IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 September 2022, accepted 3 October 2022, date of publication 5 October 2022, date of current version 13 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3212370

RESEARCH ARTICLE

Context-Aware Service Discovery: Graph Techniques for IoT Network Learning and Socially Connected Objects

AYMEN HAMROUNI[1], (Student Member, IEEE), ABDULLAH KHANFOR[2], (Member, IEEE), HAKIM GHAZZAI[1], (Senior Member, IEEE), AND YEHIA MASSOUD[1], (Fellow, IEEE)

[1]Innovative Technologies Laboratories (ITL), King Abdullah University of Science and Technology (KAUST), Thuwal 23955, Saudi Arabia
[2]College of Computer Sciences & Information Systems, Najran University, Najran 55461, Saudi Arabia

Corresponding author: Yehia Massoud (yehia.massoud@kaust.edu.sa)

This work was supported in part by the King Abdullah University of Science and Technology (KAUST); and in part by the Ministry of Education, Saudi Arabia, and the Deanship of Scientific Research, Najran University, under Grant NU/RC/SERC/11/6.

ABSTRACT Adopting Internet-of-things (IoT) in large-scale environments such as smart cities raises compatibility and trustworthiness challenges, hindering conventional service discovery and network navigability processes. The IoT network is known for its highly dynamic topology and frequently changing characteristics (e.g., the devices' status, such as battery capacity and computational power); traditional methods fail to learn and understand the evolving behavior of the network to enable real-time and context-aware service discovery in such diverse and large-scale topologies of IoT networks. The Social IoT (SIoT) concept, which defines the relationships among the connected objects, can be exploited to extract established relationships between devices and enable trustworthy and context-aware services. In fact, SIoT expresses the possible connections that devices can establish in the network and reflect compatibility, trustworthiness, and so on. In this paper, we investigate the service discovery process in SIoT networks by proposing a low-complexity context-aware Graph Neural Network (GNN) approach to enable rapid and dynamic service discovery. Unlike the conventional graph-based techniques, the proposed approach simultaneously embeds the devices' features and their SIoT relations. Our simulations on a real-world IoT dataset show that the proposed GNN-based approach can provide more concise clusters compared to traditional techniques, namely the Louvain and Leiden algorithms. This allows a better IoT network learning and understanding and also, speeds up the service lookup search space. Finally, we discuss implementing the GNN-assisted context-service discovery processes in novel smart city IoT-enabled applications.

INDEX TERMS Community detection, service discovery, graph neural networks, social Internet of Things.

I. INTRODUCTION

Smart cities are intended to provide improved quality of life for the present and future generations by exploiting the perpetual advances in digital technologies and infrastructure. With over half of the world population currently living in cities and expected to reach almost 70% by 2050 [1], several countries have started investing in innovative smart city initiatives to deliver intelligent technological solutions for their residents and ease their daily life challenges. Furthermore, novel technologies are instrumental in leveraging the quality of civil and public services, especially with the rising demand in cities, and in promoting rich and actionable insights (e.g., alerts). In this context, the Internet-of-Things (IoT) plays a significant role in the process of smartening futuristic cities by generating a tremendous amount of real-time data, called Big Data, that can be used instantaneously to assess a given situation, alert residents and authorities, and/or actuate various devices spread over the city [2]. For example, the United Parcel Service (UPS) uses sensors on its delivery vehicles to monitor speed, miles per gallon, mileage, number of stops, and engine health. The system captures more than 200 data

The associate editor coordinating the review of this manuscript and approving it for publication was Wentao Fan.

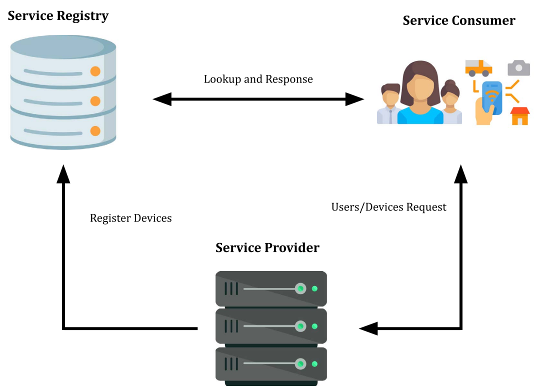


FIGURE 1. An illustration of the widely adopted service discovery technique.

points for each vehicle in a fleet of more than 80,000 daily [3]. This tremendous amount of collected data helps the company reduce idling time, fuel consumption, harmful emissions, and save money.

