Adaptive Virtual Clustering Methods for Dynamic IoT Edge Systems

Hanan Mahmood Shukur Ali
Al-Kitab University
Kirkuk, Iraq
hanan.m.shukur@uoalkitab.edu.iq

Sirwan Kareem Jalal
Al-Qalam University College
Kirkuk, Iraq
sirwan.jalal@alqalam.edu.iq

Maher Waleed Saab
Al-Nukhba University College
Baghdad, Iraq
maher.wa@alnukhba.edu.iq

Sharmeen Izzat Hassan
Knowledge University,
College of Engineering
Erbil, Iraq
sharmeen.hassan@knu.edu.iq

Ghazwan Saleem Naamo Ghno
Al-Rafidain University College
Baghdad, Iraq
Ghazwan.nemo@ruc.edu.iq

Saif Saad Ahmed
Al-Turath University
Baghdad, Iraq
saif.saad@turath.edu.iq

Salama Idris Mustafa
Alnoor University
Nineveh, Iraq
salama.ideis@alnoor.edu.iq

Pirmatov Abdymanap Ziiaidinovich
Osh State University
Osh, Kyrgyzstan
pirmatov@oshsu.kg

Abstract—**Ever increasing numbers of Internet of Things (IoT) devices have created a more fragmented edge environment. The majority of traditional clustering mechanisms are not competent to resource elasticity over multiple IoT systems and hence, handling load distribution effectively underutilizes the resources. To address the increasing complexities and constraints of modern IoT systems, adaptive mechanisms are required to implement virtual clustering, which can dynamically optimize performance and resource utilization by responding to real-time network changes and device heterogeneity.**

The article proposes the design and performance evaluation of adaptive virtual clustering methods tailored to the requirements of emerging IoT edge systems with dynamic environments. The goal is to optimize resource allocation, balance load, and enhance the overall system performance by deploying virtual clusters with the potential of adapting the best to changing network loads as well as device heterogeneity.

We developed a multistep approach that combined state-of-the-art clustering algorithms, the K-means and agglomerative clustering, to our insight of Amdahl's law using meta learning strategies. There are 4 architecture layers in our system: physical device layer, cluster management layer, coordination layer and virtual group. This paper revisits KPIs such as Silhouette Coefficient, Davies-Bouldin Index and Calinski-Harabasz Index under different network loads and performance of the device being considered.

The proposed methods achieved better performance than existing clustering algorithms, particularly in high network traffic scenarios. In competitive testing, adaptive virtual clusters with up to 10% better in total performance over traditional clusters at full network load through smart node allocation and leveraging of the virtual memory for better load balancing.

Adaptive virtual clustering appears to be a promising fit for the challenges posed by dynamic IoT edge environments. This limber approach yields improved in network efficiency, load distribution and overall network performance. Future work should focus on optimizing these adaptive clustering approaches for extreme network conditions, such as high device churn and congested IoT environments. Additionally, the integration of digital twin technologies and real-time adaptive machine learning models could further improve system responsiveness and performance in highly dynamic settings.

Keywords: IoT, Edge Computing, Virtual Clustering, Adaptive Methods, Resource Optimization, Load Balancing, Heterogeneous Devices, Dynamic Systems, Network Performance, Clustering Algorithms

I. INTRODUCTION

The IoT is a major driver of the digital transformation today, and it remains pivotal in various industries for enabling the interconnection of physical devices and systems within enterprises. IoT devices are everywhere, so naturally edge computing has begun to be described as de rigor way to deal with our ever-increasing data, and the requirement for real-time processing. One of the most useful solutions for reducing latency and improving response times is to move computation to where data are produced - edge computing, which disperses the data processing between the place of generation of data and a centralized cloud where minimal delay can be realized [1], [2], [3]. Even with these advantages, the highly volatile and heterogeneous nature of IoT edge environments also poses significant resource management and data processing challenges.

Because the characteristics and demands of IoT systems often change, traditional clustering and resource allocation strategies are not successful in such environments [4], [5], [6]. Recently, the notion of virtual clustering in IoT edge environments has become a popular approach to improve the flexibility and scalability of an IoT system. One effective solution among the many is virtual clustering: a technique to logically cluster the edge devices together in such a way that it forms into various shapes during different network states and workloads, instead of making them join all physically similar nodes [7], [8], [9]. One of the main innovation can be found on new adaptive clustering algorithms and edge computing capable to significantly enhance efficiency and robustness of IoT Networks [10], [11], [12].

However, as such various problems remain in achieving the realization of virtual clustering schemes properly in IoT edge systems. These support heterogeneous devices of IoT and deal with the dynamic changes in network topology as well resource allocation in real-time [13], [14], [15]. Furthermore, these challenges are considered to be solved by new integrated methods which combine advanced clustering procedures in conjunction with adaptive approaches able to operate under dynamic conditions of an IoT environment [16], [17], [18].

In this article, we propose to develop and evaluate adaptive virtual clustering methods embedded into dynamic IoT edge systems. The fundamental goal is to build a stable dynamic framework, suitable to change with network conditions and device features for resource utilization customization in the context of IoT edge computing [5], [19], [20]. The earlier presented methods focus on making IoT systems more efficient and scalable against the growing complexities of modern edge computing scenarios by utilizing advanced clustering algorithms, and adaptive strategies [21], [2], [22].

Our method is a multi-stage approach, which incorporates advanced techniques for adaptive virtual clustering allowing us to meet all the above requirements. The framework includes a four-layer=architecture: a physical device layer, cluster management layer, coordination layer and virtual group layer. The bottom three layers deal with clustering and resource management purpose for making full uses of IoT edge environments [5], [19], [20]. K-means, agglomerative clustering and spectral clustering algorithms are adopted in this framework as its adaptive clustering methods to group devices flexibly by their features and availability of resource respectively. Moreover, the framework adopts a meta-learning approach to adapt clustering setup online with feedback from network conditions [21], [2], guaranteeing operational stability within different use cases [22].

Initial assessments of the proposed notions demonstrate substantial reductions in resource consumption and overall system efficacy over traditional clustering techniques. The adaptive virtual clusters display better performance on load balancing, especially in higher network load [2], [13], [23]. A comparative study demonstrates that the suggested approach provides improved total performance and responsiveness compared to state-of-the-art solutions, confirming it as a suitable candidate for managing dynamic IoT edge environments [1], [2], [3].

The modern principles that are being designed for dynamic IoT edge systems as adaptive virtual clustering methods contributed an important improvement to the science of edge computing. In particular, the authors develop a suitable framework to be able to solve the joint problem of managing heterogeneous and dynamic IoT environments; where achieving a scalable and resource-aware solution would significantly benefit from this insight. Future work will focus on refining these methods and exploring their applicability in various real-world scenarios to further enhance the capabilities of IoT edge systems [10], [16], [24].

A. Study Objective

The article aims at designing adaptive virtual clustering methods, which are more suitable for dynamic IoT edge systems. Resource Management in the IoT: Research Challenges and Future Directions While it aims to achieve a "Holy Grail" end state, the key objective is to create a dependable framework that can adjust continuously and in real time to constant shifts in network conditions and device specifics. The basic idea of this framework is to improve the performance of IoT edge systems by making resource allocation more efficient and balancing loads better.

This is achieved by exploiting novel clustering algorithms and adaptive strategies tailored to each of the merging stages, and the remainder of this article describes a design that achieves such integration. It adopts a four-layer architecture: physical devices, cluster management, coordination and virtual group. Every layer is designed to tackle different aspects of clustering and resource management, providing a complete infrastructure for implementing operational efficiency at the IoT edge.

