To create new membership of xi(unique), which
    substitutes for dj, repeat steps 3-7
19. dj adopting the new median, xi
20. Repeat steps 4 through 9 until two novel cluster
    averages, dj (unique), are discovered.
```

Due to the complicated nature of a road network and the frequent entry and exit of vehicles, the connection link is unstable due to the difficulties of mobile nodes that exist with vehicular traffic, and there is a substantial danger that the link will fail [4].

Due to the dynamic nature of our suggested DEBCK, which automatically forms the backup cluster node and assumes control whenever a topology breakdown occurs as a result of vehicle redirections, it was developed to address this issue.

3.1 Mobile edge computing

MEC is a workable solution since it provides on-demand access to high-volume compute. As a result, users at the MEC split the server's processing power. The computational resources allotted to h_q^f by MEC are represented by E n. Similarly, max denotes the maximum allowable computation on the MEC server. Two other factors affect how long it took MEC to complete the work in total: (1) The duration of data transfer from the UEs to the MEC server, as described in (2); (2) MEC http computation. As a result, the following is how long it took to complete the assignment on the MEC server [5]:

$$h_q^f = \frac{\propto_n d_n}{l_m^f} + \frac{\propto_n r_n}{s_n} \tag{8}$$

The continual power from the grid also leads us to believe that the MEC's battery life is limitless. As a result, the power consumption of the Edge servers while it is handling queries is disregarded in this case. Similar to this, the authors of state that we overlook the findings' transmission rate from the Edge servers to UE because of their short size. The fundamental concept behind edge devices is to move the cloud platform from the mobile core network's inside to the mobile communication network's edge in order to achieve flexibility resource consumption. In order to enable high computational and delay programs on edge devices that have limited resources, portable edge computing brings portable computing [12], network management, and storage systems to the edges of the network.

3.2 Formulation of a problem

In this study, we sought to reduce the task's computational time by jointly maximising communications and computing capabilities between the user, MEC, and task division variable. Additionally, for the sake of simplicity, we define {q = q1, q2, qn} {k, = k 1, k 2, k n, and = 1, 2, n}. The computing time minimization problem can therefore be represented formally as follows:

$$Q1\colon \min max_{\propto,q} = \left(\frac{2-\propto_m}{h_m^f}, \frac{\propto_n d_n}{l_m^f} + \frac{\propto_n r_n}{s_n}\right) \tag{9}$$

$$D1\colon \frac{q_m \propto_m r_m}{s_n} + \mathrm{C}_n(2 - \propto_n)c_n \leq f_m^{max,} \nabla_n \tag{10}$$

$$D2\colon \sum_{m=2}^{M} h_q^f \leq f_m^{max,} \tag{11}$$

$$D3\colon q_m \leq Q^{max}, \propto_n \in (1,2), \forall_{m,} \tag{12}$$

$$D4: |\nexists_l| = 2, \nabla_l \tag{13}$$

While constraint (8d) specifies that the computational resources allocated to UEs at the MEC server should be less than the highest computational resources, constraint (7b) ensures that the overall amount of energy used to calculate the task will be less than the highest amount of energy the battery can The optimization problem that is covered in sections (7a) through (7e) is also mixed-integer, non-convex, and non-linear in nature. This is due to the logarithmic function present in the rate equation. We divided the main problem into a number of smaller difficulties in order to handle this challenge. Additionally, the sub-optimization problems are then resolved repeatedly in order to identify the best possible solution. Hold E max n. The transmission power limitations of the UEs are represented by restriction (5h), in a similar manner. Then, pitch shift control of IRS components is provided by (6h).