However, this increasing use of IoT devices, e.g., smart home systems, wearable health monitors, and autonomous vehicles, raises many concerns about security risks, higher network complexity, management, and service interoperability [4], [5]. Moreover, the massive growth of heterogeneous interconnected smart devices, which is estimated to reach 50 billion devices by 2030 [6], will lead to a dense network that is composed of multiple devices requesting various services, inter-connected with different communication protocols, and being vulnerable to internal threats and external attacks. Indeed, one of the main features of practical large-scale IoT systems is the ease of connection and access to various IoT devices. However, this could lead to severe security issues, particularly with the large-scale, widely distributed devices that enable unwanted requests from malicious devices or even initiate Denial of Services (DoS) attacks on the IoT system [7]. Ensuring trustworthy collaboration between IoT devices while preserving their security and privacy will require more efficient and scalable service discovery processes. The latter process is an automated mechanism that allows devices to find and request services from their connected peers in a context-aware manner. However, locating desirable services offered by a corresponding object is not straightforward in a diversified and large-scale network. Moreover, service discovery is highly affected by the dynamic nature of the IoT systems where devices may appear and disappear, physically move, and evolve with time, e.g., variation of their battery level and dynamic availability of their computational capacities. These complex issues will eventually limit the IoT benefits, adaptability, and scaling.

IoT systems enable individuals and organizations to collect data, monitor the environment, and use the devices for automated actuation and intelligence [8]. The process of assisting an IoT entity in finding the correct service, primarily its provider in the large-scale network, is identified as the IoT service discovery. A real-world example of such service discovery could be noticed when a CCTV tracks an object that went out of its coverage zone and requests help from other camera-equipped devices to resume the tracking.

In Fig. 1, the architecture comprises a service provider, a registry, and a consumer. The service provider will first register all the devices available to provide the services. The consumer will request permission from the service provider, and after granting the permission to access a specific device or service, the consumer will lookup through the registry for a suitable device. Finally, the requested service will be provided to the consumer. It is a standard procedure to depend on a centralized registry that collects service consumers' and providers' information to aid in service discovery. Usually, the service registry indicates the service provider's address to the service consumer to achieve the service needed. Despite that, a semantic-based IoT service discovery is required. Our work specifies a semantic-based service discovery as a context-aware approach. In addition, adopting flexible and scalable context-aware service discovery solutions to enhance security and trustworthiness, information exchange and storage, and data analysis capabilities are required to avoid undesired data manipulation and promote the use of IoT devices in a safe, smart city environment. There is a need to design automated solutions to support, facilitate, and analyze the relations between the interconnected IoT entities in such a vast network. Such relations refer to the associations that devices establish between each other such as trust, compatibility, and connectivity. Enabling such solutions guarantees secure, safe, and trustworthy IoT services and data exchange. These solutions need to better understand the network and capture its proprieties (e.g., semantic, spatial, etc.) and general structure. For example, a relationship between two IoT devices with the same owner can exhibit secured communication and service compatibility. Exploiting these IoT devices' relations can provide additional information about the network, promote better decisions, and enable insights into the overall system.

