A cluster formation algorithm for fog architectures based on mobility parameters at a geographically LAN perspective

Victoria B. Martins, Douglas D. J. de Macedo, Laércio P. Jr., Roger Immich

Abstract As Internet of Things (IoT) becomes popular, the different approaches to increasing their quality also do. One of the used paradigms to enhance these applications is fog computing. The fog intends to bring computational power closer to the users (edge). This paradigm is known to mitigate costs and energy consumption and also to benefit location-aware applications. As fog environments can cover small to medium areas, these can be used to increase location awareness. To make it possible, researchers have used cluster computing. However, in new scenarios, cluster formation can be a challenge since when manually set, geographical-location parameters can not be interpreted correctly. In this manner, this paper aimed to promote a cluster formation algorithm based on these geographical parameters. To evaluate our proposal, we compared our approach to the original using the standardized EUA dataset through iFogSim v2. The proposed algorithm was capable of creating clusters based on accepted node range and maximum nodes per cluster, even though it had lower performance.

Preprint version

Victoria Botelho Martins
Federal University of Santa Catarina (UFSC), Santa Catarina, Brazil,
e-mail: victoria.botelho@posgrad.ufsc.br

Douglas Dyllon Jeronimo de Macedo
Federal University of Santa Catarina (UFSC), Santa Catarina, Brazil,
e-mail: douglas.macedo@ufsc.br

Laércio Pioli Júnior
Federal University of Santa Catarina (UFSC), Santa Catarina, Brazil,
e-mail: laercio.pioli@posgrad.ufsc.br

Roger Immich
Federal University of Rio Grande do Norte (UFRN), Rio Grande do Norte, Brazil,
e-mail: roger@imd.ufrn.br

1 Introduction

Within four years, the global IoT market will reach more than one trillion dollars [5]. To minimize costs, service providers have been enhancing quality of service alongside service level agreements. Among applications made possible by IoT, there are location systems, which, addressing tags to their assets have been capable of understanding their behaviors [12], [10]. Although some of them utilize target Veicular Ad Hoc Networks (VANETS) [14] they can also be implemented in smaller surroundings [6].

Location systems heavily rely on mobility patterns, which involve a set of variables like speed, direction and orientation [4]. These systems have been used not only to track assets but also in transportation systems [11]. To take a closer look into these applications, fog based architectures have been used, since they can process, compute and deliver information within the network, costing considerably less than cloud-only approaches [3].

When it comes to an area, clusters, can be organized as regions, or even rooms in indoor systems, at the edge, in such way it is given more control and location awareness of the services and the users. However, clusters tend to be formed manually by the stakeholders carrying out some tendencies towards one or multiple areas. Therefore, our work aim to answer if **is it possible to organize clusters based on geographical and location parameters**. This way, this paper carries the following contributions: (i) an implemented algorithm to set up clusters using geographical parameters, (ii) a customized dataset of user mobility based on EUA dataset.

The structure of this paper is as follows: section 2 provides an overview of the related work about clustering formation algorithms, section 3 the proposed experiment with further details about the algorithm proposed, the simulated environment and the datasets, by the section 4, the reader can find the evaluation and analysis of the generated vs the original clusters and finally section 5 arranges the conclusions and future work.

2 Related Works

In this section, we present an overview of the related work, aiming location-based algorithms. Table 1 presents a comparison regarding the network used, type of client, type of algorithm, when it comes to its dynamicity and formation, and finally if the scenario is or not urban.

In [1], the authors presented an algorithm for clustering the wireless mobile adhoc networks (MANET) considering the network mobility characteristics. As proposal, it is presented a learning automata-based weighted cluster formation algorithm called MCFA in which the mobility parameters of the hosts are assumed to be random variables with unknown distributions. Using information received from its neighbors, MCFA was superior to the other presented mobility-based clustering algorithms in terms of the number of clusters, cluster lifetime, reaffiliation rate, and control message overhead.

Communication within clusters, has also been studied. Paper [9] was capable of reducing delay that energy sustainability could be accomplished through implementing a clustering algorithm on the intra-distance inter-distance between the CH and nodes. The authors developed an improved IoT-Wireless Sensor Network (WSN) model and a modified Rider Optimization Algorithm (ROA). The evaluation was based on alive nodes and normalized energy.