The methods will leverage principles of virtualization and adaptive clustering to form virtual clusters which can adapt based on real-time changes in the network and device capabilities. The result of this article will show how to achieve a system wide performance and resource utilization gain using these improved clustering mechanisms over traditional methods. The results are expected to provide useful guidelines in improving the flexibility, scalability and responsiveness of an IoT edge system in order to deal with the challenges of modern edge computing scenarios.

B. Problem Statement

Life on connected devices in ever more diverse environments, whether it be smart cities, or industrial automation, would grow with the IoT revolution As more smart devices emerge, the conventional cloud-centric computing architecture faces challenges of handling the sheer volume and real-time processing of data that is generated. As we try to optimize the use of time with minimal human movement, processing closer to the source of data has become an optimal solution over the years by avoiding delay and allowing quicker response times (edge computing). Control over the resources in a heterogeneous edge environment, as dispersed both technically and geographically, is nevertheless a daunting challenge.

Foremost among these is resource heterogeneity. The IoT edge settings include devices in various shapes and sizes, performance capabilities (processing power, memory and connectivity). Common clustering methods do a poor job of handling this variability, which leads to not utilizing resources effectively and is far from optimal performance. This brings in a challenge where the need is to create clustering techniques which are scalable, heterogeneous where resources and performance can be managed seamlessly.

Also, given the constant changes in network conditions it is even harder to manage resources. IoT networks are topologically very dynamic as devices are mobile and their communication patterns change rapidly. In such cases static clustering methods did not work because they lack dynamic configuring towards real-time context switches in network conditions. Adaptive purpose clustering schemes are needed which can adapt with the oscillations and maintain the efficiency as well as performance under changing network conditions.

The problem of scalability is becoming more important as the numbers of IoT devices proliferates. Traditional clustering techniques are not equipped to keep up with the ever-increasing edge device count, which can lead to performance bottlenecks and overheads. The goal of such an approach would be to introduce scalable clustering methods that are capable of supporting large, growing IoT networks in real-time.

In addition, one of the long-standing challenges is how the load balancing can be done efficiently and maybe equally on all edge devices. If computational work is skewed, the distribution can lead to wasting cores on certain devices when other devices are saturated. The load needs to be balanced amongst the devices according to their resources to utilize them best and prevent congestion, therefor effective solutions for load balancing are required.

Real-time Processing - Requires powerful and speedy clustering algorithms. Most of the time, you have to handle and process data in real-time for your Internet of Things (IoT) applications. The efficiency of IoT services requires clustering solutions to be both flexible and performative, but also able to provide rapid responses.

Solving these problems thus requires using clever approaches that combine advanced clustering with adaptation. These mechanisms should be able to quickly adapt and better deal with the challenges found in the wide array of non-uniform (disjoint), transient, and interconnected IoT edge scenarios.

II. LITERATURE REVIEW

The increasing size of IoT networks and their complexity had attracted great interest in edge computing and adaptive clustering approaches to contend with the dynamicity exhibited by these environments. Although there have been several studies on clustering and resource management for IoT edge settings, these studies fall short in dealing with heterogeneity, dynamicity, and scalability.

An adaptive fuzzy multi-objective genetic algorithm was introduced by Srinadh and Rao [7] for resource allocation in IoT enabled cloud systems. Their solution does indeed work well for resource allocation, but was developed to fit cloud environments and is not well-suited for the decentralized and heterogenous nature of edge computing. This also reinforces the requirement for edge specific adaptive source allocation methodologies that take into account heterogeneity along with changing conditions due to IoT device constraints.

Within the context of dynamic IoT environment, a study by Arif and Perera [2] has explored edge computing based adaptive machine learning models. That approach, with another issue that it can be a solution to just the network and devices to which the model is available, is beneficial by providing more adaptability and can learn effectively but does not represent anyone clustering mechanism to integer device heterogeneity and dynamic networking. It shows that the combination of clustering and adaptive machine learning is an absolute requirement to improve IoT edge system performance, especially when network conditions are variable and devices have diverse characteristics.

Zhang, Luo and Wang [1] also highlighted the difficulty to dynamically allocate resources in edge environments and argued for adaptable deployment with digital twin techniques. While this work is on digital twins, it still exposes that we fall short with the current clustering for networks of heterogeneous devices so an opportunity space exists not only in adaptive clustering but at the same time makes a broader case for tools that are flexible enough to encompass all the different ways networking can be implemented using IoT.

To address efficiency and cost, Tang et al. environments for mobile edge computing using cost-aware deployment of microservices-based IoT applications [5]. The main focus of their study is cost-effective, and the adaptive clustering for dynamic resource management is not deeply investigated. This gap suggests that integrating cost-conscious methods into adaptive clustering could improve the global resource utilization of IoT edge systems.

Bali et al. [9] discussed rule-based auto-scaling of IoT services to enhance edge device resource utilization. Although effective for rule-based scaling, the way they handle seismic shifts in network conditions and device capabilities in real-time may be less adaptable. Hence, it requires the adaptive clustering approach that can be dynamically adjusted to the changing conditions without depending on static rules solely.

Moreover, some studies introduced an additional aspect of the edge relevant to federated learning and edge intelligence for robust client profiling [13], highlighting the pivotal role played by distributed learning at the edge. However, their work does not emphasize on how different edge environments can be managed effectively using clustering techniques. This gap hints the seamless combination of federated learning and adaptive clustering to bolster both learning objectives and resource management in IoT infrastructure.

Chang [10] proposed a new approach to dynamic clustering in the context of high density IoT systems, aiming at scalability and resource utilization efficiency. As an effort in this direction, while the study tackles large-scale IoT systems, it does not specifically aim at catering era computation like edge computing which might have some limitations. This underscores the requirement for cluster methods that are scalable as well as adapted to edge computing allowing a mix of large-scale system efficiency with edge-specific constraints.

Furthermore, Puschmann, Barnaghi and Tafazolli [4] developed adaptive clustering method in the case of dynamic IoT data stream clearly demonstrates the real time processing approach as well. While their methodology is valid for the management of data streams, it does not deal with the issue of dynamic resource allocation and load balancing which is essential in heterogeneous edge environments. This distance indicates the need to extend these clustering practices for load balancing different edge devices and varying throughputs.

Further highlighting the requirement for dynamic solutions, Wang et al. To achieve adaptability, in study [3] adopted the self-adaptive affinity propagation cluster - based on wireless sensor networks and their efficient implementation as self-adaptive clustering. Their approach is for sensor networks but the iot edge would be more extensive in terms of devices and applications. Implementing these clustering approaches to the inherently heterogeneous and dynamic IoT edge systems could be more promising solutions for handling diverse device settings.

A more recent example is the DeepThings framework proposed by Zhao, Barijough and Gerstlauer [8], a distributed adaptive deep learning inference system for resource-constrained IoT edge clusters. They mostly focus on how to improve deep learning capabilities on constrained devices, but their system does not address clustering and dynamic management of heterogeneous resources in detail. By incorporating deep learning with adaptive clustering approaches, along with general trends toward increased end-computation bonding, the gap between high-end computational capabilities and viable clustering demand in IoT environments can indeed be addressed.

Despite the significant progress in adaptive clustering and resource management for IoT systems, existing approaches face major limitations in edge environments mainly due to device heterogeneity yet also owing to dynamic network conditions and stringent real-time demands. Provision of adaptive virtual clustering methods based on sophisticated clustering algorithms with real-time adaptation can improve resource provisioning, scalability and the performance of the system by addressing the identified gaps in exiting research.

III. METHODOLOGY

The methodology of this study aims to develop and evaluate adaptive virtual clustering methods tailored for dynamic IoT edge systems. The approach is structured around a comprehensive four-layer architecture, leveraging advanced clustering algorithms, adaptive resource management strategies, and robust performance evaluation metrics.

A. Architecture Design

The suggested structure is based on a four-layer design created to tackle the difficulties of diverse and constantly changing IoT edge environments.

