Graph Neural Networks and Dominant Set Algorithms for Energy-Efficient Internet of Things Environments: A Review

Dezhi Liao*, Xueming Huang
School of Information Engineering, Huizhou Technician Institute, Huizhou, 516000, Guangdong, China.

***Abstract*—The widespread usage of Internet of Things (IoT) devices opens up new opportunities for automated operations, monitoring, and communications across various industries. However, extending the lifespan of IoT networks remains crucial because IoT devices are energy-limited. This study investigates the convergence of Graph Neural Networks (GNNs) and dominant set algorithms to extend the longevity of IoT networks. GNNs are neural networks that capture complex relationships and node interactions based on graph-structured data. With these capabilities, GNNs are extremely effective at modeling IoT network dynamics, where devices are connected and whose interactions have a significant impact on performance. In contrast, dominant set algorithms are defined as an approach in which nodes of a network function as agents or leaders to perform resource-efficient and resource-distributed communication. A further detailed overview leverages existing techniques to describe GNNs' role in optimizing dominant set algorithms and discusses integrating these technologies into addressing energy efficiency challenges in IoT settings.**

***Keywords*—Internet of Things; energy efficiency; dominant set; Graph Neural Networks**

I. Introduction

A. Context

In the last few years, technology has profoundly reached all areas of society and has transformed them. As technology has progressed during the last couple of shows, it has dramatically shifted our way of life and how we communicate, work, and live. Ultimately, all these lead to our lives based on data-driven decisions and greater connectedness. The digital transformation also benefits from the presence of potentially billions of connected objects, which now can be transformed into intelligent and connected ones through the Internet of Things (IoT) [1]. IoT encompasses a vast range of physical devices, from everyday items like refrigerators and thermostats to complicated equipment for industrial use [2]. These devices feature software, sensors, and a variety of components specifically engineered to capture and transmit data through the Internet [3].

The advancement of connectivity enhances the functionality and convenience of technologies [4]. The rapid growth of the Internet of Things (IoT) is fueled by advancements and widespread online access [5]. At the core of IoT is computing, which seamlessly integrates computers into our routines without us even noticing it. IoT design enables it to work discreetly in the background. Offers users a multitude of benefits and options. In industries and sectors today, IoTs are widely applied to show their flexibility and versatility [6]. One of the most relatable instances is how IoT transforms household tasks within residences, with the help of home technologies, making our daily lives more efficient and convenient.

B. Motivation

Energy efficiency is a critical concern in designing and operating IoT networks, given that most IoT devices operate on limited energy sources such as batteries, which are often difficult or impractical to replace [7]. The continuous operation of these devices, responsible for tasks ranging from environmental monitoring to smart city infrastructure, requires minimizing energy consumption to extend the network's life and ensure uninterrupted operations [8].

Typical ways to boost energy efficiency, like other algorithms such as duty cycling, energy-aware routing, and clustering algorithms, are often difficult to scale or adjust to changing network conditions. With the growth of IoT networks, these problems have become more apparent, and there is a need for new intelligent solutions to solve them. Significant energy savings can be achieved by reducing the number of active nodes through techniques such as dominant set algorithms and optimizing network operations using Graph Neural Networks (GNNs). These new ideas allow for quick changes to the network as they happen, cutting down on energy use and making IoT systems work better and last longer overall.

C. Contribution

The present study contributes to IoT network optimization. First, a comprehensive overview of the combination of GNNs and dominant set algorithms is provided, highlighting the potential of these ideas to enhance the energy efficiency of IoT networks. Second, existing methods are systematically reviewed, and several insights are provided regarding how GNNs are employed to optimize dominant sets and minimize energy consumption dynamically. Third, current challenges and limitations of traditional approaches are identified, and the synergistic use of GNNs and dominant set algorithms as a novel solution to these problems is proposed. Finally, future research directions are outlined, and the importance of scalability, real-time adaptation, and security for developing more efficient and sustainable IoT networks is highlighted.

II. BACKGROUND

This section presents basic information on concepts and technologies that underpin the integration of GNNs and dominant set algorithms for energy efficiency in IoT networks.

A. IoT

The IoT, coined by Kevin Ashton in 1999, represents a global infrastructure in which physical and virtual entities are connected through advanced communications technologies. As defined by the International Telecommunication Union (ITU), IoT enables innovative services. As shown in Fig. 1, the IoT essentially represents a vast network of devices capable of collecting data, exchanging operational information, and performing autonomous tasks. Integrating sensors into various devices, from cell phones to home appliances, makes this functionality possible.

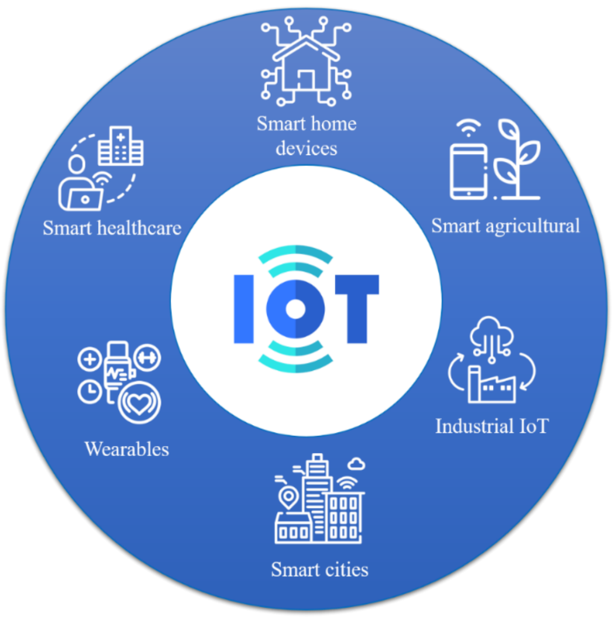

Fig. 1. IoT-related sectors.