3.3 The system model

In the Mobile nodes, clustering is the technique of assembling nearby moving cars on a street into stable groups to speed up information sharing between vehicles. A roadway model with a lot of automobiles moving in one direction is taken into consideration. The network is made up of several vehicles that are located on the street and are regarded to be members of various groups, with one of the members of each group acting as the head of the group. Presuming that vehicles can connect with one another to exchange information, such as safeguard texts, the network is made up of vehicles that assume this assumption (CH). A number of characteristics, including position, velocity, and steadiness, define the CH. Regardless of the speed and density of the cars, all groups evolve over time because a new group leader is always selected. The message was sent over the network via the cluster head from one cluster head vehicle to another cluster head vehicle in the neighbouring cluster until it reached the destination vehicle. Clustering model illustration is shown in Figure 2 [6].

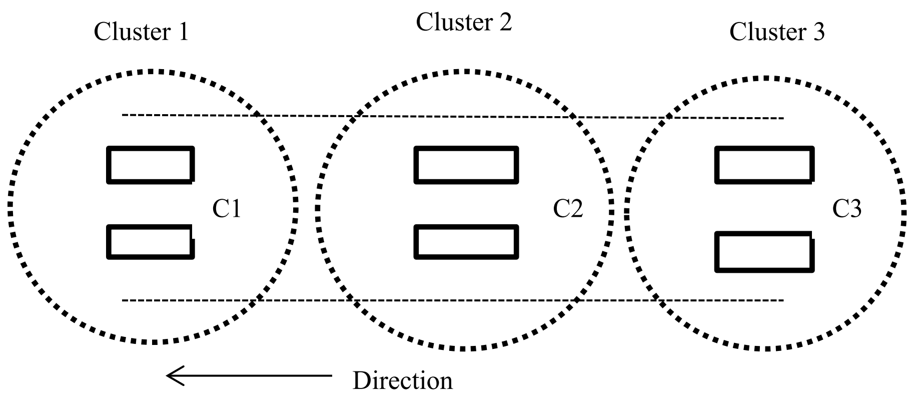

Fig. 2. Clustering model illustrations

3.4 Cluster head selection and cluster composition

The network starts out as a collection of automobiles on a route, split up into several vehicle groups within its communication range. The cluster head (CH), which serves as the group's leader, is symbolized by the number of vehicles that make up each cluster (CM). The first CH is selected at random, and the second cluster head is decided upon as the vehicle that is placed furthest away from the target vehicle inside the first head's range. All vehicles that are in the head's field of vision are regarded as belonging to the group. The steps in cluster composition are shown in Figure 3.

Fig. 3. Steps in cluster composition

3.5 Method for selecting cluster heads

The proposed LEC-steps SEP's are shown in below Method. The CHs broadcast themselves to the network after being chosen by the system in Figure 4. The non-cluster head nodes decide to join the closest cluster after receiving the message and joining the cluster. The following formula describes how to cluster data [10]. CH number is the optimal cluster set as determined by the calculations, CH member is a cluster member node, and CH count is the total number of clusters in the current round.

Algorithm: Cluster head selection and cluster composition:

```
1. All vehicles must be on the roadway
2. idly set each node's track
3. determine each vehicle's speed
4. choosing the first CH
5. Identify the sending vehicle
6. The first cluster should contain all vehicles within
   the FCH's range.
7. as long as (nodes!=unfilled)
8. find the following CH
9. All nodes + node clustering
10. Clusters are updated by changing the CH during the
   designated time.