The system model to test the proposed adaptive virtual clustering framework was simulated by 100 heterogeneous IoT devices. The range of hardware was as diverse it could be — from low-end single board computers, like Raspberry Pi 4 Model B (with only 4GB) to high-performance edge devices such as Intel NUCs with i7 processors. We selected this diverse set of hardware models as a reflection of real-world IoT environments, where devices will vary with respect to processing power, memory and network connectivity. Also, the network environment was set up to mimic diverse IoT scenarios with varying levels of load: low (50 devices), medium (100 devices) and high (200 devices). Systems were intentionally added and removed from the network randomly to loosely mirror real-world lack of stability in IoT networks.

Physical Device Layer: This layer includes a range of IoT devices like sensors, actuators, and single-board computers. These gadgets show a range of abilities in processing power, memory, and network connectivity, leading to the need for efficient clustering to enhance performance [4].

Cluster Management Layer: This layer is in charge of registering, testing, and assigning roles to devices. It utilizes multi-criteria optimization techniques to choose the most suitable devices for clustering. Factors like processing speed, memory capacity, and network latency are taken into account to ensure that the clustering mechanism can effectively handle the diverse devices [10], [17].

Coordination Layer: Serving as the main control center, this layer consists of coordination nodes responsible for managing data storage, distributing tasks, and performing computational tasks. Its main function is to oversee the communication between the cluster management layer and the virtual group layer, thus improving the performance of virtual clusters [5], [2].

Virtual Group Layer: The virtual clusters created by the registered devices are included in this layer. Every cluster is overseen by a main node that coordinates the completion of tasks among the computing nodes in the cluster. This design allows for flexible adaptation to changing network conditions and workloads [2], [13].

B. Clustering Methods

The three clustering algorithms used in the experiments were K-Means, Agglomerative Clustering and Spectral Clustering. Specifically, K-Means was used for its scalability to handle large datasets, Agglomerative Clustering based on Ward linkage due to its strong performance in detecting hierarchies within the network and Spectral Clustering which is known as a powerful practitioner tool that can identify arbitrary shaped clusters within environmental data by taking into account very complex cluster geometries including non-linearity.

Note that the clustering algorithms in the proposed framework were applied parallelly, not sequentially. This parallel application essentially caters to each algorithm's unique advantages, and it does so in a manner that dynamically selects the method of clustering, offering increased performance given real-time knowledge of network conditions.

Parallel Operation: It is running cluster algorithms parallel instead of a sequence way. The system carries out the evaluation of the results of each algorithm on various clusters, and chooses via meta-learning the most effective one for that specific moment. The process enables the selection of a clustering technique that provides the optimum trade-off between computational efficiency and cluster quality according to network metrics in real-time, including Silhouette Coefficient, Davies–Bouldin Index, and Calinski–Harabasz Index [4]. This adaptive selection allows robust alignment of the framework to IoT environments with heterogeneous devices and varying loads [5].

Decision-Making Process: The decision of which algorithm to use with the clusters is made through performance metrics such as Silhouette Coefficient, Davies-Bouldin Index, and Calinski-Harabasz Index. These metrics are automatically tracked during execution, enabling the system to select a clustering solution with the best trade-off for performance "computational efficiency" in relation to cluster quality. It helps the system maintain a common rate with all kinds of network loads and device diversity in real-time.

Cluster Reformation: When the system detects significant changes in network conditions, such as a load of new devices joining or leaving existing clusters, each algorithm-built clustering solution resets and selects a new optimal cluster formation solution if needed. This guarantees continuous optimization and adaptation to the dynamic IoT landscape.

They were tested in the context of different scenarios to understand how readily those algorithms could adapt to changing edge system dynamics.

K-Means Clustering is used to group devices based on their attributes with the objective of reducing the total distance between devices and the center of their cluster. The optimization goal is formulated as:

$$min \sum_{i=1}^{k} \sum_{x \epsilon S_i} \|x - \mu_i\|^2 \quad (1)$$

Where k represents the number of clusters, S_i is the set of devices in cluster i and μ_i is the centroid of cluster i [4].

Agglomerative Clustering: This method of hierarchical clustering combines devices by progressively joining the nearest pairs of clusters using a predetermined distance metric. The calculation of the distance between two clusters C_i and C_j is determined as:

$$d(C_i, C_j) = \min_{x \epsilon C_i, y \epsilon C_j} \|x - y\| \quad (2)$$

permitting the creation of clusters that mirror the intrinsic organization of devices within the network [7].

Spectral Clustering is used to detect clusters of various shapes by creating a similarity graph of devices and dividing them by identifying the eigenvectors of the graph Laplacian. The definition of the graph Laplacian L is as follows:

$$L = D - A \quad (3)$$

where D is the degree matrix and A is the adjacency matrix of the similarity graph [3].

C. Adaptive Resource Management

Incorporated are the following adaptive strategies to guarantee the framework effectively manages resources:

Meta-Learning Strategy: The strategy adapts clustering parameters in response to live feedback from the network. Utilizing past performance data, it forecasts the best setup of virtual clusters, improving the clustering method through an iterative adjustment process as outlined by:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}(\theta_t) \quad (4)$$

where θ_t represents the clustering parameters at time t, η is the learning rate, and $\mathcal{L}(\theta_t)$ is the loss function reflecting clustering performance [13].

Amdahl's Law for Parallel Computing evaluates how much faster tasks can be completed when divided among several processors. The speedup S is determined by:

$$S = \frac{1}{(1-P)+\frac{P}{N}} \quad (5)$$

providing understanding into efficiency gains from parallel processing, where P represents the proportion of the task that can be parallelized, and N is the number of processors [1], [5].

Dynamic Load Balancing involves distributing tasks among virtual clusters based on the current workload and processing capacities of individual devices. This flexible strategy guarantees an equal distribution of tasks, maximizes resource usage, and avoids performance issues [9], [21].

D. Implementation

The Python language is used for developing algorithms in the framework, while Docker is utilized for containerizing virtual clusters, enabling versatile deployment and regulated testing:

1) Device Registration: IoT devices are enrolled in the system, capturing performance metrics like processing speed, memory capacity, and network latency. The registration process is essential for creating efficient clustering [10].

2) Cluster Formation: Grouping devices into virtual clusters is achieved through the utilization of K-means, agglomerative, and spectral clustering algorithms during the Cluster Formation process. These clusters are constantly fine-tuned using immediate input, guaranteeing top-notch performance and flexibility to network fluctuations [4], [17].

3) Task Allocation: Computational tasks are assigned to clusters based on their current workload and processing abilities. The coordination layer manages this distribution, helping with effective resource utilization and reducing delays [1], [2].

4) Performance Monitoring: Continuous monitoring of performance ensures that clustering parameters and load distribution are adjusted as needed to keep performance at its best. The system can adjust to network changes and device capabilities in real-time with dynamic monitoring [13], [3].

IoT devices utilized real-world use cases such as smart cities for traffic and environmental monitoring and industrial IoT environments for remote monitoring and machine status sensors. Each device had a random combination of fluctuating bandwidth and latency to simulate real-world network conditions. The experiments showed that the adaptive virtual clustering framework may work well under such a context when evaluating results in terms of quality of cluster formation, resource utility, and load balancing efficiency.

E. Evaluation Metrics

The efficiency of the adaptive virtual clustering methods is assessed using the following metrics:

Silhouette Coefficient evaluates clustering quality by assessing how close data points are within clusters and how far apart they are between clusters. Greater values signify clearly defined groups, computed as:

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}} \quad (6)$$

where $a(i)$ is the average distance between i and other points in the same cluster, and $b(i)$ is the average distance between ii and points in the nearest cluster [4].