An IoT ecosystem includes web-enabled devices with processors, sensors, and communications hardware to collect, transmit, and process data. Collected data is typically transferred to cloud platforms or processed locally before being shared with other connected devices to initiate actions [9]. While human interaction for configuration, guidance, or data access is still possible, IoT devices work independently of each other. IoT applications use specific connectivity, network, and communication protocols. In particular, IoT has the potential to leverage machine learning, a subset of artificial intelligence, to optimize data processing and improve system dynamics. IoT generates and analyzes enormous amounts of data in real-time, driving big data analytics. IoT enables companies to monitor employee performance across multiple locations and optimize operations.

B. Graph Neural Networks

GNNs represent a practical paradigm for processing graph-structured data, demonstrating exceptional performance across diverse domains, including physical systems, protein structure analysis, and knowledge graph management [10, 11]. Graphs can be classified based on edge directionality (directed or undirected), node and edge homogeneity (homogeneous or heterogeneous), and structural complexity (graphs or hypergraphs). Directed graphs exhibit unidirectional edges, while undirected graphs imply bidirectional relationships between connected nodes. Homogeneous graphs contain a single node and edge type, whereas heterogeneous graphs accommodate multiple types. Hypergraphs extend the graph concept by allowing edges to connect arbitrary numbers of vertices. Fig. 2 shows examples for each graph type up to this point.

A directed (simple) graph is formally defined as a tuple $G = (V, E)$, where V is a set of nodes and E is a set of directed edges represented as tuples. A directed (generalized) hypergraph is a similar tuple with hyperedges and a numbering map f_i for each edge to indicate node order. These graphs are considered elementary, as other graphs can be constructed from their composition. Directed graphs are undirected if $(u, v) \in \mathcal{E}$. In this case, edges can be represented as sets rather than tuples. Directed hypergraphs are undirected if $f_i: x \rightarrow \{0\} for\ all\ (x, f_i)_i \in \mathcal{E}$.

A multigraph is a graph where edges or nodes can appear multiple times. A heterogeneous graph is one where nodes or edges have different types. These types can be formally appended to nodes and edges. An attributed graph is one where nodes or edges are associated with attributes, represented by node and edge attribute functions. If only nodes have attributes, it's called node-attributed, and if only edges have attributes, it's called edge-attributed. The graph is considered weighted if edge attributes are a subset of edge types.

The core principle of GNNs is to iteratively aggregate information from neighboring nodes and integrate this aggregated data into the representation of the central node [12]. This process, known as spreading, repeats itself over several layers. Each layer includes aggregation and update operations formulated as follows.

$$\begin{gathered} Aggregation: n_v^{(l)} = Aggregator_l(\{h_u^l, \forall u \in N_v\}) \\ Update: h_v^{(l|+1)} = Updater_l(h_u^l, n_v^{(l)}) \end{gathered} \quad (1)$$

h_u^l represents the node u's representation at layer l, while $Aggregator_l$ and $Updater_l$ denote the respective functions for aggregation and update at layer l.

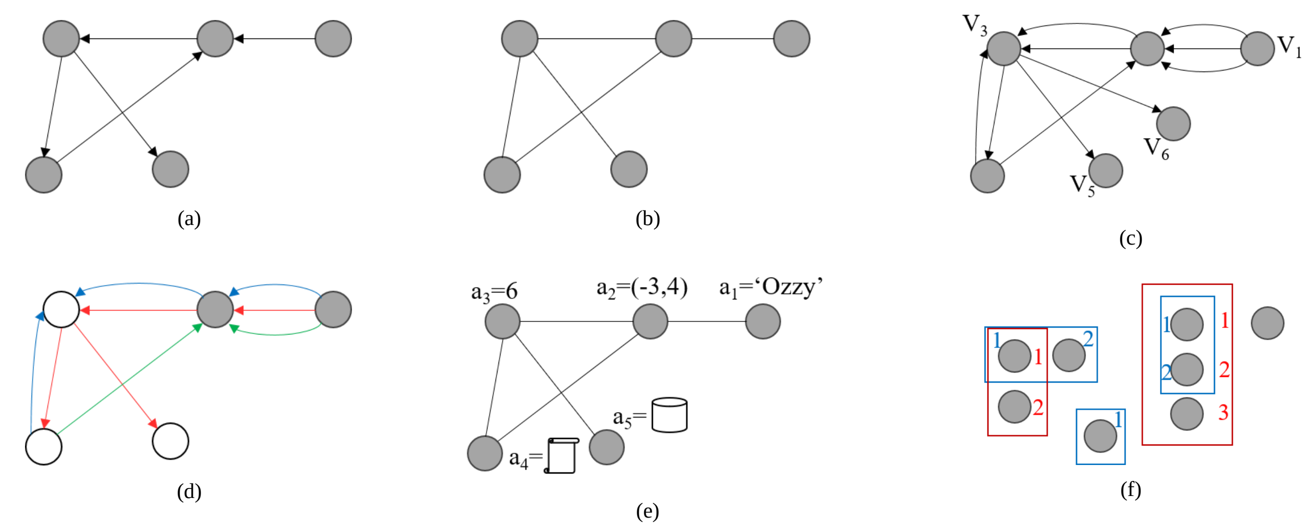

Fig. 2. An overview of different graph types: directed (a), undirected (b), multigraph (c), heterogeneous (d), attributed (e), and directed hypergraph (f).

TABLE I. COMPARISON OF AGGREGATION AND UPDATE OPERATIONS ACROSS DIFFERENT GNN FRAMEWORKS

Framework	**Aggregation operation**	**Update operation**
HGNN	Aggregates information using hyperedge convolution, where the node representations are transformed through a learnable matrix and normalized using degree matrices.	Updates node representations with a non-linear activation function applied to the aggregated information.
GGNN	Aggregates information uniformly from neighboring nodes using mean pooling.	Updates node representations using a gated recurrent unit, integrating the aggregated information iteratively.
GAT	Aggregates information from neighbors using an attention mechanism, where the contribution of each neighbor is weighted based on its relevance.	Updates node representations by applying a non-linear activation function to the weighted aggregation of neighbors' information.
GraphSAGE	Aggregates information from a sampled subset of neighbors using various pooling strategies (mean, sum, max).	Updates node representations by concatenating aggregated neighbor information with the underlying node representation and applying a learnable transformation.
GCN	Aggregates information using a weighted sum of neighbor representations, where the weights are derived from a normalized adjacency matrix.	Updates node representations by applying a non-linear activation function to the aggregated information.