In this context, Social IoT (SIoT), which refers to the social relationships established between IoT devices, was introduced to provide more information and context about the nature of connections between the devices, consequently enhancing the IoT system's relations and trustworthiness [9], [10]. Moreover, the SIoT, by establishing connections between the available devices, helps model the IoT network with semantic and latent relations while capturing important information such as the devices' characteristics and capabilities. Applications relying on human social network analysis and tools, such as Blogjects [11] and Things Twitter [12], were investigated in several studies to benefit the IoT system. However, IoT is much more heterogeneous and complex due to the heterogeneity of its entities. These entities vary in their technical specifications, capabilities, and offered services. Moreover, the relations interconnecting these entities vary based on criteria such as location, ownership, and nature of collaboration. Nevertheless, IoT devices' relations are similar

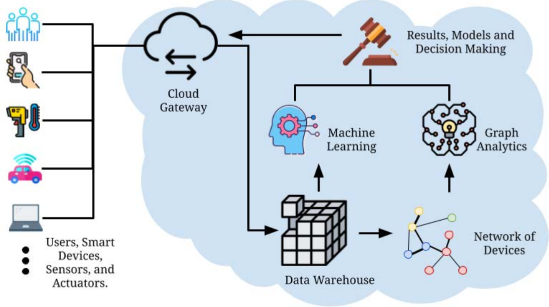

Results, Models and Decision Making
Cloud Gateway
Machine Learning
Graph Analytics
Network of Devices
Data Warehouse
Users, Smart Devices, Sensors, and Actuators.

FIGURE 2. A high-level architecture of the IoT system where machine learning and graph analytic tools can be jointly utilized to perform real-time service discovery and decision-making processes.

to the ones of people's social relations, which are opportunistic networks. Thus, the concepts used in social network analysis can benefit from establishing trustworthy relations between the devices to aid service discovery.

One of the SIoT concepts' main goals is to enable reliable service discovery and navigation methods to serve in a large-scale and complex smart city system such as Intelligent Transportation Systems (ITS), university campuses, or industrial zones (i.e., industrial IoT). The service discovery in SIoT assures the trustworthiness among the connected devices. Furthermore, it aims to exploit the SIoT relations interconnecting the social objects, such as their geolocation, social relations, and relations establishment policies, to deliver more effective and trustworthy service discovery outputs [13].

Nevertheless, even when relying on SIoT, providing a solution for the challenge of service discovery is a complex task. The profoundly dynamic and substantial number of devices connected to the SIoT hinders the raw use of SIoT. Exploiting the relations between the devices as a solution relies heavily on how and what information to extract. The SIoT devices and their different relations can be modeled as a mathematical graph to leverage the variety of the existing graph analytic tools. In Fig. 2, a conceptual architecture for a SIoT data analytical framework is provided. The figure illustrates how graph analysis and machine learning techniques can be used to perceive the SIoT system's structure via a cloud gateway to exchange the necessary data, such as the location and specifications of the devices. From this information and other IoT devices' features, graphs that model the different social relations between the devices can be determined and exploited to devise effective context-aware service discovery techniques in large-scale IoT networks [14].

This paper overviews graph-based techniques to enable service discovery processes in large-scale IoT networks. We propose leveraging the SIoT concept to create graphs reflecting the different relations that can interconnect the devices. First, we present the main SIoT relations and discuss traditional community detection techniques, mainly the Louvain and Leiden methods. We also highlight their use and application in facilitating service discovery and helping determine suitable devices able to address the required services. Afterward, we propose a low-complexity deep learning approach as an alternative solution to these conventional unscalable methods. We design a context-aware community detection approach based on Graph Neural Networks (GNNs). This approach involves the neural network components to generate an embedding for each node of the graph while capturing not only the social relations in the network, as with the traditional methods but the devices' features such as battery capacity, computational resources, existing sensors, etc. The GNN approach is followed by an embedding-based clustering technique to determine several groups for different contexts and with strong social relations. The proposed approach allows a better understanding of the network's nodes by simultaneously incorporating the devices' features and relations. It also helps in shrinking the service space. We show that the GNN approach speeds up the service lookup search space and outperforms the traditional graph-based techniques. Finally, we present practical IoT-enabled applications where service discovery using the SIoT concept can be massively leveraged.