11. Finish while
12. finish
```

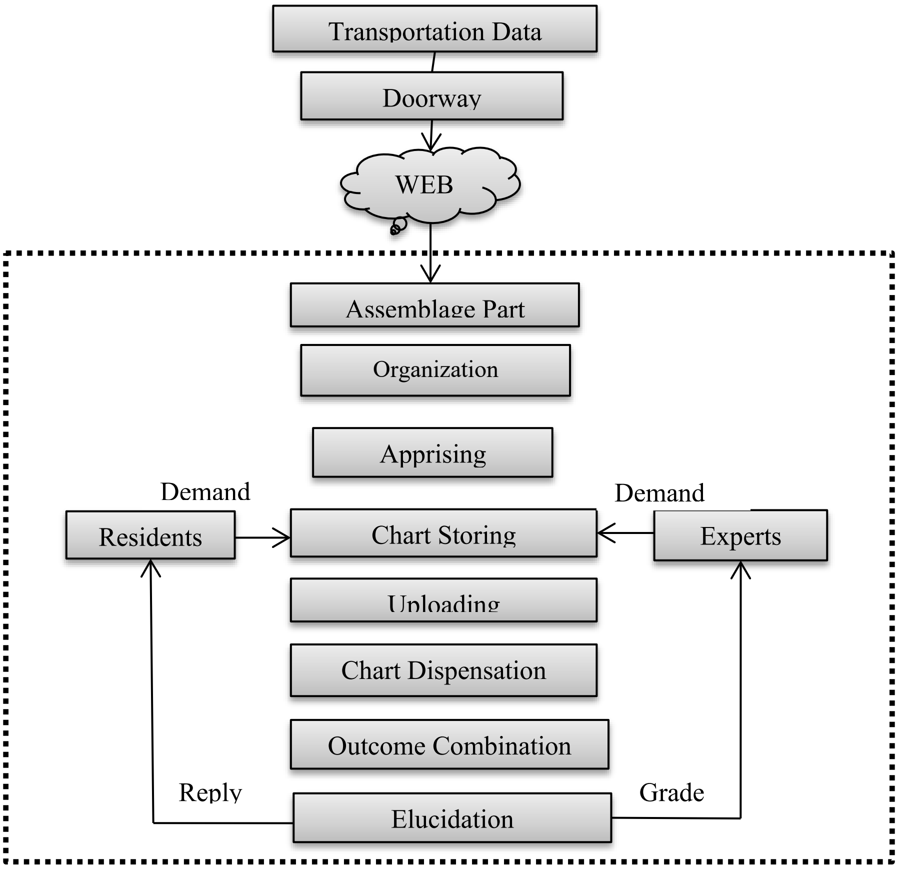

Fig. 4. Design of the suggested system's deployment for intelligent transportation

4 Implementation and experimental results

A MATLAB simulator is used to examine the energy usage according to the number of clusters in order to evaluate the effectiveness of the proposed Pectorals. The effectiveness is assessed using a random system made up of 150 servers that are distributed randomly around a normalized circular area with mg flows [7]. User job characteristics, channel capacity disparities across base stations, and accessible computing resources from mobile edge computing servers are carefully taken into account while keeping in mind user fairness for mobile systems that allow dense connectivity [8].

Table 2. Determining the Multi-Stage Asset Allocation Server's Simulation Parameters

Limitations	Assessment
capacity for base stations	550
D2D link speed	100
The processing power of an MEC server	40/SEC
The quantity of computing resources needed to finish the task	[0.8-0.9]cycles
The capability of Server J	mg
factor in heaviness	1.5
The ability to compute(mg)	[0.11-0.13]cycles

The suggested method first analyses the traffic dataset that was obtained from Aarhus, Denmark's second-largest city by people. An investigation of how traffic density affects vehicle speed is shown in Figure 5. When the transportation intensity is higher, that is, when there are more between two locations on the road, the average vehicle speed is seen to be meaningfully lower between any two road sections with a resolution of 500 m. Similar to this, the contrary is also seen, namely that the average speed increases as fewer cars are on the road. The graph's unfilled circle line illustrates how the average speed drops dramatically during different times of the day when there are a substantial number of cars (550–650). However, at times when there aren't many cars (1 000–4 000), yet the observed average speed of vehicles is considerable, as indicated by the full circle line on the graph. These situations do not always occur, and some abnormalities, like a relative low seed despite a low automobile intensity, can also be seen. These circumstances are primarily attributed to causes like construction on the roads, incidents, or weather conditions like rainfall, foggy, etc [9].