Aggregation strategies include uniform treatment of neighbors (mean pooling) and weighted contributions based on attention mechanisms. The update step integrates the neighborhood information with the central node representation to create a refined node representation. Various techniques have been proposed to effectively combine these components, including GRU mechanisms, concatenation with non-linear transformations, and summation. Generally, five prominent GNN frameworks are commonly employed in IoT networks, each with distinct aggregation and update mechanisms, as summarized in Table I.

1) Hypergraph Neural Network (HGNN): This framework excels at capturing higher-order data correlations within hypergraph structures [13]. The hyperedge convolution layer defined by Eq. (2) uses a non-linear activation function, a learnable transformation matrix ($W^{(l)}$), and degree matrices for edges (D_e) and vertices (D_v) to calculate node representations.

$$\begin{aligned} Aggregation: N^{(l)} &= D_v^{-\frac{1}{2}} E W^0 \tilde{D}_e^{-1} E^T D_v^{-\frac{1}{2}} H^{(l)} \\ Update: H^{(l+1)} &= \delta\left(W^{(l)} N^{(l)}\right) \end{aligned} \tag{2}$$

2) Gated Graph Neural Network (GGNN): GGNN incorporates a Gated Recurrent Unit (GRU) for the update step [14]. While effective, its iterative nature over all nodes can hinder scalability on large graphs.

$$\begin{aligned} Aggregation: n_v^{(l)} &= \frac{1}{|N_v|} \sum_{j \in N_v} h_j^{(l)} \\ Update: h_v^{(l+1)} &= GRU\left(h_v^{(l)}, n_v^{(l)}\right) \end{aligned} \tag{3}$$

3) Graph Attention Network (GAT): Recognizing the varying influence of neighbors, GAT employs an attention mechanism to differentiate neighbor contributions [15]. The attention function, typically $LeakyReLU\left(a^T\left[W^{(l)} h_v^{(l)} \oplus W^{(l)} h_j^{(l)}\right]\right)$, assigns weights to neighbors based on their relevance.

$$\begin{aligned} Aggregation: n_v^{(l)} &= \sum_{j \in N_v} a_{vj} h_j^{(l)}, a_{vj} \\ &= \frac{exp\left(Att\left(h_v^{(l)}, h_j^{(l)}\right)\right)}{\sum_{k \in N_v} exp\left(h_v^{(l)}, h_j^{(l)}\right)} \\ Update: h_v^{(l+1)} &= \delta\left(W^{(l)} n_v^{(l)}\right) \end{aligned} \tag{4}$$

4) GraphSAGE: This framework introduces neighborhood sampling to manage computational efficiency, followed by aggregation (mean, sum, or max pooling) and concatenation for the update step [16].

$$Aggregation: n_v^{(l)} = Aggregator_l(\{h_u^l, \forall u \in N_v\})$$
$$Update: h_v^{(l+1)} = \delta\left(W^{(l)}.\left[h_v^{(l)} \oplus n_v^{(l)}\right]\right) \quad (5)$$

5) Graph Convolutional Network (GCN): GCN simplifies the aggregation process by approximating the graph Laplacian's eigendecomposition [17]The node representation is updated iteratively based on Eq. (6), which involves a non-linear activation function, a learnable transformation matrix, and adjacency weights.

$$Aggregation: n_v^{(l)} = \sum_{j \in N_v} d_{vv}^{-\frac{1}{2}} \tilde{a}_{vj} d_{jj}^{-\frac{1}{2}} h_j^{(l)}$$
$$Update: h_v^{(l+1)} = \delta\left(W^{(l)} n_v^{(l)}\right) \quad (6)$$

C. Dominant Set Algorithms

Dominant set algorithms identify subsets of nodes within a network capable of representing or leading a group of nodes, optimizing communication and resource allocation [18]. In the context of IoT networks, where energy efficiency is paramount, dominant set algorithms are crucial in minimizing the number of active nodes required for effective network operation [19]. By designating certain nodes as "dominant," these algorithms reduce overall communication overhead, conserving energy.

Let $G = (V, E)$ be a graph representing the IoT network, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of IoT nodes, and E is the set of edges representing the communication links between these nodes. A dominant set $D \subseteq V$ is a subset of nodes such that every node $v \in V$ is either in D or adjacent to at least one node in D. The goal is to minimize the size of the dominant set D while ensuring network coverage. This can be mathematically expressed as:

$$Minimize\ |D| \quad subject\ to \quad \forall v \in V, v \in D\ or\ \exists\ u \in D\ such\ that\ (u, v) \in E \quad (7)$$

In energy-constrained IoT networks, the energy consumption of a node v_i is denoted by $E(v_i)$. The total energy consumption of the dominant set D can be expressed as:

$$E(D) = \sum_{v_i} E(v_i) \quad (8)$$

The objective is to minimize $E(D)$ while maintaining a dominant set that covers the entire network, ensuring that:

$$Minimize\ E(D) \quad subject\ to \quad D\ is\ a\ dominant\ set\ of\ G \quad (9)$$

Traditional dominant set algorithms often rely on heuristic or optimization-based approaches to select the most suitable nodes. These algorithms typically consider node degree, connectivity, and proximity to other nodes. For instance, a common approach is to iteratively select the node with the highest degree (most connections) as part of the dominant set:

$$Select\ v_i \in V\ \ such\ that\ degree\ (v_i) \geq degree(v_j) \forall v_j \in V \quad (10)$$

This process is repeated until all nodes are either in the dominant set or adjacent to a node in the dominant set.