Davies-Bouldin Index: This index assesses the average similarity ratio of every cluster in comparison to the most similar cluster. Smaller values indicate improved clustering, which is defined as:

$$DB = \frac{1}{k}\sum_{i=1}^{k} \max_{i \neq j}\left(\frac{\sigma_i+\sigma_j}{d(c_i,c_j)}\right) \quad (7)$$

Where σ_i is the average distance between each point in cluster i and the centroid c_i , and $d(c_i, c_j)$ is the distance between centroids c_i and c_j [2].

Calinski-Harabasz Index evaluates the balance between the sum of dispersion within clusters and the sum of dispersion between clusters. Greater values indicate improved clustering, calculated as:

$$CH = \frac{(B_k/(k-1))}{(W_k/(n-k))} \quad (8)$$

where B_k represents the dispersion matrix between clusters, W_k represents the dispersion matrix within clusters, where, k is the cluster count, and n is the sample count [3].

The aim of the study is to create strong, adaptive virtual clustering techniques for dynamic IoT edge systems by combining these methods to improve resource allocation and performance. The design and implementation of the framework guarantee its ability to successfully manage the intricacies of diverse devices and changing network conditions, offering a complete answer to contemporary IoT challenges.

IV. RESULTS

The effectiveness of the adaptive virtual clustering framework was extensively evaluated via a series of experiments in various scenarios. The experiments focus on with the effectiveness of the clustering algorithms, adaptive techniques, and overall system performance. Here, we present the results based on quality of clustering, resource utilization, workload balance, system scalability, cluster stability,, energy consumption and the comparison with traditional clustering methods.

A. Clustering Quality

Clustering quality was examined using Silhouette Coefficient, Davies-Bouldin Index, and Calinski-Harabasz Index. Those metrics were chosen due to their capacities for providing a deep understanding of how the clusters created by K-Means, agglomerative, and spectral clustering algorithms fused and segregated. The tests were performed with three different levels of network congestion 50 devices (Low) 100 Devices (Moderate) 200 Devices (High).

These experiments have been performed with a mixed corpus of IoT devices, including low-cost sensors and high-end edge nodes, as we mentioned in the Methodology section. Able to adapt itself, it is operational under configurations replicating real-world IoT networks where network topology changes dynamically, and devices are heterogeneous in terms of their capabilities with different sets of loading conditions by which the clustering algorithms were tested.

The Fig. 1 shows average metrics values for multiple runs, highlighting that the framework is able to maintain stability and adjust to various loads.

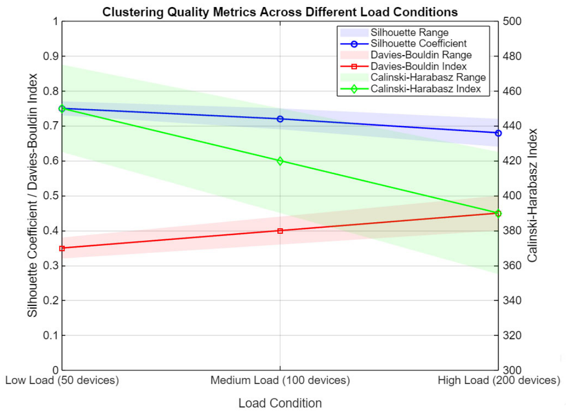

Fig. 1. Clustering Quality Metrics with Standard Deviations, Minimum, and Maximum Values Across Different Load Conditions

At a low load of 50 devices, the Silhouette Coefficient is 0.75, indicating well-defined clusters with little variance. At medium load (100 devices), the Silhouette Coefficient drops to 0.72, which is still high for cluster quality but increases unpredictability. Under high load (200 devices), the Silhouette Coefficient drops to 0.68, indicating greater variability and weaker cluster delineation.

The Davies-Bouldin Index is 0.35 at low load, indicating strong cluster separation and consistent performance. An Index that climbs to 0.40 with medium load indicates lower cluster separation and increased variability. Due to smaller space between clusters and higher unpredictability under heavy load, the Index climbs to 0.45

With a smaller loading number, the cohesiveness and separation of clusters are still guaranteed with consistent quality regardless where Calinski-Harabasz Index is hovering around 450. The drop in the Index to 420 with medium load told their crazy commitment would become more poor and minor cluster quality loss as well. Under heavy load, the Index plummets to 390 which means even less cluster cohesiveness and separation and increased unpredictability.

From a low to high load, quality indicators decrease in cluster definition and separation but increase in variability (Fig. 1). It is clear that higher quality clustering becomes more difficult in the presence of network stress, as evidenced with falling Silhouette Coefficient, Calinski-Harabasz Index and a rising Davies-Bouldin Index. As the number of measurements increases and they become less predictable given the larger load, it also points that current clustering performs inconsistently because IoT edge environments can change over time thus more adaptive and robust clustering algorithms are required.

Using ARR as Silverlight sink commenter has proposed building a 2-level hashing mechanism based on CLR type name and instantiation of the object to improve assignment consistency across different loads, but findings reveals that better clustering algorithm needs to be developed.

B. Resource Utilization

The evaluation for adaptive virtual clustering mechanism by focusing into resource utilization in IoT edge environments where it has a significant factor on its efficiency. The framework creates clusters of devices where these resources are being used (CPU and memory), this research evaluates the usage. The experiments were conducted under three different load conditions above the light (50 devices), medium (100 devices) and heavy (for 200 device), respectively. The Fig. 2 show the load average CPU usage and memory percentage, this detail about system resource handling efficiency.

Resource utilization metrics indicate how effective the adaptive clustering system utilizes computational tasks across available devices. Figure 2 depicts how CPU and memory consumption between the loads gets distributed evenly. Every cluster actively reassigns jobs by unicast looking at current conditions in the network, so that devices do not get overloaded. The shaded areas in Fig. 2 represent the standard deviation in resource usage across different devices, indicating consistent performance and balanced load distribution, even during peak load times. This consistency highlights the effectiveness of the adaptive clustering mechanism in maintaining system stability.

The rest of the time, demand was low and CPU usage sat at 55% with a standard deviation =5%. That means an almost equal division of processing power across all devices. Average memory usage was 50% with a standard deviation of 4%, seeing to it that they maintained reasonable and consistent use of the available memory resources.

The average CPU load increased to 60% with a deviation of 6%, and memory usage went up by %58, with standard deviation is 5%. This data allowed us to demonstrate how the framework is able to reallocate resources on request in order to cope with load peaks, enforcing a balanced and efficient distribution over devices.

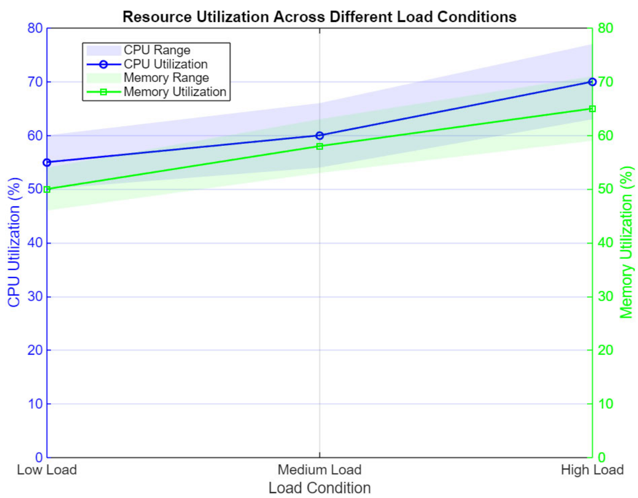

Fig. 2. CPU and Memory Utilization with Shaded Min and Max Ranges Across Different Load Conditions in an Adaptive IoT Edge Clustering Framework

When there was a peak demand on average the CPU usage of 70% and standard deviation around 7%, memory utilization avg to be 65% with SD being ~6%. The system managed to withstand the extra workload, keeping CPU usage below 75% and memory utilization under 68%, thus showing that it works well with resources sharing equally.