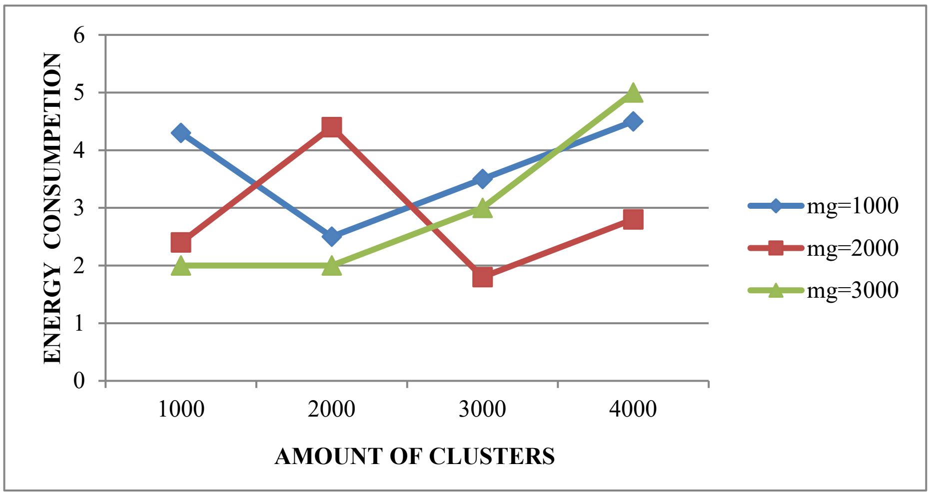

Fig. 5. When mg = 1000, power consumption of MEC is dependent on the cluster centres

This is due to the fact that handling focused requests by a small number of servers while leaving a large number of others idle uses more energy than handling fairly spread queries by all servers using PECS. If m is small, a small number of mg servers handle all requests for m servers, and many member servers evenly split requests between them. Since, in our system model, queries to mg servers and to member server are constant. Thus, as shown in Figure 5, the load demand increases when k is small and reduces as mg increases as a result of the distribution of load among MEC servers. Beyond a certain threshold, however, the concentration of demands on a limited number of members causes the energy usage to tend to increase. For instance, the MEC's computational capabilities may not be available, or the offload may not be carried out if it is not profitable [13].

When mg = 550 in Figure 6, the dynamic positioning of MVEs causes an increase in power usage. The start of the graph is eliminated because l ought to be a part of mg. Because more requests result in increased CPU demand, PECS similarly grows as the amount of flows does.

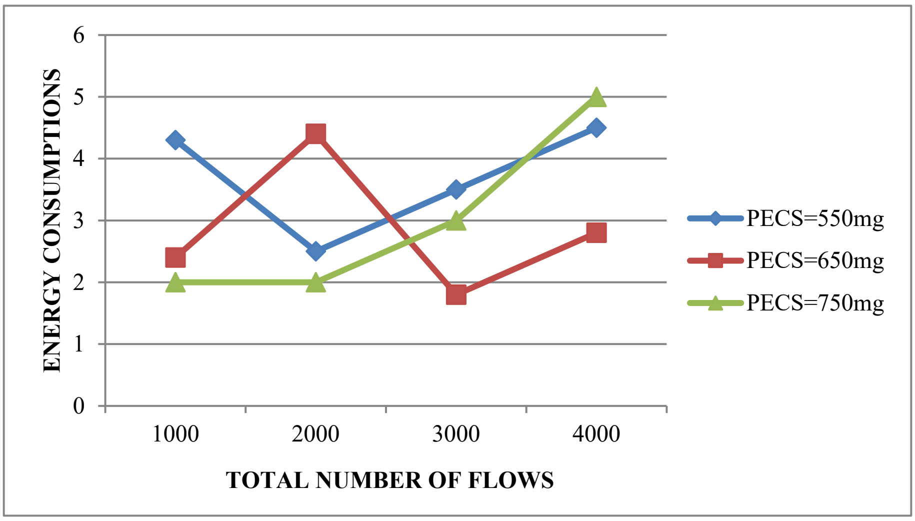

Fig. 6. Comparison of energy use when mg = 550

In Figure 7 and 8, researchers contrast the PECS program's average latency and power usage with that of two instances of PECS providing 5G service with Treq of 550 and 1000 mg, correspondingly. Pecs(l) rises in tandem with the number of flows. When the k value that minimises the average delay deviates from K, we identify a replacement mg value for l that is somewhat close to the original k value. The task execution overhead of the system continues to decline as the MEC server's processing capacity rises [8]. This is the cause of the saw-tooth pattern on the PECS graph. The best performance for PECS occurs when Treq is 600ms. A 13.34% reduction in power usage is attained with PECS when mg is 4000.