Still into green computing, the authors in [15] proposed an algorithm to mitigate the large number of energy voids or hotspots in WSNs, called Energy-efficient Cross-layer-sensing Clustering Method (ECCM) using fog computing. The algorithm is capable of electing a group of optimal nodes as the cluster heads, without cost competition overhead, therefore effectively reducing and and balancing the ecosystem.

A multi-head clustering algorithm that utilizes mobility-based metrics to form clusters in a vehicular network (VANET) was proposed in [14]. In this paper, the algorithm assumes that each vehicle can be a member of multiple clusters, thus creating stable clusters that are robust in this environment. The performance evaluation is based on metrics regarding the clusters' lifetime. Positive differences were noted when it came to cluster stability and size.

Highly dynamic scenarios like urban scenarios can also benefit from cluster computing. The paper [11] develops a mobility-based algorithm for VANETS, using vehicle's moving direction, relative position and link lifetime estimation.

Internet of Things is another environment that has been an application to clustered systems. Using the fog paradigm, the authors in [7] used the concept of federated fogs along with Genetic Algorithm and Machine Learning models. As matter of subject, this paper targeted quality of service and time-sensitive applications which showed notable improvement in some QoS requirements.

Our proposal differs from the others mainly because it relies on geographical data regarding the fog resources. Also, the mobile agent in our case are smartphones carried by people connected to an IoT-Fog location based system, instead of vehicles in VANETs.

Table 1: Comparison of the proposed to the related works

	MANET	**VANET**	**User-walk**	**Fog**	**Adaptive clusters**	**Location-based formation**	**Urban scenario**
[1]	X		X		X	X	
[2]	X	X			X	X	
[8]					X	X	
[11]		X			X	X	X
[14]		X			X	X	X
[15]				X		X	
Our proposal			X	X		X	X

3 Experimentation setup

In this section, details about the architecture, application and the algorithm is presented, in addition to all the fundamental elements. The experiment is entirely available [1].

3.1 Algorithm

We developed an algorithm that creates clusters based on accepted node range (ANR) and maximum nodes per cluster (MAX). As input, the user can provide a .csv file based on EUA dataset. If the user does not provide these vales, by default, the algorithm automatically calculates them. The Algorithm 1 presents a pseudo-code perspective of our proposal.

The maximum number of nodes per cluster is defined as the area divided by the number of nodes and the range, the area divided by the number of meters per node. The algorithm is divided in two steps: formation and optimization.

In the formation step, for each node presented in the .csv file, the location of the current node is compared to the next based on the ANR. If this comparison is true, the next node is assigned to the current's node cluster. If the group already have reached the MAX, a new set of cluster is created.

As for the optimization part, nodes unassigned are ignored as well as the clusters formed by only one node, since for each cluster, it is needed one orchestrator node (proxy) and at least one other computing node (gateway). Then, for each cluster, the first node is defined as proxy. At the end of the optimization step, the algorithm finally links the selected group with their respective coordinates, therefore providing the .csv file. This last part is not presented in Alg 1 for the sake of simplicity.

3.1.1 Limitations

As mentioned before, our algorithm organizes clusters based on a range (for each node), calculated automatically and omits clusters with less than two nodes, meaning that there can be places that even though the whole area is covered, the closest node can be locally far. By the simulator default behavior, if the users' distance is greater than the ANR, the user will still be connected to the closest cluster.

Another limitation is the fact that the entire calculation depends on the user input, that is the algorithm does not support individual suggestion of locations for each node presented. In this way, testing clusters would turn to be a careful process.

[1] https://github.com/vyk1/cluster-formation

Algorithm 1: Proposed cluster formation algorithm

```
Input  : Array containing fog coordinates
Output: Array containing fog coordinates and regions
  metersPerNode ← AREA/nodes;
  range ← AREA/metersPerNode;
  maxNodesPerCluster ← nodes/metersPerNode;
  clusters ← null;
  {FORMATION}
  for node in nodes do
    if clusters.size() is null then
      adds node to clusters and sets responsible as 0
    else
      if responsible is null then
        tries to get responsible for node and breaks
      end if
      if responsible is still null then
        adds node to clusters
        sets responsible as clusters.size() - 1
        adds node to added array
      end if
    end if
    for node in nodes do
      if nextNode is equal to current then
        continue;
      end if
      if added contains current then
        continue;
      end if
      if the size of clusters[responsible] is less or equal to the maxNodesPerCluster then
        if distance between current and nextNode is within range then
          adds nextNode to clusters[responsible]
          adds nextNode to added array
        end if
      else
        if distance between current and nextNode is within range then
          adds node and nextNode to clusters with node as responsible
          adds node and nextNode to added array
        end if
      end if
    end for
  end for
  {OPTIMIZATION}
  for responsible in clusters do
    if the size of clusters[responsible] has more than one node then
      adds group to selected array
    end if
  end for
  return selected
```