One of the main challenges of traditional dominant set algorithms is their static nature. They do not easily adapt to the dynamic conditions of IoT networks, such as fluctuating energy levels or changing network topologies. The computational complexity of finding the optimal dominant set can also be high, especially in large-scale IoT networks.

These limitations have spurred research into more advanced methods, such as integrating dominant set algorithms with machine learning techniques like GNNs. By leveraging the learning capabilities of GNNs, it is possible to develop more sophisticated dominant set algorithms that can dynamically adjust to real-time network conditions, offering a more robust solution for energy-efficient IoT network management. For example, GNNs can predict each node's energy consumption and connectivity dynamics, leading to a more effective and adaptive selection of the dominant set.

D. Energy Efficiency in IoT Networks

Energy efficiency is a paramount concern in the design and operation of IoT networks due to the inherent constraints of IoT devices, which often rely on limited power sources such as batteries. The operational lifespan of these devices and, by extension, the network depends heavily on how efficiently energy is utilized. Given that IoT networks are typically deployed in large numbers and diverse environments, ranging from smart cities to remote agricultural fields, the challenge of maintaining energy efficiency while ensuring continuous, reliable operation is critical. As shown in Fig. 3, the energy consumption in IoT networks is influenced by several factors, including:

1) Communication overhead: Data transmission and reception are among the most energy-intensive activities in IoT devices. The frequency of communication, the distance over which data must be transmitted, and the protocol used all significantly impact energy usage.

2) Idle listening: Nodes in an IoT network often consume energy while listening for potential communication, even if no data is transmitted. This idle listening can account for a substantial portion of energy expenditure.

3) Computation: Local processing tasks, such as data aggregation, encryption, or decision-making algorithms, consume energy. Although typically less than communication activities, computation energy must still be managed effectively, especially in devices with minimal processing power.

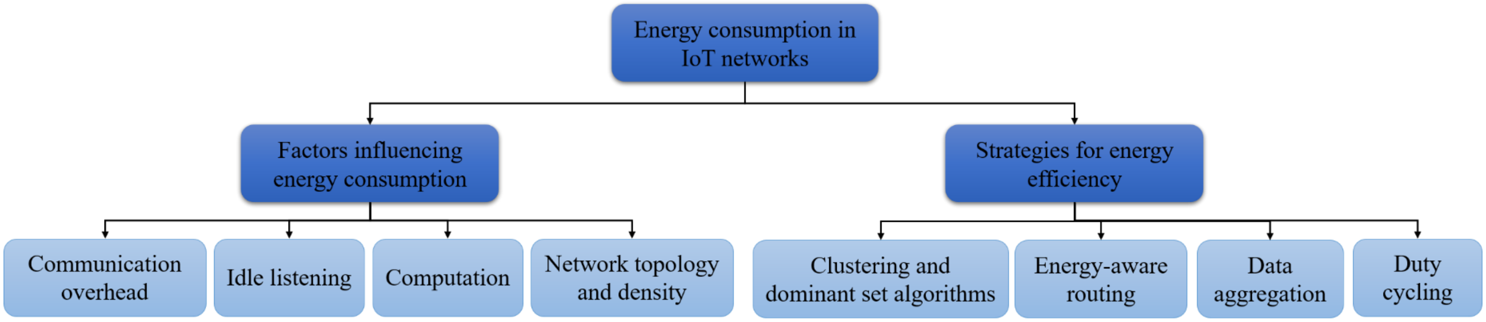

Fig. 3. Energy consumption in IoT networks.

4) Network topology and density: The arrangement and density of nodes within the network affect the routing paths and the number of hops required for data transmission, influencing energy consumption. Dense networks might reduce transmission distances but increase interference and collisions, while sparse networks might require longer transmission distances.

Various strategies have been developed to enhance energy efficiency in IoT networks, including:

1) Duty cycling: This technique involves turning off a device's radio transceiver when it is not needed, thus reducing energy consumption during idle periods. However, the challenge lies in coordinating wake-up times among nodes to maintain network connectivity.

2) Energy-aware routing: Routing protocols in IoT networks can be designed to consider the remaining energy of nodes when selecting routes, thus balancing energy consumption across the network and avoiding the depletion of individual nodes.

3) Clustering and dominant set algorithms: Clustering techniques group nodes into clusters with a designated cluster head, reducing the number of nodes involved in long-distance communication. As discussed in the previous section, dominant set algorithms are a form of clustering that selects a subset of nodes to manage communications, reducing energy consumption.

4) Data aggregation: Reducing the amount of data transmitted by aggregating or compressing data at intermediate nodes can significantly lower energy consumption. This strategy reduces the number of transmissions required and the volume of data sent [20].

Despite the effectiveness of these strategies, challenges remain, particularly in scaling these methods to large and heterogeneous networks, where nodes may have vastly different energy capacities, processing powers, and communication requirements. Furthermore, the dynamic nature of IoT networks, where nodes may move, join, or leave the network, complicates the implementation of static energy-saving techniques.

III. GNNS AND IOT NETWORKS

A. GNNs for Node Representation and Feature Learning

GNNs are effective tools for manipulating and examining graph-based data. This is especially useful for IoT networks, in which devices (nodes) and their connections (edges) naturally form graphs. One of the main advantages of GNNs is their capacity to acquire efficient node representations and extract significant characteristics from the graph. These features may be used for various downstream tasks like optimization, prediction, and classification.

Within IoT systems, individual nodes symbolize devices with distinct features, like energy levels, communication range, computing power, and connectedness to other nodes. The connections between nodes, such as the ability to directly interact with one another, are represented as edges in the graph. Conventional approaches to representing nodes typically depend on predetermined characteristics or rules, which may not completely portray the many relationships and interdependencies in a changing network environment.