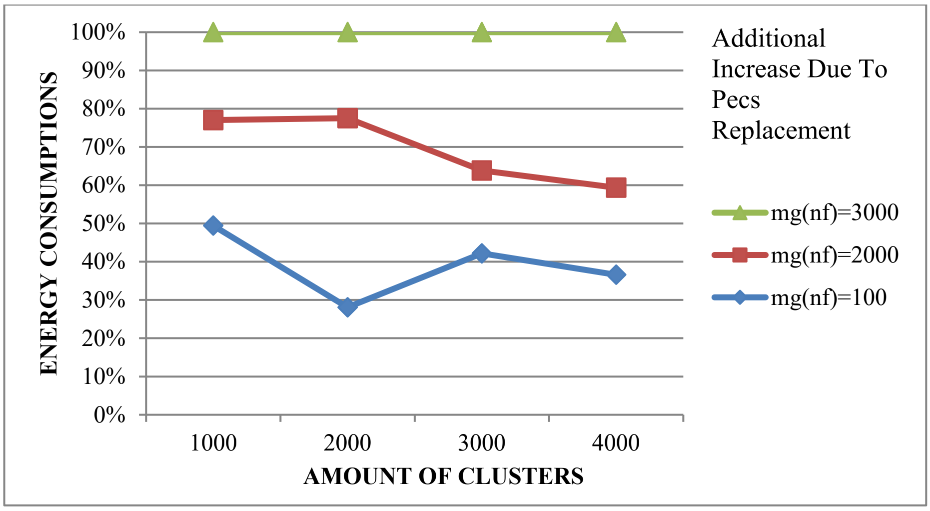

Fig. 7. When mg = 2000, the power usage of PECS is determined by the number of clusters

The overall latency of inputs processed by PECS servers, mg, is sustained by PECS at the acceptable level, T(request), independent of the mg value, as shown in Figure 8. Because PECS searches for the average delay is smaller than L (request) when there are few flows in the MEC environment, the optimal mg to minimise energy usage in these circumstances has a longer average delay than the PECS technique.

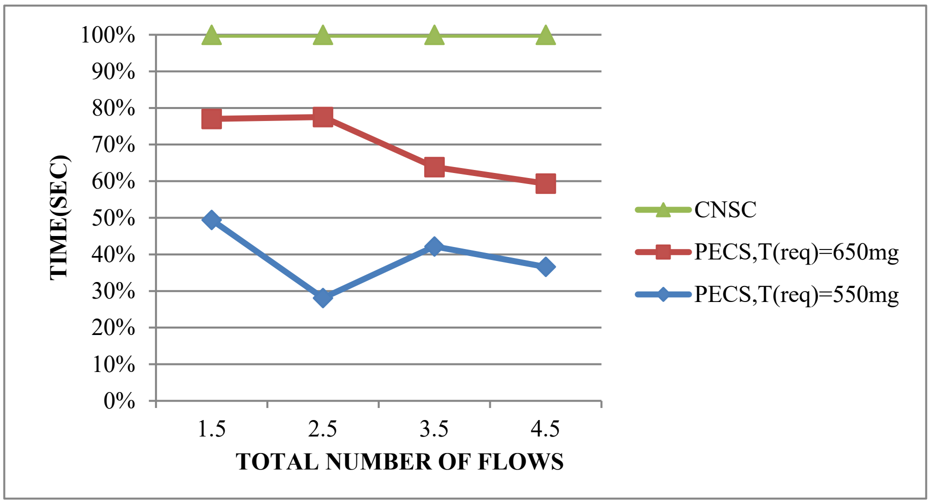

Fig. 8. When mg = 650, compare the average delay

5 Conclusion

The country's economy and commuters' social lives are both strongly impacted by transportation systems. This study concentrated on the tasks of traffic analysis and prediction, and it provides a current collection of accessible datasets and tools as a resource for people looking for open-source materials. In order to more accurately estimate and predict traffic states, it is also advised to collect and use external data, such as weather forecasts, calendar details, air reduced pollution, and sound. The simulation findings demonstrate that, when the suggested algorithm is less impacted by fluctuations in the number of cluster heads, it can more efficiently extend the stability period and transfer more data.