The contributions of this paper can be summarized as follows:

- We overview the service discovery in SIoT and highlight its potential in enabling a more reliable and trustworthy network navigability in the IoT network.
- We provide a practical approach to model SIoT relationships such as co-location, social friendship, and ownership relations to ensure robust service allocation.
- We design a context-aware community detection approach, based on Graph Neural Networks, that maps devices' attributes and their social relationships from a 3-D format into a 2-D format then clusters them to form groups of communities.
- We conduct extensive experimental evaluations on a real-world dataset and show the effectiveness the designed neural network based community detection approach for enabling service discovery, specifically in reducing the services lockup search space and reducing the problem complexity.
- We provide some practical applications in large-scale IoT networks where the developed context-aware service discovery approach can be applied.

The remainder of this paper is organized as follows. Related work is reviewed in Section II. In Section III, we discuss the SIoT relations while highlighting the different possible virtual relationships between the devices in the network and pointing out two conventional community detection techniques. After that, Section IV presents a deep learning approach for community detection based on GNNs. Later on, we include several simulations and experimental results in Section V to evaluate and compare the different community detection approaches. In Section VI, we showcase several use cases of community detection in SIoT networks for various practical approaches such as mobile crowdsourcing, edge computing, and real-time navigation. Finally, Section VII concludes the paper.

II. LITERATURE REVIEW

This section reviews the related literature that covers service discovery and graph analytics in IoT systems. We discuss some of the studies that approached two classical community detection techniques, mainly Louvain and Leiden algorithms, and go through some of the GNNs approaches applied to enable graph nodes' clustering in general.

Over the last decade, several service discovery approaches in IoT systems have emerged [15]. They can be mainly encapsulated into three categories: directory-based architecture, directory-less architecture, or both [16]. In the directory-based architecture, an IoT device can act as a service provider, service consumer, or service directory. Service providers register their services to the service directory, and service consumers become aware of the available services by querying the service directory. On the other hand, directory-less architectures do not rely on directories. Service providers and service consumers interact directly to advertise and retrieve services according to predefined patterns. A hybrid architecture considers the coexistence of the directory-based architecture and directory-less architectures. In this case, service providers can register their service to a service directory if they locate any in their vicinity or broadcast service advertisements. In the same way, service consumers can query the service directory or broadcast their requests and wait for responses from service directories or other IoT devices. These approaches, although proven to be useful under certain conditions [17], cannot be applied for large-scale IoT systems since they are based on the traditional Universal Description Discovery and Integration (UDDI) registry, which suffers from maintaining the state availability of services and generating an unnecessary amount of traffic in the network. Alternative approaches proposed the use of synthetic description languages, inspired by the traditional web services, which provide a description for the devices using a key-value structure and allow the discovery of the services to be performed using keywords matching [18]. However, these methods lack the ability to enable several services, such as the criteria-dependent ones (e.g., the geolocation), since an exact keyword matching for the producer and the consumer in the IoT system is needed to query the desired service.

On the other hand, service discovery can also be viewed from two different categories: 1) the protocol-based IoT service discovery and 2) the semantic-based IoT service discovery [15]. The first category is the widely adopted and standard procedures used to assist in the interaction between the devices in the network. These protocols are easy to apply and well defined, but they assume homogeneous communications and interactions between the devices in the IoT system. Therefore, this category is protocol dependent and does not sustain systems interoperability. The second category, on the other hand, relies on the use of semantic web technologies. In the literature, these approaches use ontologies to give a semantic description for web services. However, applying the semantic web directly on constrained devices imposes an excessive load on these devices. With all of this being said, the nature of IoT systems is large and dynamic. These protocol-based solutions can be useful in the context of small-scale IoT networks. However, on a large scale, the semantic and synthetic descriptions added to provide sufficient service discovery generate massive data traffic to the system.