3.2 Environment

Using the iFogSim v2 simulator [13] and the EUA dataset [2] with predefined edge resources (fog nodes) locations, the testing environment uses two sets of variables: user related and node-related. User-related variables were regarding to their interactions, which means their walking path during the simulation. In our case, the nodes' variable were only related to their region (predefined), as all fog nodes had the same computing power such as RAM, MIPS, uplink and downlink.

The clusters generated by the proposed algorithm were compared to the default clusters provided by the EUA dataset. To compare the performance of each, there were used two different datasets: directional and randomized. In the first, each user had a different waypoint following a ”walk-logic”, which means that each user trailed the street directions. However the randomized dataset was generated by the simulator from one waypoint, placing locations in anyway within a preordained maximum distance. In this dataset, there were sets of one, five, ten, fifteen and twenty users. As for the directional, one, five, ten and fifty.

Consider the users' smartphones are connected to the LAN, which provides a location-based system. As shown in Fig. 1, for the sake of simplicity, we used the basic 3-step task: request-process-respond. The storage module is the part that needs connection to the cloud. The entire experiment is available [3].

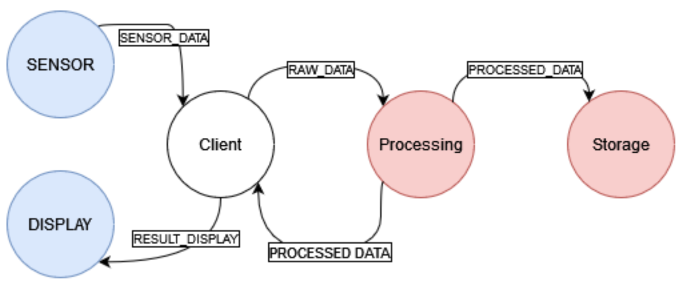

Fig. 1: Application modules

The architecture here presented follows the 3-tiered cloud-fog-edge model. On the bottom of the architecture, the users, using their smartphones are connected to the network environment. These users send their raw location data based on the nearest Wi-Fi waypoint to the fog. In the fog-layer, the processing happens inside the cluster in which the user is connected. By the end of processing, the data is sent back to the edge, so that data is displayed to the user in the app and to the cloud (0-tier) for storing.

---

[2] https://github.com/swinedge/eua-dataset

[3] https://github.com/vyk1/cluster-formation

4 Evaluation Results

The Fig. 3 displays the original EUA dataset for edge resources and Fig 2 shows the clusters suggested by the algorithm. The yellow markers represent the proxies, and therefore the number of regions; the grey the cloud abstract representation and the dark blue the gateways.