6 References

[1] Rathore, M. M., Attique Shah, S., Awad, A., Shukla, D., Vimal, S., & Paul, A. (2021). A cyber-physical system and graph-based approach for transportation management in smart cities. *Sustainability*, *13*(14), 7606. https://doi.org/10.3390/su13147606

[2] Uma, D., Udhayakumar, S., Tamilselvan, L., & Silviya, J. (2020). Client aware scalable cloudlet to augment edge computing with mobile cloud migration service.

[3] Sarfraz, M., Alshahrani, H. M., Tarmissi, K., Alshahrani, H., Elfaki, M. A., Hamza, M. A., & Khurshaid, T. (2022). Intelligent Reflecting Surfaces Enhanced Mobile Edge Computing: Minimizing the Maximum Computational Time. *Sensors*, *22*(22), 8719. https://doi.org/10.3390/s22228719

[4] Liu, L., Zhao, Y., Qi, F., Zhou, F., Xie, W., He, H., & Zheng, H. (2022). Federated Deep Reinforcement Learning for Joint AeBSs Deployment and Computation Offloading in Aerial Edge Computing Network. *Electronics*, *11*(21), 3641. https://doi.org/10.3390/electronics11213641

[5] Sarfraz, M., Alshahrani, H. M., Tarmissi, K., Alshahrani, H., Elfaki, M. A., Hamza, M. A., & Khurshaid, T. (2022). Intelligent Reflecting Surfaces Enhanced Mobile Edge Computing: Minimizing the Maximum Computational Time. *Sensors*, *22*(22), 8719. https://doi.org/10.3390/s22228719

[6] abd Al_kadum Hassan, H., Hasan, Z. Y., & Al Taie, R. H. (2022). A simulation approach to improve the VANETs communication. *iJIM*, *16*(12), 137. https://doi.org/10.3991/ijim.v16i12.31423

[7] Ahn, J., Lee, J., Park, S., & Park, H. S. (2019). Power efficient clustering scheme for 5G mobile edge computing environment. *Mobile Networks and Applications*, *24*(2), 643-652. https://doi.org/10.1007/s11036-018-1164-2

[8] Zhang, X., Shen, H., & Lv, Z. (2021). Deployment optimization of multi-stage investment portfolio service and hybrid intelligent algorithm under edge computing. *PLoS One*, *16*(6), e0252244. https://doi.org/10.1371/journal.pone.0252244

[9] Rathore, M. M., Attique Shah, S., Awad, A., Shukla, D., Vimal, S., & Paul, A. (2021). A cyber-physical system and graph-based approach for transportation management in smart cities. *Sustainability*, *13*(14), 7606. https://doi.org/10.3390/su13147606

[10] Du, R., Liu, Y., Liu, L., & Du, W. (2020). A lightweight heterogeneous network clustering algorithm based on edge computing for 5G. *Wireless Networks*, *26*(3), 1631-1641. https://doi.org/10.1007/s11276-019-02144-x

[11] Potts, C. M. (2022). Interactive Data Analysis for Experts: Tools, Techniques, and Assessments.