Other state-of-the-art studies approached service discovery from another perspective and relied on the SIoT paradigm and graph analytic tools. In fact, as devices with their different SIoT relations can be modeled as a mathematical graph, the analytical tools enable the interpretation of the interactions between these devices and provide clear ideas of the overall structure. This is mostly done by developing a computation method based on the graph attributes (e.g., their relationships) to extract several information. For instance, Amin et al. [19] highlighted the importance of using the SIoT and the analysis of such networks to serve numerous research fields and applications. The authors underlined the significance of using analytical tools in SIoT social networks to reveal the structure and its effects, also addressing the challenges such as the network. Premarathne et al. [20] proposed a trust computation method in SIoT to compute the strength of the trust among SIoT members based on the pre-defined set of social relationships they have. Furthermore, using graph analytic tools helps in service discovery and trustworthiness management for SIoT systems [21]. The SIoT concept can be modeled as a dynamic and complex network with heterogeneous devices and relationships [22], [23]. Therefore, considering the networks' temporal and spatial aspects is crucial to SIoT networks. The relationships between the objects raise the emergence of trust and friendlessness between objects for navigating the network [24], by identifying the different trusts, such as direct trust where the two objects establish a link, and indirect trust, such as a friend of friend relationship between the devices. By exploiting the relationships among devices to recommend services or providers, a lot of applications can be enabled, such as predicting new relations or nodes [25], [26] in the SIoT. An additional application is to model the relationships differently using a hyper-graph model in which each hyper-edge connects the users, objects, and services in an IoT system and conducts further network analysis [27]. It can further assist the diverse groups of devices with common characteristics based on their relationships [28].

A community detection paradigm is also an interesting approach to identifying communities in real-world graphs. The main idea behind community detection is to identify the communities in real-world graphs. However, its applications are not limited to finding different communities within a network but also extend to analyzing these communities and explaining why they are clustered together. A community is defined by Flake et al. [29] as a set of nodes that are connected to a greater extent than the rest of the network. Radicchi et al. [30] extended the definition with notions of a strong community and a weak community based on the density of the links within the community. Moreover, the promise of community detection is to gain a deeper understanding of a complex system by revealing the structural patterns within a network structure. In this regard, one of the challenges that are apparent in IoT graphs is the high volatility and dynamism of the networks. Several nodes will be introduced or removed from the system, which increases the complexity of computing and finding communities in such a dynamic network. Some GNNs approaches have targeted community detection. In fact, Chen et al. [31] proposed to solve community detection problems in supervised learning settings using Line Graph Neural Networks (LGNNs). They showed that their model achieves close performance to Belief Propagation (BP) under certain simplifications and assumptions. However, their results were not conducted on SIoT, and they were purely based on the hypothesis that the number of communities to be detected is fixed. In [32], a service discovery process for a mobile crowdsourcing application where a conventional community detection technique, either Louvain or OSLOM, is implemented followed by a Natural Language Processing (NLP) solution to recognize the submitted requests and extract information to match them with the devices of IoT dataset and select the suitable works. Agrawal et al. [33] presented various existing GNN techniques and discussed several simulations and comparisons with different human social network datasets.

All of these related works have either discussed classic web service discovery approaches lacking applicability and scalability for heterogeneous large-scale IoT networks, proposed techniques to enable community detection in large-scale IoT systems, or only focused on clustering communities in human social networks. Moreover, such approaches cannot be applied in large-scale IoT systems since they suffer from maintaining the state availability of services and generating an unnecessary amount of traffic in the network. In other words, such techniques are able to allocate a service in a large-scale network but with a high time-complexity. Alternative approaches that use synthetic description language are relatively faster but they are criteria-dependent, which means that they fail to provide some services such as geo-location matching where the device is looking for a service from another device in a specific region. To the best of our knowledge, we are the first to study service discovery in large-scale IoT networks while leveraging neural network techniques to exploit SIoT relations. The novel contribution in our paper is the design of a neural network architecture that efficiently divides the devices into several communities that encapsulate the provided services, while exploiting the social relations of the SIoT. By embedding the SIoT graph and the devices' characteristics using GNN, the limitations of the conventional service discovery methods can be surpassed by providing an effective service lookup.

III. RELATIONSHIP MANAGEMENT IN SIOT AND CONVENTIONAL COMMUNITY DETECTION METHODS

In this section, we explain the idea of relationship management between IoT devices. We start by defining the concept of SIoT and show the purpose behind enabling service discovery. After that, we present several possible relationships that can be considered among the IoT devices while mentioning two traditional community detection techniques leveraging the grouping of similar devices to detect their provided services.