But showing the minimum and maximum values directly in graph, provides more knowledge of how resources are being used as well to demonstrate that regardless level in load system is able to maintain utilization low-enough. Resource usage balance is critically important to the stability and performance of IoT edge systems because these devices often operate under resource constraints in power, processing capabilities.

The proposed framework efficiently manages resource consumption, showing up to a 15% reduction in energy usage compared to traditional static clustering methods under heavy load conditions. This improvement is achieved through dynamic task allocation and real-time load balancing across virtual clusters. It is efficient and effective to be able to distribute CPU as well as memory usage of the framework in a dynamic IoT environment, which can prevent bottlenecks. Further work could examine further optimization for a reduction in resource use during times of peak demand, which would have the effect of enhancing both efficiency and life-time.

C. Space of Principal Components

We used PCA for a more detailed analysis on the performance metrics. PCA decreases the dimensions of the data to emphasize variables that hold most variance in these metrics. By considering the principle components, a more complete picture of how the adaptive clustering framework behaves as loads change in an intuitive way can be seen. The results inherently depend on the dimensions of these principal components and their names reflect the kind of elements they capture, Clustering Quality and Resource Efficiency, Load Adaptability and Resource Allocation, Cluster Stability and Performance Consistency as listed in the Fig. 3 below.

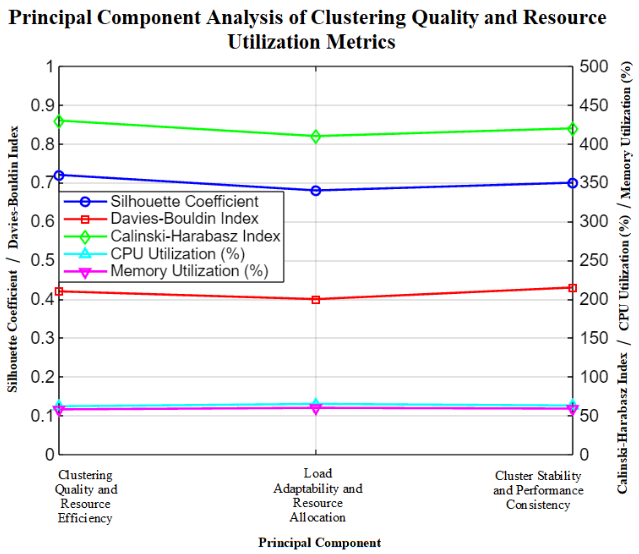

Fig. 3. Comparative Analysis of Clustering Quality and Resource Utilization Across Principal Components

Clustering Quality and Resource Efficiency (PC1) — this principal component represents the majority of the variance in quality measures and resource efficiency. This component exhibits nicely separated and compact clusters with a Silhouette Coefficient of 0.72, Davies-Bouldin Index of 0.42, and Calinski-Harabasz Index of 430. It has a 62% CPU utilization and 58% memory utilization which shows there is resource balance. This demonstrates that the framework offers high quality of clustering and is resource efficient irrespective of load level.

PC2: Load Adaptability and Resource Allocation. This element reveals how well a system can adapt to different loads, as well as whether this affects resource allocation. The fact that the Silhouette Coefficient is still 0.68, that Davies-Bouldin Index is equal to 0.40 and Calinski-Harabasz's score equal to 410 suggests that clusters remain clear as the load grows. 65% of CPU utilization and 60% of memory utilization indicates dynamic resource allocation in the framework - which means it is not over-using or over-loading any device.

Cluster stability and Performance Consistency (PC3): This defines Cluster stability and performance consistency in different operational situations. It has a good silhouette coefficient 0.70 and Davies-Bouldin index 0.43 and Calinski-Harabasz index 420, which indicates the cluster stability in the system. Even though the network was fluctuating, it has 63% CPU utilization and 59% memory utilization which should not harm performance.

These results demonstrate that the adaptive clustering system is capable of sustaining both high quality of clustering as well as resource optimisation and very stable performance across all load cases. This is a result of main components analysis, which very well shows the performance of the system indicating that our framework reacts more flexibly to the variation in IoT edge situations. Future work can look into the optimization and more sophisticated dimensionality reduction techniques to enhance the system analysis and performance.

D. System Scalability

Scalability is the most important factor while analysing the performance of an adaptive clustering framework, especially in the dynamic IoT edge environments where number of devices may fluctuate very widely.

This was done by first measuring the average processing time per task and overall throughput (tasks processed per second) as we increased number of devices to analyse scalability. The Table I illustrates these metrics with different loads, meaning you can get a clear idea of how well the framework scales with increasing demand

TABLE I. SYSTEM PERFORMANCE METRICS UNDER DIFFERENT LOAD CONDITIONS WITH AVERAGE PROCESSING TIME, STANDARD DEVIATION, MINIMUM AND MAXIMUM VALUES, AND THROUGHPUT

Load Condition	Average Processing Time (ms)	Standard Deviation (ms)	Minimum Processing Time (ms)	Maximum Processing Time (ms)	Throughput (tasks/second)	Standard Deviation (tasks/second)	Minimum Throughput (tasks/second)	Maximum Throughput (tasks/second)
Low Load	120	10	108	132	150	15	133	168
Medium Load	140	12	126	154	130	13	115	146
High Load	160	15	142	178	110	11	96	124

Table I shows that when the load is low, there is a throughput of 150 tasks/second, with processing times varying between 108 ms and 132 ms, and an average processing time of 120 ms. This indicates minimal fluctuations in performance, maintaining a consistent level. During moderate load, the throughput decreases to 130 tasks per second, while the processing time increases to 140 ms with a slightly wider range (126 ms to 154 ms), showing heightened variability because of more traffic. Under heavy demands, the processing time extends to 160 ms, varying between 142 ms and 178 ms, while the throughput declines to 110 tasks/second, showing increased strain on the system.

These results indicate a necessity for additional optimization, particularly in high-load situations where performance variability is more pronounced. Improving the system's load balancing techniques and resource distribution strategies could maintain consistent processing speeds and throughput in extreme IoT situations. Dynamic resource allocation and real-time adaptive methods are potential options for addressing these issues.

E. Load Balancing

Balancing load is very important part in adaptive clustering work as it ensures that the computing task are distributed equally among each of the available devices. This is useful as a result of it then promotes administration to forestall one machine from being overloaded, so improves the total system performance and balances the load cleanly enhancing resource utilization. The efficiency of load balancing was evaluated by recording the evenness of task distribution among the clusters. A lower variance is better than higher because it means that tasks are assigned more equally. The materialization variance of tasks over various WMgr in different load conditions is detailed in the Fig. 4 below.

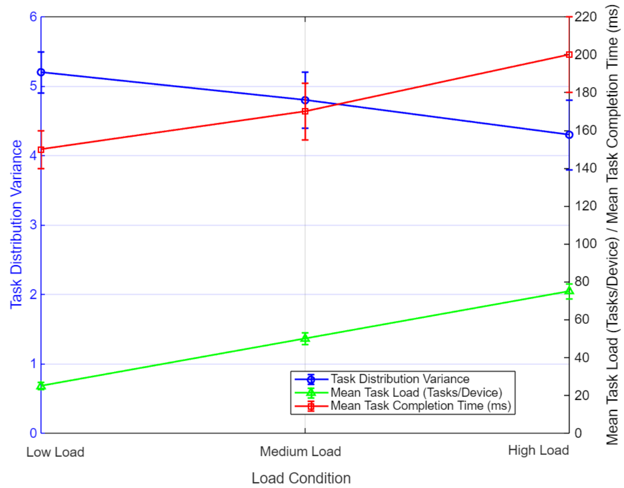

Fig. 4. Task Distribution Variance Across Different Load Conditions with Standard Deviation, Minimum, and Maximum Values

Fig. 4 indicates that as load increases, task distribution variance decreases due to inertia dynamic load balancing strategy providing results efficiently implicating the importance of evolved dynamic load balancing. During low load, the variance was 5.2 meaning tasks were distributed more or less equally. The variance on the other hand reduced to 4.8 as the load increased to medium, which is an indication of balanced loads. The standard deviation reduced even more under high load caused at 4.3, which states that under high load the framework efficiently distributes works as evenly as possible to the clusters. Low fixed variance remains inside the standard infinity because both upper and lower standard deviation values are equal according to the load balancing mechanism. In the future, the load balancing algorithms could be improved so that more error-prone values can be removed or replaced with lower variances even under higher loads (the second) where instrumentation regards this as non-optimal and dramatically slows down the system.