[12] Wang, T., Lu, Y., Cao, Z., Shu, L., Zheng, X., Liu, A., & Xie, M. (2019). When sensor-cloud meets mobile edge computing. *Sensors*, *19*(23), 5324. https://doi.org/10.3390/s19235324
[13] Mach, P., & Becvar, Z. (2017). Mobile edge computing: A survey on architecture and computation offloading. *IEEE communications surveys & tutorials*, *19*(3), 1628-1656. https://doi.org/10.1109/COMST.2017.2682318
[14] Julian Menezes R, Albert Mayan J, Breezely George M. (2015). Development of a Functionality Testing Tool for Windows Phones. *Indian Journal of Science and Technology, 8*(22), 1-7. https://doi.org/10.17485/ijst/2015/v8i22/79101

7 Authors

Dr. J. Albert Mayan is an Associate Professor in the School of Computing, Sathyabama Institute of Science and Technology, India. With 21 Years of experience in teaching and R&D, Dr. Albert had published a number of papers in Indexed Journals. Dr Albert had presented various research-based papers at several national and international conferences. Dr Albert had obtained his Ph.D in the Field of Software Testing and his areas of interest include Machine Learning, Image processing and Data Mining. Dr Albert is an active member of ACM Student Chapter and OWASP Student Chapter at Sathyabama Campus (email: albert.cse@sathyabama.ac.in).

Dr. S. V. Manikanthan is a Director of Melange Academic Research Associates, Puducherry, India. His area of Research Interest is Wireless Sensor Networks. He has 21 years of experience in Anna University Affiliated Colleges and in Industrial Research Projects (email: prof.manikanthan@gmail.com).

Azham Hussain is the Associate Professor of Software Engineering at UUM School of Computing. He is the founder and head of Human-Centered Computing Research Group which is affiliated with the Software Technology Research Platform Center at School of Computing, Universiti Utara Malaysia. Assoc. Prof. Azham Hussain is a member of the US-based Institute of Electrical and Electronic Engineers (IEEE), and actively involved in both IEEE Communications and IEEE Computer societies. Azham is published in the areas of software evaluation and testing, user behaviours, group collaboration, and ubiquitous and mobile technology design. He has authored and co-authored more than 100-refereed technical publications, served as reviewer and referee for refereed journals and conferences on computing as well as the examiner for more than twenty doctoral and postgraduate scholars in his research areas (email: azham.h@uum.edu.my).

Dr. S. Nithyaselvakumari presently working at Saveetha school of Engineering as Assistant professor SG in the Department of Medical Instrumentation and experienced around 11 years in teaching and 5 year in Industry. She has completed Bachelor of Engineering in Electronics and communication Engineering at Mahendra Engineering college Salem under madras university and done Master of Engineering in Sathyabama university in the department of Applied Electronics. In 2021, she has completed Ph.D under Anna University. Her area of research is in Bio Medical, wireless communication, Embedded system and Internet of Things (email: nithyaselvakumari.s@gmail.com).

A. Vinnarasi is an Assistant professor at the Department of ECE, SRMIST, Chennai. Doing PhD.in SRMIST with total working experience of more than 17 yrs. M. Tech in Embedded Systems and Technology, in SRM 2005. Completed BE in ECE 2002. Pursuing PhD in STMIST. Worked in Saveetha Engineering College as Lecturer for 1yr 6 months. Worked in Sri Muthukumaran Institute of Technology as Assistant professor for 4yrs 5 month. Currently working in SRMIST as Assistant professor till date with 12yrs experience (email: vinnaraa@srmist.edu.in).

Article submitted 2022-11-05. Resubmitted 2022-12-16. Final acceptance 2022-12-28. Final version published as submitted by the authors.