A. SOCIAL INTERNET-OF-THINGS

The concept of SIoT has put forward the idea of socialization in IoT, where devices can establish social relations, autonomously similar to people's social networks, to indict a collaboration or permission between each other. In other words, IoT devices are allowed to establish social relationships under certain conditions, where they can exchange information and build their own network. Each IoT device has its ego-centric network. For example, devices sharing the same communication protocols and located in the same region can be leveraged with their computing resources as a temporary distributed data center in the case of critical system failure (e.g., traffic light control center outage). The adoption of the SIoT paradigm presents several advantages. In fact, the resulting structure of the things' social network can be shaped as required to guarantee network navigability to perform the discovery of objects and services effectively and to guarantee scalability as in human social networks. Also, a level of trustworthiness can be established for leveraging the degree of interaction among things that are friends. Furthermore, models designed to study social networks can be reused to address IoT-related issues (e.g., service discovery, privacy, etc.). Network navigability establishes different types of connections between IoT devices in an efficient and scalable manner to serve in service discovery [34]. In addition, trustworthiness management is required to ensure a reliable and secure IoT system. This being said, SIoT still presents some challenges [35]. Heterogeneity, for example, is one as the network contains different natures of objects leading to issues such as interoperability and compatibility. Morover, resource-constrained devices also present limitations to the SIoT network. In fact, the network contains devices with limited resources and this issue has a direct impact on the life of the network and the exchange of information. Further SIoT challenges and limitations could be found in recent studies [36], [37].

B. TYPE OF RELATIONS

Relationship management [43] in an IoT context is concerned with providing a dynamic and intelligent method that builds connections between the devices in which the objects can realize friends and foes to start, update, or terminate friendships with other objects. Efficient relationship management can assist in requesting the relevant services from a trustworthy object in the IoT network, according to Afzal et al. [44]. The type of relationships between objects can be established based on the usage and context of the SIoT. There are several evolving events that can occur in a network of devices. First, the different types of relationships between the devices and the devices' features can change over time. The network's

TABLE 1. Qualitative comparison of our proposed framework with recent relevant studies.

IoT Graph Study	Weighted Graphs	Year	IoT Network Size	Social Relations	Clustering algorithm	Application
[19]	No	2018	Medium	OOR	Louvain Algorithm, Statistical, & Greedy method	Relationship Management
[28]	Yes	2019	Large	CLOR & SOR	Louvain & Bron-Kerbosch Algorithms	Relationship Management & Service Discovery
[38]	No	2016	Small	CLOR	Modified Clustering Head & LEACH Algorithms	Relationship Management & Network Navigability
[39]	No	2018	Small	CLOR	Heuristic-based & Graph-based clustering algorithms	Relationship Management & Network Navigability
[40]	Yes	2019	Large	CLOR	Bayesian Probabilistic Graphical Model and quality-based clustering algorithm	Trustworthiness Management
[41]	Yes	2020	Medium	CLOR	Multiple Kernel Clustering	Clustering performance Relationship Management
[42]	Yes	2022	Large	SFOR	Hybrid Artificial Neural Network & K-means++	Service Discovery
Proposed	Yes	2022	Large	CLOR & SFOR	GNN-based Approach	Service Discovery

spatial properties can also change over time with the devices' motions in different positions within the system. With regard to the temporal aspects of the graph, the features of the devices and the relations between them change over time. The most important and usually used social relations in SIoT are presented as follows [23]:

- **Parental–child relation (PCR):** In this type of relationship, a tree structure is formed. Thus, each node in the graph is connected to the upper-level node. For example, ZigBee and IEEE 802.15.4 protocol layers show the connected devices. This type of graph or network is known as a tree or hierarchical topology. Furthermore, Nitti et al. [34] defined this type of network as similar objects built in the same period by the same manufacturer. This definition disregards the structure of the network and the limitation of the definition and implies it will result in a limited effect on the network topology in many use cases.
- **Co-Location/Co-Work based Relation (CLOR):** The network structure is influenced by either the spatial or other types of tasks. Therefore, co-location can be a network of nodes that shares a physical location (co-habituation) or is within the network domain. Similarly, the type of tasks may need a set of nodes, devices, or objects to accomplish the task.
- **Social Object Relation (SOR):** This is the relationship established when the objects come into contact with a network. These relationships can be sporadically or continuously based on the needs of the policies of the devices. Social objects can be based on the same types of objects, such as devices or people looking for specific services.
- **Object Ownership Relation (OOR):** This is based on the ownership of the object for different objects. This network can be influenced by how the objects are connected to each other. As the simplest example, the ownership of a set of smart devices can control the connectivity and type of authentication needed to access this set of devices. Devices such as phones, tablets, game consoles, and home sensors can be connected based on these devices' ownership.