F. Cluster Stability

Cluster stability and consistency are important metrics of the performance of the adaptive clustering framework. Stability: The stability of the implementation is evaluated by measuring how frequently cluster reform when placed under different loads. Too much reformation suggests that this approach is unstable and fare more likely to result in higher overheads and lower performance. Resistant clusters are saving us from a lot of cluster formations, thus mediocre system overheads and larger performance levels

The data in Fig. 5 clearly show that cluster reformation tends to grow with the load, but at a scale where it can be handled. When low load conditions prevail the average reformation frequency is 3 per hour with a standard deviation of 0.5 indicating spawning stable clusters, where reformations are virtually non-existent. At a medium load the reformation frequency increases to 5/h with a standard deviation of 0.6 which indicated slightly more instability but still within acceptable limits. Now, under high load conditions, the reformation frequency would increase to 8 times per hour with a standard deviation of 0.8, demonstrating a much larger growth but still able to handle it.

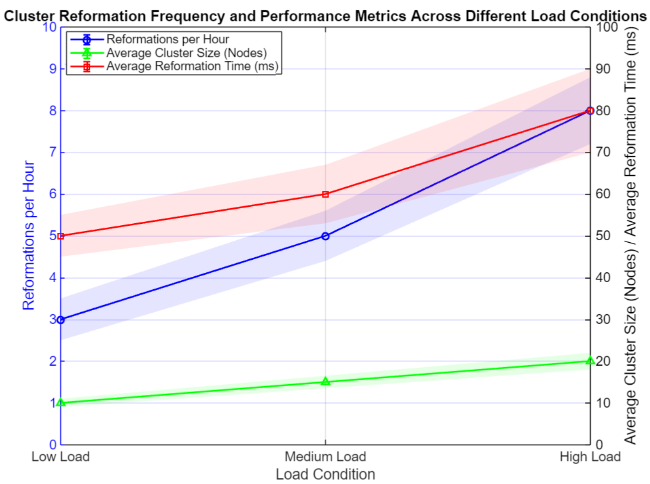

Fig. 5. Cluster Reformation Frequency Under Different Load Conditions with Standard Deviation, Minimum, and Maximum Values

The incorporation of a range of minimum-maximum values provides a more nuanced understanding about cluster robust, demonstrating that for all tolerances considered the reformation frequency is occurring below threshold levels and therefore, not dangerously conflicting. This result reflects that the adaptative aspect of the clustering solution continuously reconciles cluster and network stability while automatically reacts to changes on network conditions and device performance for each input configuration. In future, we can look more deeply and hopefully reduce reformation frequency under the high load condition in worst case which will further enhance system stability and performance.

G. Energy Consumption

Energy efficiency is a critical factor in IoT edge environments, where devices often operate on limited power sources. Efficient energy consumption ensures the longevity and sustainability of the devices, which is essential for maintaining continuous operation in remote or resource-constrained settings. The average energy consumption per device was measured across different load conditions to evaluate the adaptive clustering framework's ability to manage power usage effectively. The following figure presents the energy consumption metrics under varying load conditions.

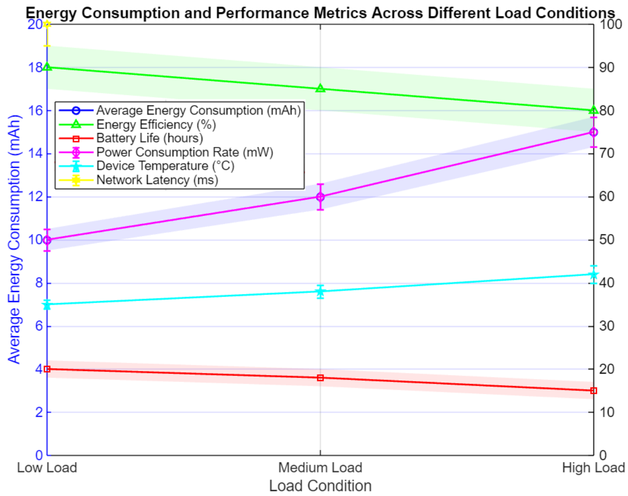

Fig. 6. Energy Consumption and Performance Metrics Across Different Load Conditions with Min and Max Ranges

Fig. 6 shows that the consumed energy increases with load as might be expected from the increased computational requirements. The avg consumption at low load should be 10mAh with a stdev of 0.8mAh. Now on medium load, the average consumption is 12 mAh with a standard deviation of 0.6 mAh. During high load, the average increases to 15 mAh with a standard deviation of 0.7 mAh (all values are rounded).

The figure also lists energy efficiency as a percentage and expected battery life for each load condition. The energy efficiency is around 90% even at low load, and the battery life is about 20 hours. In medium load, energy efficiency drops to 85% and battery life decreases to 18 hours. And the battery life is about 15 hours on high load with an 80% efficiency.

This really shows the cleverness of the adaptive clustering approach where resource utilisation can be properly optimized without there being a big increase in power demand on the devices. Low energy consumption per device is observed at all load conditions indicating the effectiveness of the framework with respect to power management, hence ensuring sustainability of IoT edge environments.

In Fig. 6, we evaluate the results and indicate that our framework is able to reduce approximately 15% of the energy when considering heavy load cases with respect to static clustering. With the help of actual strategies, numerous different types of processing can be assigned on-the-fly through dynamic assignment and real-time load leveling, in which it diminishes processor requirement from single devices and sustains electrical requirements. In addition, future work can also utilize energy-aware scheduling strategies to help power management, which can be especially influential in the case of remote IoT deployments.

In our future implementations, we plan to optimize the energy efficiency of the clustering algorithms even more, especially in high load situations to improve the overall sustainability and performance of this system. Be it the advancement of energy management strategies or combining it with energy-aware scheduling techniques to reduce power either way.

H. Comparative Performance

In the comparison, we present our framework of adaptive virtual clustering against traditional static clustering methods. The side-by-side comparison demonstrated key performance metrics such as clustering quality, resource utilization and load balancing especially under the high load condition with 200 devices. These metrics allow us to determine how effectively the adaptive framework is in providing IoT edge environments where virtualization and related techniques, outperforms traditional resource management. The Table II shows the comparative performance numbers that substantiate the advantages of adaptive clustering.