GNNs address this constraint by acquiring node representations influenced by the overall network topology. This is accomplished by passing a message in which each node continuously changes its representation by gathering information from neighboring nodes. The whole procedure may be represented numerically in the following manner [21].

$$h_v^{(k)} = \sigma\left(W^{(k)}.AGGREGATE\left(\left\{h_u^{(k-1)}: u \in N(v)\right\}\right)\right) \quad (11)$$

Where $h_v^{(k)}$ is the representation of node *v* at the k^{th} iteration, $N(v)$ denotes the set of neighbors of node *v*, *AGGREGATE* is a function that combines the representations of neighboring nodes (e.g., sum, mean, max), $W^{(k)}$ is a learnable weight matrix for the k^{th} layer, and σ is a non-linear activation function (e.g., ReLU).

During this repeated process, the encoding of each node $h_v^{(k)}$ includes the node's characteristics, the characteristics of its neighboring nodes, and the wider network context. GNNs can acquire comprehensive and contextually aware representations of nodes, which are very useful for tasks like clustering, dominant set selection, and energy-efficient routing in IoT networks. GNNs may capture many crucial components of the network via the learned features, including:

1) *Node centrality:* This feature denotes the significance or impact of a node inside the network.

2) *Connectivity patterns:* It comprehensively represents local and global network configuration, including identifying nodes that form clusters or key routes.

3) *Energy consumption patterns:* It uses predictive analysis to identify nodes more likely to spend more energy depending on their position in the network. This allows for proactive load balancing.

GNNs may enhance decision-making processes in IoT networks by using these acquired characteristics. In the dominant set method, nodes with high centrality and low energy consumption are prioritized as leaders. This optimization aims to improve network coverage and energy efficiency. Utilizing GNNs for node representation and feature learning in IoT networks offers the following benefits:

1) *Scalability:* GNNs can effectively manage networks of significant size by gaining knowledge in a decentralized and concurrent fashion.

2) *Adaptability:* The learned representations may be continuously modified as the network grows, enabling immediate adjustment to changes in network structure or node conditions.

3) *Generalization:* GNNs can apply learned knowledge to diverse network setups, which allows them to remain strong and effective even when faced with varied situations and network architectures.

B. GNNs for Clustering and Dominant Set Selection

GNNs have shown substantial promise in improving IoT network clustering and dominant set selection procedures. Clustering represents categorizing nodes (IoT devices) into smaller groups, known as subsets or clusters. Each cluster is led by a cluster head, who handles communication and coordination inside the cluster. Dominant set selection is a notion that involves choosing a subset of nodes to serve as leaders or representatives responsible for controlling communication throughout the whole network. These processes are essential, enhancing energy efficiency, lowering communication overhead, and lengthening IoT networks' lifespan.

Conventional clustering methods often depend on fixed rules or predetermined standards, which may not completely adjust to the dynamic nature of IoT environments. GNNs provide a more adaptable and data-oriented clustering method by using acquired representations of nodes that include both the local and global structure of the network.

Within the realm of clustering, GNNs generate embeddings reflecting the similarity between nodes, taking into account their characteristics and connections. These embeddings may be used to create clusters more informedly. Nodes that possess comparable energy levels, communication patterns, or responsibilities within the network may be categorized together, establishing more efficient clusters. Mathematically, after learning node representations h_v through a GNN, clustering can be performed by applying a clustering algorithm, such as k-means, directly on these embeddings [21].

$$Cluster\ assignment = k - means(\{h_v | v \in V\}) \tag{12}$$

This methodology guarantees the formation of clusters based on acquired, multi-dimensional characteristics that include intricate interconnections and dependencies within the network instead of merely depending on unprocessed, pre-established criteria.

Dominant set selection entails identifying a subset of nodes that can efficiently handle communication and data aggregation for the whole network. Conventional approaches for choosing dominating sets often include picking nodes based on node degree, closeness, or energy levels. Nevertheless, these approaches may be constrained by their fixed characteristics and incapacity to adjust to evolving network circumstances.

GNNs can be used to dynamically improve the selection of dominant sets by learning to predict the most suitable nodes as representatives. This prediction considers both the network's current state and likely future situations. The method can be defined as a node classification problem in which the GNN is trained to categorize nodes into dominant (leader) or non-dominant (follower) groups based on their acquired embeddings [21].

$$y_v = GNN(G, v) \tag{13}$$

where, y_v is the predicted label for node *v*, indicating whether it should be included in the dominant set. The GNN model considers the entire graph *G* and the specific node *v*, using the message-passing framework to aggregate information from neighboring nodes and the broader network. The inclusion of a node in the dominant set may be determined by a range of acquired characteristics, including:

1) *Energy efficiency:* Nodes with more remaining energy may be prioritized to prevent the depletion of crucial nodes.

2) *Network centrality:* Nodes centrally positioned or with excellent connections are more suitable since they can effectively interact with other nodes.

3) *Load balancing:* The GNN may spread the predominant function across numerous nodes to prevent certain nodes from becoming bottlenecks.

The use of GNNs in clustering and dominate set selection has noteworthy ramifications for IoT networks, especially when conserving energy and ensuring network lifespan is of utmost importance.

1) *Smart cities:* Within urban IoT networks, GNNs can enhance the arrangement of sensors and devices into clusters, decreasing energy use while ensuring a reliable connection and efficient data processing.

2) Industrial IoT: In industrial environments, GNNs may effectively oversee the management of crucial monitoring devices, reducing energy depletion in vital nodes.

3) Environmental monitoring: For IoT networks deployed in remote or difficult-to-access areas, GNNs can help form optimal clusters and select dominant sets that minimize energy usage, extending the network's operational lifespan.

IV. Dominant Set Algorithms in IoT Networks

A. Traditional Dominant Set Algorithms

Traditional dominant set methods were previously employed in IoT networks to enhance network efficiency and minimize energy use by choosing a subset of nodes that efficiently handle communication and data aggregation duties. These algorithms identify a "dominant set" of nodes that may include the whole network, guaranteeing that every node is either a member of the dominant set or is directly linked to a node in the dominant set. This method effectively minimizes the required communication since it only involves a small number of critical nodes in transmitting messages and handling data. This saves energy across the network.