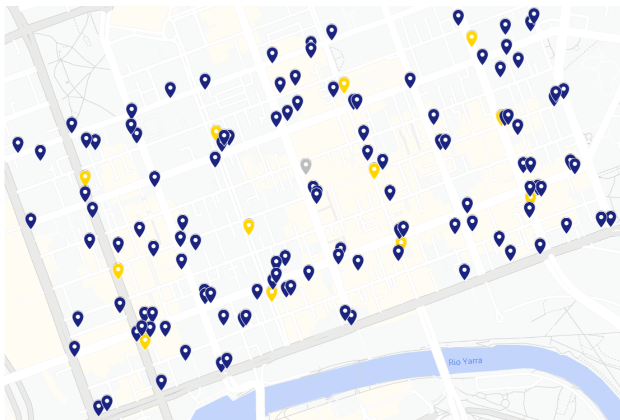

Fig. 2: original clusters

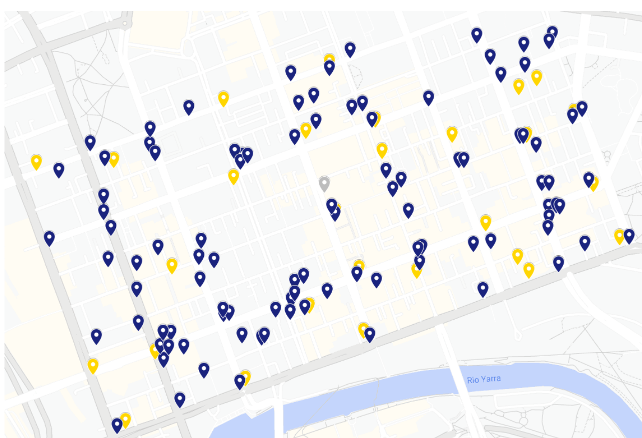

Fig. 3: Generated clusters

The configurations were: the original EUA dataset (original), using 12 regions and the generated clusters (generated) by the proposed algorithm, which covered

28 regions. We ran each configuration 10 times to ensure the ALD using standard deviation.

To evaluate the configurations we used application loop delay (ALD), migration time (MT) and network usage, which are:

ALD expresses the performance time overall
MT migration performance and specifically delayed by the application modules, triggered by mobility induced (user)
NU indicates how much bandwidth the architecture used

These users' interactions were experimented using two different datasets, as it follows.

4.1 Randomized Dataset

As in the random-walk dataset the users always started in the same point, their interactions were concentrated in southern regions. Comparing the original and generated datasets, both had a similar ALD performance, since these fog nodes also were fairly near each other, as shown in Fig. 4. For the 50 users random experiment, we generated an excluive dataset.

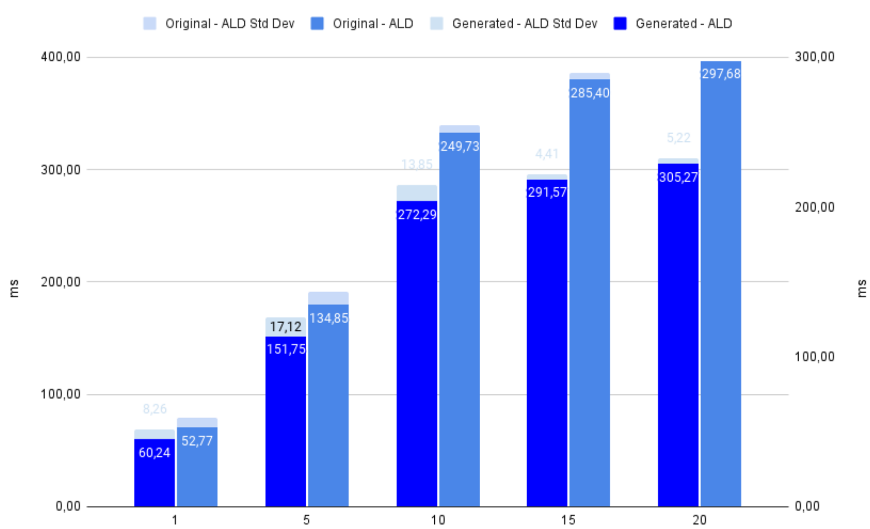

Fig. 4: ALD in original and generated clusters using randomized dataset vs number of users

In the process of creating more clusters in comparison to the original, it is possible to note that for the generated clusters, the migration time was also greater, Fig. 5.

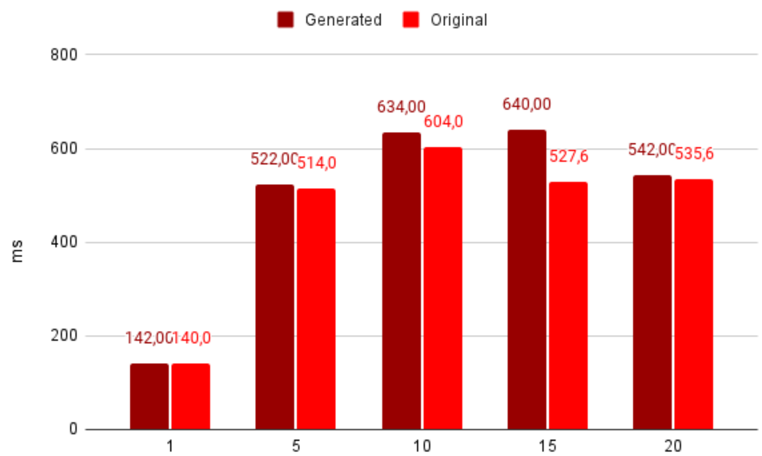

Fig. 5: MT in original and generated clusters using randomized dataset vs number of users

4.2 Directional Dataset

The directional dataset aimed to be more realistic, since the users start from any direction and follow a logic. Even though the original clusters had a better overall performance, it is possible to observe a consistent relation between the ALD and its standard deviation for both datasets. However, the high ALD is a direct consequence for the elevated number of small clusters, Fig. 6.