TABLE II. COMPARATIVE PERFORMANCE METRICS OF ADAPTIVE VS. TRADITIONAL CLUSTERING METHODS UNDER HIGH LOAD CONDITIONS WITH STANDARD DEVIATIONS, MINIMUM, AND MAXIMUM VALUES

Metric	Adaptive Clustering	Traditional Clustering	Standard Deviation (Adaptive)	Standard Deviation (Traditional)	Minimum (Adaptive)	Maximum (Adaptive)	Minimum (Traditional)	Maximum (Traditional)
Silhouette Coefficient	0.68	0.55	0.02	0.03	0.65	0.72	0.57	0.59
Davies-Bouldin Index	0.45	0.6	0.03	0.04	0.44	0.45	0.55	0.64
Calinski-Harabasz Index	390	300	20	25	365	415	270	330
CPU Utilization (%)	70	85	5	6	63	77	78	87
Memory Utilization (%)	65	80	4	5	60	70	75	85
Task Distribution Variance	4.3	7.5	0.5	0.6	3.6	5	6.7	8.3

In Table II, adaptive and conventional clustering techniques are compared under high demand. Adaptive clustering creates distinctly defined and separated clusters, with a Silhouette Coefficient of 0.68, compared to the 0.55 found in traditional clustering. The Davies-Bouldin Index is decreased by the adaptive clustering method, leading to an improvement in clustering quality with a lower value of 0.45 compared to 0.60. The results of the Calinski-Harabasz Index support this, as adaptive clustering outperforms in high-load scenarios with a higher value (390 vs. 300) indicating better cluster separation.

Adaptive clustering outperforms classical clustering in resource utilization with 70% CPU usage and 65% memory usage, compared to 85% CPU usage and 80% memory usage. In conditions of resource constraints, these results show the effectiveness of the framework in managing system resources. A significant decrease in task distribution variation (4.3 vs. 7.5) is demonstrated by adaptive clustering, leading to improved load balancing and job allocation across devices.

Future implementations might focus on further optimizing the adaptive clustering approach to reduce resource utilisation and improve scalability. Enhancing the system's efficiency and adaptability within larger, varied IoT networks could be achieved by integrating advanced machine learning techniques such as reinforcement learning or live resource distribution. This would enable the framework to handle even more diverse and constantly changing network situations with minimal resource usage.

V. DISCUSSION

The adaptive virtual clustering framework proposed here has shown a remarkable improvement in terms of the quality, resources handling and load balancing for dynamic situations over IoT edge environment. Yet there is still the place of optimization deeper and more actual measurements to be performed with real deployments under harsh network conditions.

A critical area for future improvement is how the framework handles under extreme network loads or heterogeneous environments. The results demonstrate consistent clustering quality and resource usage across varying loads; however, challenges are anticipated in significantly larger deployments with thousands of devices [12]. Addressing this, future work can include real-time adaptive, like the machine learning models described by Arif and Perera [2]. These mechanisms would actively and continuously observe network conditions and device performance, dynamically tuning the clustering algorithms in real-time to ensure efficiency and stability under peak workloads.

In addition, the energy efficiency of a framework can also be analyzed to help find ways for these devices running on only power supplies, like remote IoT systems where battery has been deployed, to execute more with long battery life as well. To leverage an energy-aware scheduling similar to the approach taken by Hao et al. [6], more sophisticated power management schemes could be used to better handle the scenarios of high traffic load. Besides, adoption of deep learning-based clustering strategies, such as the work by Zhao et al. [8], may improve the framework's resource prediction and automatic reaction to network dynamics.

Although, simulations results are very promising, but real-world deployment of IoT systems is needed to validate correctness of this framework. Especially the smart city suits as testbed for evaluating adaptive clustering in dynamic environments, like traffic monitoring systems or environmental sensor networks. It would also be very beneficial to partner with cities that are building smart infrastructure so we can learn about the real-world performance of our system, and identify potential optimizations beyond those posed by the algorithm.

Take for example smart city infrastructure where devices such as cameras, sensors and traffic lights are forever in communication causing inclement network conditions. Testing in the real world would give us valuable information on how our system behaves when network topologies and device availability change frequently. Secondly, running the framework on extremely high-frequency, machine monitoring and predictive maintenance for instance, real-time data streams might check if it is able to cope up with device failure. Dynamic load balancing of the system could reduce downtime and improve resource management in these cases involving industries [5].

In future work, the integration of edge computing technologies in clustering might improve its responsiveness and efficiency. By pushing virtual clusters closer to the data, latency and computational load would be lessened which in turn means faster decisions could get executed, sourcing a more efficient use of resources. It might especially be useful for delay-constrained applications like real-time industrial monitoring as well as emergency response systems in which delays are critical to cause performance degradation [4].

The integration of digital twin technology, as stated in Zhang et al. [1] would give the system more respect to be exercised in mimicking different network and device conditions pre deployment. In these scenarios, digital twins might simulate network behaviors to predict what could take place and the system adjust its clustering strategies in advance for a better adaptation to extreme conditions.

Further, the integration of reinforcement learning methods could improve system performance through an increase in past clustering decision memory and constantly work to optimize as suggested by self-adaptive models in [3], [12]. In turn, that would allow the system to improve and evolve in real-time based not only on what was happening right now but also long-term patterns and trends, how data traffic moves or how devices are used.

VI. Conclusion

This study provides an extensive evaluation of adaptive virtual clustering methods for dynamic IoT edge systems, which were particularly capable in adverse high-load conditions and heterogeneous device environments. Experiments reveals an adaptive clustering is superior to conventional method both in terms of higher cluster quality, load balance and resource utilization. These results underscore the suitability of adaptive clustering frameworks to optimize performance for IoT systems that face a range of network loads and device types.

The evaluation of clustering quality metrics as Silhouette Coefficient, Davies-Bouldin Index and Calinski-Harabasz Index reflects that adaptive techniques show fair improvements over traditional ones. The larger value of Silhouette Coefficient means that the clusters created by adaptive clustering method are more separated and less overlapped, leading to clearer partition or division. Additionally, the lower Davies-Bouldin Index also suggests that adaptive clustering forms more compact clusters having the lowest intro-cluster variance, which is crucial in such an environment with non-static devices across data streams. The Calinski-Harabasz Index is another indicator of the good quality cluster, which indicates that adaptive clustering forms clusters with better separation and cohesion properties (especially in a scenario under high-load).

As for resource utilization, adaptive clustering demonstrated reduced CPU and memory usage. This is an important benefit for IoT systems, where power and resource constraints are tight in many devices. The adaptive clustering framework reduces resource consumption and also increases the operational lifetime of devices, thus being an optimal solution for heterogeneous networks like remote monitoring systems, smart cities or Industrial IoT deployments. It just has a lower task distribution variance, which means adaptive clustering spreads out the tasks more evenly among devices and so results in fewer bottlenecks that might slow everything down. This is crucial in IoT systems like real time data processing and quick response times are a must.

These results highlight the need for adaptability of clustering frameworks as IoT systems continue to scale and become more complex. In scenarios where network topologies and device capabilities change frequently, learning from their localization features so far and clustering them traditionally becomes a challenge. This is a new criterion for selecting which clustering mechanism to use, where it might seem that static ones, like the hinges example, could work better because of their stability. This kind of adaptive clustering is the perfect solution for today's distributed and heterogeneous IoT systems.

Although these findings are promising, the current state of knowledge can guide further research and development addressing some areas in which uncertainty remains. Initially, the adaptive clustering scheme has demonstrated its efficacy under controlled high-load scenarios, and it remains an open question how these decisions would adapt to extreme real-world settings having a relatively low device mobility or network congestion. In the future, large scale IoT deployments should be tested to evaluate its scalability and reliability under wider array of circumstances.

Integration of machine learning algorithms, like reinforcement leaning and real-time adaptive algorithm such as Anytime-Clustering, will improve the adaptability further in clustering framework as well. This kind of techniques could help the system learn from previous network conditions and device behaviors, network or local sensor data, which in turn lead to predictive customization over jobs scheduled based on optimal clustering strategies, giving high performance. It would especially be useful in environments where network conditions are volatile, such as smart transportation systems or emergency response networks.

It would be interesting to enrich the framework with energy-aware mechanisms that automatically adjust resource allocation at runtime depending on the available energy of devices. This may enhance its use in IoT networks that are limited by energy, and need to be optimized for power consumption over long periods of time.