The greedy algorithm is a frequently used method for identifying a dominant set. It operates by repeatedly selecting nodes with the greatest degree (i.e., the nodes with the most connections) to be included in the dominant set. The reasoning is that nodes with a high degree are more likely to include a significant percentage of the network, decreasing the number of nodes required in the dominating set. The fundamental procedures of a greedy dominating set algorithm may be summarized as follows:

Start with an empty set $D=\emptyset$.

Select the node v with the maximum degree in the remaining graph and add it to D.

Remove v and its neighbors from the graph.

Repeat until all nodes are either in D or have a neighbor in D.

Heuristic techniques are often used to enhance the efficiency of selecting dominating sets. These approaches may include other factors such as node energy levels, proximity to other nodes, or the general structure of the network. For instance, a heuristic may prioritize nodes with more energy to prevent the dominant set from rapidly exhausting its resources.

Computing the precise minimal dominating set in large-scale IoT networks may be computationally burdensome. Approximation algorithms provide a means to discover solutions close to optimum efficiently. These algorithms generally ensure that the size of the dominating set is within a certain ratio of the ideal size. An example of a well-recognized approximation technique is the 2-approximation algorithm. This approach ensures that the dominating set it discovers will be no more than twice the size of the best answer.

B. Challenges in Existing Approaches

Although conventional dominating set algorithms have played a crucial role in enhancing the energy efficiency of IoT networks, they still face some difficulties, as outlined in Table II. With IoT networks' growing complexity and dynamism, certain limitations in current techniques have been revealed. These problems highlight the need for more sophisticated approaches to effectively manage contemporary IoT systems' distinct requirements.

A major obstacle classic dominating set algorithms face is their dependence on static network assumptions. These methods typically function using a static snapshot of the network, assuming that the network's architecture, node connection, and energy levels stay unchanged throughout the network's operation. Nevertheless, IoT networks include an intrinsic dynamism, where nodes can join or depart from the network, relocate to new positions, or encounter variations in energy levels. Using a static technique may lead to the formation of inefficient dominating sets that do not adjust to the changing circumstances of the network. This can result in inefficiencies in both energy usage and communication.

TABLE II. Challenges of Conventional Dominant Set Algorithms in IoT Networks

Challenge	Description	Impact on IoT Networks
Static network assumptions	Dependence on a static snapshot of the network, assuming fixed architecture, node connections, and energy levels.	Leads to inefficient dominating sets that do not adapt to dynamic network changes, resulting in energy inefficiencies and communication issues.
Scalability issues	Increasing computational complexity as the network size grows, making it difficult to find efficient dominating sets in large-scale networks.	Limits the applicability of conventional methods in large-scale IoT networks, causing performance bottlenecks and reducing overall network efficiency.
Energy unawareness	Focus on network coverage and connectivity without considering the varying energy levels of individual nodes.	Causes rapid depletion of selected nodes' energy, leading to uneven energy distribution and shortening the overall network lifespan.
Lack of real-time adaptability	Inability to adapt to real-time changes in the network, such as node mobility or environmental fluctuations.	Results in the use of outdated configurations, leading to inefficiencies in energy use and increased energy consumption.
Suboptimal node selection	Reliance on simple heuristics like node degree or proximity, which may not consider the complex factors affecting network performance.	May result in the selection of nodes that are not optimally suited to act as leaders, reducing the network's overall efficiency and effectiveness.
Security vulnerabilities	Algorithms not designed with security in mind, making them susceptible to attacks like node compromise or denial-of-service (DoS) attacks.	Leaves IoT networks vulnerable to security breaches, as crucial nodes in the dominating set may be targeted by adversaries, disrupting network operations.

As the size of IoT networks grows, the computational intricacy of conventional dominating set methods becomes a notable issue. Discovering the most efficient or nearly efficient dominating set in a vast network may be a demanding and time-consuming computing task. The complexity of these algorithms increases exponentially as the number of nodes in the network rises, making it unfeasible to apply them to large-scale IoT networks without experiencing significant performance limitations. The limited scalability of conventional techniques hinders their usefulness in contexts with broad and constantly growing IoT networks.

Traditional dominant set algorithms often focus on optimizing network coverage and connectivity without fully considering the varying energy levels of individual nodes. In many cases, these algorithms may select nodes with low remaining energy to be part of the dominant set, leading to rapid depletion of those nodes' power reserves. This "energy blindness" can result in uneven energy distribution across the network, with some nodes exhausting their energy supply quickly while others remain underutilized. Consequently, the overall network lifespan may be shortened as key nodes fail prematurely due to energy depletion.

Another notable constraint is the lack of real-time adaptability of conventional dominant set methods to network changes. IoT networks often function in dynamic settings characterized by fast changes in circumstances, such as mobile IoT scenarios or networks installed in harsh and fluctuating environments. Conventional algorithms, usually created to calculate a dominating set by a single network examination, have difficulty keeping up with these modifications. Consequently, they could persist in using obsolete settings, resulting in inefficiencies and heightened energy use.

Classic dominant set methods often rely on node degree, proximity, or simple heuristics to select a node. However, these criteria may not necessarily result in the most energy-efficient or effective dominant set. These criteria sometimes fail to include the intricate and multi-faceted elements that impact the functioning of an IoT network, such as the diverse responsibilities of nodes, environmental circumstances, or the unique communication patterns inside the network. This may lead to the selection of nodes not optimally suited to act as leaders, reducing the network's overall efficiency.

Security is a crucial problem in several IoT applications. Conventional dominating set algorithms, on the other hand, are often not developed with security as a primary consideration. The algorithms' static and predictable nature renders them susceptible to assaults, such as node compromise or denial-of-service (DoS) attacks. In these attacks, an adversary targets crucial nodes in the dominating set to disrupt the network. Conventional methods might leave IoT networks vulnerable to possible security breaches without inherent mechanisms to adjust to such threats.