On the subject of MT, Fig 7, both datasets presented an almost identical performance. This demonstrates how the aspect of speed can impact the application, once in this dataset each interaction was closer to each other and users moved slower than the randomized dataset.

5 Conclusions and Future Work

As location based systems heavily depend on mobility aspects, region-based approaches can enhance these applications increasing quality of service and reducing

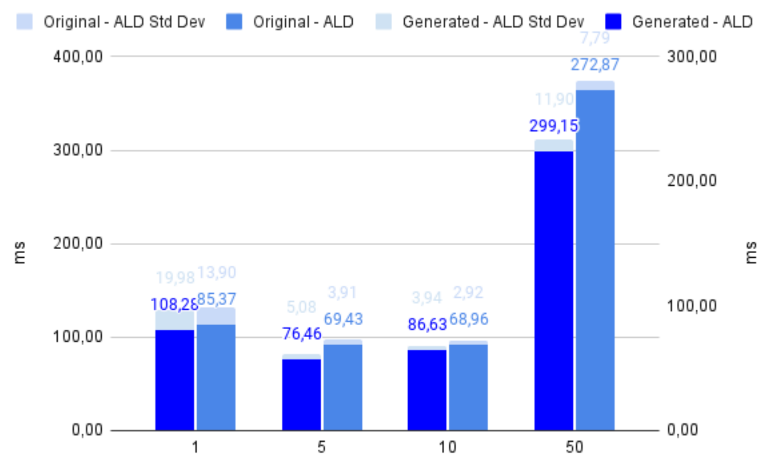

Fig. 6: ALD in original and generated clusters using directional dataset vs number of users

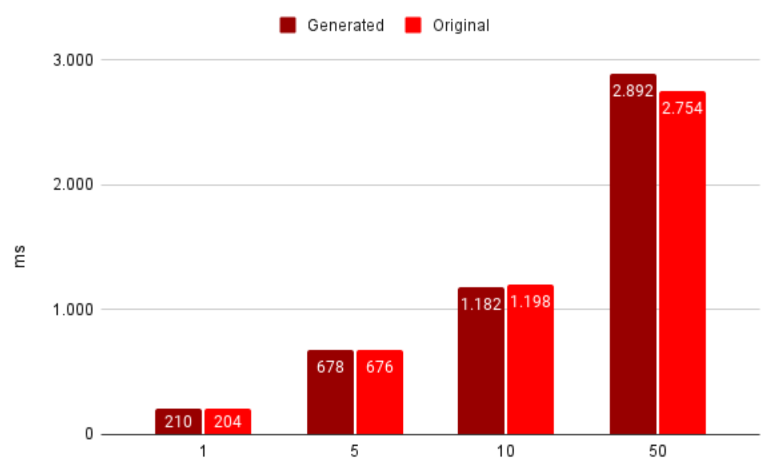

Fig. 7: MT in original and generated clusters using directional dataset vs number of users

computational and financial costs. This way, this study presented an algorithm that can help defining clusters through geographical parameters, such as the location data, range and area. Although the clusters generated did not have a better outcome to all cases, the performance was similar. And for other studies can be used to any other fog node locations, since it is already integrated in the simulator.

As results it was possible to compare the performance of both generated and original clusters. Even with twice the number of clusters, the generated one was capable of handling the tasks, without doubling the amount of computational resources.

For future work, a possibility to the creation of the large number of clusters is to join them depending on a threshold, like range in meters and/or the number of fog nodes in the cluster, reducing the number of clusters. Also, it would be possible to add a weight feature, for example, based on previous datasets, the algorithm could be enhanced to prioritize regions and therefore expanding certain clusters.