In a previous study [32], we have combined the SOR and OOR relations and proposed a new relationship called Social Friendship and Ownership Relation (SFOR). This SFOR relation can be defined as follows:

- **Social Friendship and Ownership Relation (SFOR):** This relationship is established by considering the social relationships of the owners of the IoT devices. For example, two devices having the same owner are assumed to have an SFOR relationship. Another example could be the case of two devices owned by two different entities. However, if these owners have any kind of social relationship (friends or collaborators), they have some privileges regarding accessing their respective devices.

These relationships can be modeled using undirected weighted or multiple graphs. The IoT devices constitute the graphs' vertices, while the edges represent the social relationships between them. Some nodes may not be connected to specific social relations. In that case, an edge will not be established. Moreover, the graph has no self-loop edges for the nodes. Finally, the weights on each edge indicate the strength of a relationship.

In Table 1, we perform a qualitative comparison between recent studies on service discovery and network navigability that uses graph-based algorithms. First, we examine the main characteristics of these studies, such as the type of graphs, the size of the IoT network, the clustering algorithm, and the application domain. To reduce the search space, we notice the employed of diverse conventional clustering techniques such as heuristic and greedy methods. Still, none of them employed learning algorithms except [42] where the IoT devices are clustered based on their computational capabilities and hardware specifications. After that, an Artificial Neural Network (ANN) is used to predict the suitable devices that can execute the requested task.

For the applications, we align the examined papers with the four application types: 1) relationship management, 2) trustworthiness management, 3) network navigability, and 4) service discovery. We can notice the mentioned literature focuses on the service discovery that can help in many challenges facing the IoT systems. For the size of the IoT network, we consider that if the IoT graph contains less than 100 nodes, it is labeled as a small network. We consider the medium-size network between 100 to 2000 devices as medium size IoT network, and with more than 2000 devices, we label it as an extensive IoT network. For the IoT devices, we notice most of the studies are done in simulation, or small-scale experimental settings except [28]. Therefore, utilizing a real-world data set that uses smart cities or smart campus-scale can be helpful to see the impact of proposed techniques in large IoT systems. Besides that, most studies consider small-scale networks and only focus on the devices' geographical locations. The objective is to search devices in the neighborhood, and hence, location-based clustering is performed.

C. SERVICE DISCOVERY

Service discovery describes the process where IoT devices locate each other's offered services in the IoT network. This paradigm enables the automatic detection of the characteristics and features made available by the devices in the network. In fact, IoT devices rely on these announced features in order to target other devices and enable smooth and fast service exchange. Service discovery promotes IoT devices to have information about the other devices in the network. These information include the characteristics and specifications of the available devices such as communication protocols, capabilities, built-in sensors, availability, location, etc. The devices can exploit these information to gain knowledge about the other devices and consequently target the appropriate class of devices that could satisfy their needed services. For example, when requesting a resource allocation for computing sensitive information with privacy concerns, IoT devices must target devices that are considered to be trustworthy.

Several service discovery protocols have been used in content-delivery management and client-server architecture. This usage has been successful in web applications. But in IoT, there are fundamental differences and challenges. One of these challenges is that devices and services are heterogeneous and diverse, based on each entity's need in the network. Furthermore, the massive number of IoT devices that interact directly and indirectly makes the web-applications approaches inadequate. Therefore, conceiving an efficient service discovery procedure in such large-scale networks remains impeded by the system's increasing complexity. In our paper, we exploit the SIoT paradigm to leverage the social relationship between devices. These relationships generate information about the IoT network structure and highlight the connectivity and compatibility of the devices with each other. Furthermore, the different SIoT relations presented in Section III-B help better map the topology of the IoT network, understand the relations between devices to consequently enable a low-complexity service discovery for large-scale IoT systems. In the following section, we refer to some community detection techniques that divide the IoT network into several communities with close SIoT relationships and common features and services as part of the service discovery approach.