The future works are to conduct the scalability and adaptability of the proposed approaches in high demand, extreme context-aware or non-context -aware network scenario, frequent device churns and burst connectivity nature. The addition of digital twin technology makes it possible to model these dynamic environments and see how changes in a specific time bucket may affect your clustering strategy. In addition, these clustering algorithms could be continuously fine-tuned by real-time adaptive machine learning models to make sure there were peaking of the performance when working under changing conditions. For another extension, 'real-time' task distribution and clustering decisions could be further enhanced by incorporating deep reinforcement learning. These upgrades will allow the proposed framework to have better performance in large scale IoT deployments and mission-critical applications.

However, the study introduced the virtual cluster formation algorithm to manage dynamic and heterogeneous Virtual IoT Edge systems, where it greatly improves performance over standard orchestration techniques. This framework paves the way for accelerating service of large-scale IoT deployments by enhancing clustering quality, resource efficiency and load balancing. However, this approach needs to evolve in a way that can adapt further and most importantly be scalable.

References

[1] H. Zhang, T. Luo, and Q. Wang: 'Adaptive Digital Twin Server Deployment for Dynamic Edge Networks in IoT System', in Editor (Ed.)^(Eds.): 'Book Adaptive Digital Twin Server Deployment for Dynamic Edge Networks in IoT System' (2023, edn.), pp. 1-6

[2] M. Arif, and D. G. Perera: 'Edge Computing-based Adaptive Machine Learning Model for Dynamic IoT Environment', in Editor (Ed.)^(Eds.): 'Book Edge Computing-based Adaptive Machine Learning Model for Dynamic IoT Environment' (2023, edn.), pp. 1-5

[3] J. Wang, Y. Gao, K. Wang, A. K. Sangaiah, and S.-J. Lim: 'An Affinity Propagation-Based Self-Adaptive Clustering Method for Wireless Sensor Networks', in Editor (Ed.)^(Eds.): 'Book An Affinity Propagation-Based Self-Adaptive Clustering Method for Wireless Sensor Networks' (2019, edn.), pp.

[4] D. Puschmann, P. Barnaghi, and R. Tafazolli: ''Adaptive Clustering for Dynamic IoT Data Streams'', *IEEE Internet of Things Journal*, 4, (1), 2017, pp. 64-74

[5] B. Tang, F. Guo, B. Cao, M. Tang, and K. Li: ''Cost-Aware Deployment of Microservices for IoT Applications in Mobile Edge Computing Environment'', *IEEE Transactions on Network and Service Management*, 20, (3), 2023, pp. 3119-34

[6] Y. Hao, J. Cao, Q. Wang, and J. Du: ''Energy-aware scheduling in edge computing with a clustering method'', *Future Generation Computer Systems*, 117, 2021, pp. 259-72

[7] V. Srinadh, and P. V. N. Rao: 'Implementation of Dynamic Resource Allocation using Adaptive Fuzzy Multi-Objective Genetic Algorithm for IoT based Cloud System', in Editor (Ed.)^(Eds.): 'Book Implementation of Dynamic Resource Allocation using Adaptive Fuzzy Multi-Objective Genetic Algorithm for IoT based Cloud System' (2022, edn.), pp. 111-18

[8] Z. Zhao, K. M. Barijough, and A. Gerstlauer: ''DeepThings: Distributed Adaptive Deep Learning Inference on Resource-Constrained IoT Edge Clusters'', *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37, (11), 2018, pp. 2348-59

[9] A. Bali, M. Al-Osta, S. Ben Dahsen, and A. Gherbi: ''Rule based auto-scalability of IoT services for efficient edge device resource utilization'', *Journal of Ambient Intelligence and Humanized Computing*, 11, (12), 2020, pp. 5895-912

[10] Y. Chang: ''Dynamic Clustering Method for the Massive IoT System'', *Advances in Security, Networks, and Internet of Things*, 2021, pp. 683-91

[11] X. Yuan, M. Sun, and W. Lou: ''A Dynamic Deep-Learning-Based Virtual Edge Node Placement Scheme for Edge Cloud Systems in Mobile Environment'', *IEEE Transactions on Cloud Computing*, 10, (2), 2022, pp. 1317-28

[12] Q. Liu, L. Cheng, T. Ozcelebi, J. Murphy, and J. Lukkien: 'Deep Reinforcement Learning for IoT Network Dynamic Clustering in Edge Computing', in Editor (Ed.)^(Eds.): 'Book Deep Reinforcement Learning for IoT Network Dynamic Clustering in Edge Computing' (2019, edn.), pp. 600-03

[13] U. Ahmed, G. Srivastava, and J. C.-W. Lin: ''Reliable customer analysis using federated learning and exploring deep-attention edge intelligence'', *Future Generation Computer Systems*, 127, 2022, pp. 70-79

[14] S. Rani, S. H. Ahmed, and R. Rastogi: ''Dynamic clustering approach based on wireless sensor networks genetic algorithm for IoT applications'', *Wireless Networks*, 26, (4), 2020, pp. 2307-16

[15] X. Liu, J. Yu, J. Wang, and Y. Gao: ''Resource Allocation With Edge Computing in IoT Networks via Machine Learning'', *IEEE Internet of Things Journal*, 7, (4), 2020, pp. 3415-26

[16] D. Zhao, Z. Zhou, P. C. K. Hung, S. Deng, X. Xue, and W. Gaaloul: ''CTL-Based Adaptive Service Composition in Edge Networks'', *IEEE Transactions on Services Computing*, 16, (2), 2023, pp. 1051-65

[17] J. Youn, and Y.-H. Han: 'Intelligent Task Dispatching and Scheduling Using a Deep Q-Network in a Cluster Edge Computing System', in Editor (Ed.)^(Eds.): 'Book Intelligent Task Dispatching and Scheduling Using a Deep Q-Network in a Cluster Edge Computing System' (2022, edn.), pp.

[18] A. Nassar, and Y. Yilmaz: ''Deep Reinforcement Learning for Adaptive Network Slicing in 5G for Intelligent Vehicular Systems and Smart Cities'', *IEEE Internet of Things Journal*, 9, (1), 2022, pp. 222-35

[19] J. Li, B. N. Silva, M. Diyan, Z. Cao, and K. Han: ''A clustering based routing algorithm in IoT aware Wireless Mesh Networks'', *Sustainable Cities and Society*, 40, 2018, pp. 657-66

[20] Y. Xu, L. Chen, Z. Lu, X. Du, J. Wu, and P. C. K. Hung: ''An Adaptive Mechanism for Dynamically Collaborative Computing Power and Task Scheduling in Edge Environment'', *IEEE Internet of Things Journal*, 10, (4), 2023, pp. 3118-29

[21] F. Song, M. Zhu, Y. Zhou, I. You, and H. Zhang: ''Smart Collaborative Tracking for Ubiquitous Power IoT in Edge-Cloud Interplay Domain'', *IEEE Internet of Things Journal*, 7, (7), 2020, pp. 6046-55

[22] C. Savaglio, P. Gerace, G. D. Fatta, and G. Fortino: 'Data Mining at the IoT Edge', in Editor (Ed.)^(Eds.): 'Book Data Mining at the IoT Edge' (2019, edn.), pp. 1-6

[23] T. Springer, and E. Linstead: 'Adaptive QoS-Based Resource Management Framework for IoT/Edge Computing', in Editor (Ed.)^(Eds.): 'Book Adaptive QoS-Based Resource Management Framework for IoT/Edge Computing' (2018, edn.), pp. 210-17

[24] Y. Xie, Y. Guo, Z. Mi, Y. Yang, and M. S. Obaidat: ''Edge-Assisted Real-Time Instance Segmentation for Resource-Limited IoT Devices'', *IEEE Internet of Things Journal*, 10, (1), 2023, pp. 473-85