References

1. Akbari Torkestani, J. & Meybodi, M. A mobility-based cluster formation algorithm for wireless mobile ad-hoc networks. *Cluster Computing*. **14**, 311-324 (2011,12), https://doi.org/10.1007/s10586-011-0161-z.
2. Basu, P. et al., ”A mobility based metric for clustering in mobile ad hoc networks” in International Conference on Distributed Computing Systems Workshops, pp. 413–418, 2001, doi: DOI: 10.1109/CDCS.2001.918738.
3. Bonomi, F., Milito, R., Zhu, J. & Addepalli, S. Fog Computing and Its Role in the Internet of Things. *Proceedings Of The First Edition Of The MCC Workshop On Mobile Cloud Computing*. pp. 13-16 (2012), https://doi.org/10.1145/2342509.2342513
4. Carlo Puliafito, Diogo M. Gonçalves, Márcio M. Lopes, Leonardo L. Martins, Edmundo Madeira, Enzo Mingozzi, Omer Rana, Luiz F. Bittencourt, MobFogSim: Simulation of mobility and migration for fog computing, Simulation Modelling Practice and Theory, Volume 101, 2020, 102062, ISSN 1569-190X, https://doi.org/10.1016/j.simpat.2019.102062.
5. Elarbi Badidi, Awatif Ragmani, An Architecture for QoS-Aware Fog Service Provisioning, Procedia Computer Science, Volume 170, 2020, Pages 411-418, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.03.083.
6. Guo, Y., Zhao, L., Wang, Y., Liu, Q. & Qiu, J. Fog-Enabled WLANs for Indoor Positioning. *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*. pp. 1-5 (2019)
7. H. Shamseddine et al., ”A Novel Federated Fog Architecture Embedding Intelligent Formation,” in IEEE Network, vol. 35, no. 3, pp. 198-204, May/June 2021, doi: 10.1109/MNET.011.2000505.
8. Lin, C. et al., ”Adaptive clustering for mobile wireless networks” in IEEE Journal on Selected areas in Communications, vol. 15, pp. 1265–1275, 1997, doi: DOI: DOI: 10.1109/49.622910.
9. Mamoun Alazab, Kuruva Lakshmanna, Thippa Reddy G, Quoc-Viet Pham, Praveen Kumar Reddy Maddikunta, Multi-objective cluster head selection using fitness averaged rider optimization algorithm for IoT networks in smart cities, Sustainable Energy Technologies and Assessments, Volume 43, 2021, 100973, ISSN 2213-1388, https://doi.org/10.1016/j.seta.2020.100973.
10. Meunier, B., Pradel, P., Sloth, K., Cirié, C., Delval, E., Mialon, M. & Veissier, I. Image analysis to refine measurements of dairy cow behaviour from a real-time location system. *Biosystems Engineering*. **173** pp. 32-44 (2018), https://www.sciencedirect.com/science/article/pii/S1537511017302179, Advances in the

Engineering of Sensor-based Monitoring and Management Systems for Precision Livestock Farming.

11. M. Ren, L. Khoukhi, H. Labiod, J. Zhang and V. Veque, ”A new mobility-based clustering algorithm for vehicular ad hoc networks (VANETs),” NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium, 2016, pp. 1203-1208, doi: 10.1109/NOMS.2016.7502988.
12. Pešić, S., Tošić, M., Iković, O., Radovanović, M., Ivanović, M. & Bošković, D. BLEMAT: Data analytics and machine learning for smart building occupancy detection and Prediction. *International Journal On Artificial Intelligence Tools*. **28**, 1960005 (2019)
13. Redowan Mahmud, Samodha Pallewatta, Mohammad Goudarzi, Rajkumar Buyya, iFogSim2: An extended iFogSim simulator for mobility, clustering, and microservice management in edge and fog computing environments in Journal of Systems and Software, vol. 190, 2022, https://doi.org/10.1016/j.jss.2022.111351.
14. S. Vergis, G. Tsoumanis and K. Oikonomou, ”A Proposed Multi -Head Clustering Algorithm for VANET Environments,” 2021 6th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM), 2021, pp. 1-8, doi: 10.1109/SEEDA-CECNSM53056.2021.9566263.
15. Z. Sun et al., ”An Energy-Efficient Cross-Layer-Sensing Clustering Method Based on Intelligent Fog Computing in WSNs” in IEEE Access, vol. 7, pp. 144165-144177, 2019, doi: 10.1109/ACCESS.2019.2944858.