C. Enhancements through GNN Integration

Incorporating GNNs with conventional dominant set techniques signifies notable progress in overcoming the constraints of current methodologies in IoT networks. As summarized in Table III, GNNs can effectively represent and analyze intricate data structures through graphs. This makes them a powerful tool for improving the process of selecting dominating sets and clustering in IoT settings. GNNs provide a flexible, scalable, and adaptable framework for this purpose. This part examines the integration of GNNs with dominant set algorithms to address the issues related to static assumptions, scalability, energy efficiency, and real-time adaptation.

TABLE III. ENHANCEMENTS PROVIDED BY GNN INTEGRATION WITH DOMINANT SET ALGORITHMS IN IOT NETWORKS

Enhancement	Description	Impact on IoT Networks
Dynamic network adaptation	GNNs continuously update node representations in real-time as network conditions evolve, ensuring the dominant set adapts to current network states.	Maintains optimal performance and energy efficiency by dynamically responding to node mobility, energy changes, and communication patterns.
Scalability and efficiency	GNNs handle large-scale IoT networks efficiently through parallel processing and distributed computing, minimizing global calculations.	Enhances scalability of dominant set algorithms, making them suitable for large-scale deployments like smart cities and industrial IoT systems.
Energy-aware node selection	GNNs incorporate energy-awareness into node selection by analyzing current and historical energy data, prioritizing nodes with higher remaining energy.	Extends network lifetime by preventing the premature depletion of key nodes and ensuring balanced energy consumption across the network.
Improved node representation	GNNs generate rich, context-aware representations of nodes that include their characteristics and network context, enabling more informed dominant set selection.	Optimizes energy consumption and maintains network coverage by selecting strategically positioned, well-connected nodes for the dominant set.
Real-time decision making	GNNs support continuous updates and real-time decision-making in volatile network conditions, allowing for ongoing adjustments to the dominant set.	Ensures the network remains efficient and resilient even in dynamic environments like mobile IoT networks or uncertain deployment areas.
Security and robustness	GNNs enhance network security by detecting and responding to abnormal patterns, such as compromised nodes, and adjusting the dominant set accordingly.	Increases the resilience of IoT networks against security threats by proactively isolating or bypassing suspicious nodes, maintaining overall network integrity.

GNNs provide a significant improvement by being able to adjust to changing network circumstances in real-time. Contrary to conventional dominant set techniques that usually function with fixed network snapshots, GNNs continuously modify node representations as network circumstances evolve. GNNs accomplish this dynamic adaptation by using the message-passing process, which involves iteratively updating the representation of each node depending on the information it receives from its neighboring nodes. GNNs can adapt the dominating set to match the current state of the network, which includes factors such as node mobility, changes in energy levels, and different communication patterns. This ensures that GNNs can maintain maximum performance and energy efficiency as the network develops.

GNNs have an innate ability to scale, which makes them highly suitable for accommodating large-scale IoT networks. Conventional dominant set techniques sometimes encounter difficulties due to the high computing cost of finding the most effective nodes within extensive networks. GNNs, on the other hand, are capable of effectively managing enormous graphs via the use of parallel processing and distributed computing. By acquiring data from local neighborhoods in the network, GNNs minimize the need for global calculations, enabling scalable solutions as the network expands. The capacity to scale allows dominating set algorithms to be improved with GNN in large-scale IoT deployments, such as smart cities or industrial IoT systems, where conventional methods may struggle.

One significant benefit of combining GNNs with dominant set algorithms is the capacity to include energy-awareness in selecting nodes. GNNs can analyze node information, such as current energy levels, historical energy use, and connection, to learn and anticipate trends in energy consumption throughout the network. This data may be used to prioritize nodes with more remaining energy or to divide the communication workload equally across the network, thereby preventing the premature exhaustion of crucial nodes. Using GNNs for node selection, focusing on energy efficiency, extends the network's lifetime. This is achieved by preventing any one node from being excessively burdened, resulting in a more equitable distribution of energy consumption.

GNNs are very efficient in acquiring comprehensive and contextually aware representations of individual nodes within a network. The acquired representations include not only the immediate characteristics of the nodes (such as energy levels and connections) and the wider structural environment in which these nodes function. The GNN's learned embeddings include several aspects of a node, such as its centrality, its position in the network, and its closeness to other important nodes. These improved representations enable a more advanced and knowledgeable selection of dominating sets, where the chosen nodes are well-connected and strategically positioned to optimize energy consumption and sustain network coverage.

GNNs can efficiently handle and acquire knowledge from constantly changing data, making them especially powerful when network circumstances are volatile. In scenarios like mobile IoT networks or networks deployed in uncertain areas, GNNs can consistently update the dominating set as nodes relocate or as environmental conditions change. This real-time decision-making capability guarantees the network maintains its efficiency and resilience, even when faced with continuous changes. GNNs may be used with reinforcement learning methods to improve their capacity to adapt and optimize over time, leading to ongoing improvements in the performance of the dominant set algorithm.

By incorporating GNNs into dominant set algorithms, the security and resilience of IoT networks are improved. GNNs may be taught to detect and react to abnormal patterns in a network that may suggest security risks, such as nodes that have been hacked or strange traffic patterns. GNNs may safeguard the network from assaults and preserve its general operation by adapting the dominating set to bypass or isolate suspect nodes. The proactive strategy towards security, together with the robust adaptability of GNNs in managing network dynamics, leads to a more resilient IoT network that can survive various problems.

V. Conclusion

The exponential expansion of the IoT has introduced fresh prospects for automation, surveillance, and communication in several fields. Nevertheless, the energy efficiency of these networks continues to be a significant obstacle because of the constrained power resources of IoT devices. This research has examined the combination of GNNs with dominant set algorithms as a new method to improve the energy efficiency and operational longevity of IoT networks. We have examined conventional dominating set algorithms, emphasizing their use in minimizing communication overhead and preserving energy in IoT networks. Nevertheless, these approaches encounter substantial obstacles, such as their fixed characteristics, problems with expanding to larger scales, insufficient capacity to adjust in real-time, and inadequate consideration of energy fluctuations among nodes. To overcome these restrictions, incorporating GNNs presents a favorable option, as it brings about dynamic adaptability, scalability, energy-conscious node selection, and enhanced node representation, all of which are crucial aspects of IoT network management.

The GNN-based advances outlined in this research, including dynamic network adaptability, real-time decision-making, and security upgrades, signify significant progress compared to conventional methods. These improvements enable IoT networks to be more robust, streamlined, and capable of managing the intricacies of contemporary IoT settings. Nevertheless, there are still obstacles to overcome, such as the need for GNN models that can be easily expanded and operate well, the implementation of strong security and privacy measures, and the meticulous handling of trade-offs between energy efficiency and other factors. To fully harness the capabilities of GNN-based dominant set algorithms in IoT networks, it is essential to prioritize ongoing research and development in this field. To construct more intelligent and adaptable IoT networks that match the expectations of future applications, we may solve difficulties and explore new avenues, such as integrating GNNs with other optimization methods. The combination of GNNs and dominant set algorithms provides a robust foundation for developing IoT networks that are both sustainable and energy-efficient, enabling them to operate well in dynamic and resource-limited contexts.

REFERENCES

[1] B. Pourghebleh and V. Hayyolalam, "A comprehensive and systematic review of the load balancing mechanisms in the Internet of Things," Cluster Computing, pp. 1-21, 2019.

[2] O. A. Ilori, A. A. Willoughby, O. F. Dairo, and A. O. Soge, "Design and Construction of a Photovoltaic Monitoring System Based on Wireless Sensor Networks and Internet of Things Technology," Journal of The Institution of Engineers (India): Series B, pp. 1-16, 2024.

[3] F. Kamalov, B. Pourghebleh, M. Gheisari, Y. Liu, and S. Moussa, "Internet of medical things privacy and security: Challenges, solutions, and future trends from a new perspective," Sustainability, vol. 15, no. 4, p. 3317, 2023.

[4] A. Mukherjee, K. Chatterjee, P. K. Kundu, and A. Das, "Probabilistic neural network-aided fast classification of transmission line faults using differencing of current signal," Journal of The Institution of Engineers (India): Series B, pp. 1-14, 2021.

[5] Z. Lu and X. Deng, "A cloud and IoT-enabled workload-aware Healthcare Framework using ant colony optimization algorithm," International Journal of Advanced Computer Science and Applications, vol. 14, no. 3, 2023.

[6] L. Hu and Y. Shu, "Enhancing Decision-Making with Data Science in the Internet of Things Environments," International Journal of Advanced Computer Science and Applications, vol. 14, no. 9, 2023.

[7] V. Moudgil, K. Hewage, S. A. Hussain, and R. Sadiq, "Integration of IoT in building energy infrastructure: A critical review on challenges and solutions," Renewable and Sustainable Energy Reviews, vol. 174, p. 113121, 2023.

[8] M. H. Alsharif et al., "A comprehensive survey of energy-efficient computing to enable sustainable massive IoT networks," Alexandria Engineering Journal, vol. 91, pp. 12-29, 2024.

[9] B. Pourghebleh, N. Hekmati, Z. Davoudnia, and M. Sadeghi, "A roadmap towards energy - efficient data fusion methods in the Internet of Things," Concurrency and Computation: Practice and Experience, vol. 34, no. 15, p. e6959, 2022.

[10] G. Corso, H. Stark, S. Jegelka, T. Jaakkola, and R. Barzilay, "Graph neural networks," Nature Reviews Methods Primers, vol. 4, no. 1, p. 17, 2024.

[11] B. Khemani, S. Patil, K. Kotecha, and S. Tanwar, "A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions," Journal of Big Data, vol. 11, no. 1, p. 18, 2024.

[12] R. Liu, P. Xing, Z. Deng, A. Li, C. Guan, and H. Yu, "Federated graph neural networks: Overview, techniques, and challenges," IEEE Transactions on Neural Networks and Learning Systems, 2024.

[13] N. Yin et al., "Messages are never propagated alone: Collaborative hypergraph neural network for time-series forecasting," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023.

[14] C. Liu, Y. Li, H. Lin, and C. Zhang, "GNNRec: Gated graph neural network for session-based social recommendation model," Journal of Intelligent Information Systems, vol. 60, no. 1, pp. 137-156, 2023.

[15] U. A. Bhatti et al., "MFFCG–Multi feature fusion for hyperspectral image classification using graph attention network," Expert Systems with Applications, vol. 229, p. 120496, 2023.

[16] T. Liu, A. Jiang, J. Zhou, M. Li, and H. K. Kwan, "GraphSAGE-based dynamic spatial–temporal graph convolutional network for traffic prediction," IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 10, pp. 11210-11224, 2023.

[17] H. T. Phan, N. T. Nguyen, and D. Hwang, "Aspect-level sentiment analysis: A survey of graph convolutional network methods," Information Fusion, vol. 91, pp. 149-172, 2023.

[18] W. Zheng and B. Doerr, "Mathematical runtime analysis for the non-dominated sorting genetic algorithm II (NSGA-II)," Artificial Intelligence, vol. 325, p. 104016, 2023.

[19] B. Alwasel, A. Salim, A. M. Khedr, and W. Osamy, "Dominating sets-based approach for maximizing Lifetime of IoT-based heterogeneous WSNs enabled sustainable smart city applications," IEEE Access, vol. 12, pp. 44069-44079, 2024.

[20] B. Pourghebleh and N. J. Navimipour, "Data aggregation mechanisms in the Internet of things: A systematic review of the literature and recommendations for future research," Journal of Network and Computer Applications, vol. 97, pp. 23-34, 2017.

[21] J. M. Thomas, A. Moallemy-Oureh, S. Beddar-Wiesing, and C. Holzhüter, "Graph neural networks designed for different graph types: A survey," arXiv preprint arXiv:2204.03080, 2022.