D. CONVENTIONAL COMMUNITY DETECTION TECHNIQUES

Community detection has been lately a trending research topic in the field of ubiquitous computing. One of the critical features of social networks is the community's structure. So far, several methods have been proposed to detect communities, which represent the high importance of discovering communities for understanding social networks and detecting the useful and hidden patterns in the aforementioned network. Community detection can help people understand the topology of a network and identify meaningful clusters. Since there can be any number of communities in a given network with varying sizes and properties, community detection is not a straightforward procedure, and some techniques are more adapted to some problems than others. One of the well-known community detection algorithms are the Louvain method [45] and the Leiden algorithm [46] that we will be presented and employed in Section V-B as benchmarking approaches to our proposed method.

IV. GRAPH NEURAL NETWORKS FOR COMMUNITY DETECTION

A large variety of community detection algorithms are applied to IoT network graphs. However, some of these algorithms lack the ability to capture the weights and directions of the edges of the SIoT network. In this study, we aim to enable service discovery in large-scale IoT networks using a more sophisticated approaches to better assess and understand the SIoT network structure and its components. Thus, a probabilistic context-aware approach relying on GNNs is used with the SIoT graphs to reduce the dimensionality of the large-scale network features and relations. Afterwards, an unsupervised machine learning algorithms is applied to cluster the devices into groups of devices sharing common characteristics and social relations. The obtained groups can be then used to enable fast service discovery. The overall mechanism can be divided into two main parts: i) SIoT network embedding and ii) clustering. During the first part, the GNN operates on the graph structure and embeds each device's structure in the SIoT into a two-dimensional

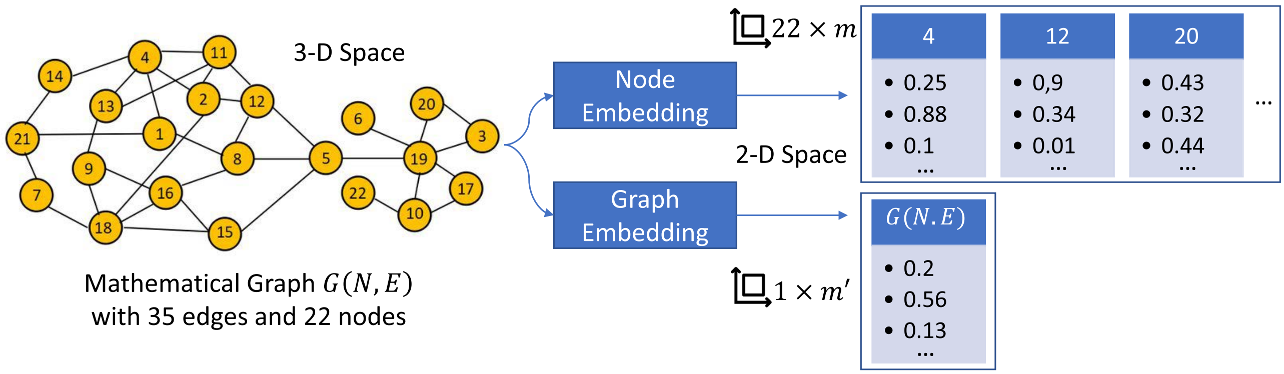


FIGURE 3. Visualization of the embedding process where the original 3-D network is transformed into an embedding space with 2-D vectors. The node embedding techniques transform the graph into a vector with 22 × m dimension, where m represents the embedding dimension. The graph embedding techniques transform the graph into a single vector with m' rows, where m' represents the graph embedding space.

vector. In the second part, an unsupervised machine learning approach is applied to the obtained vectors to perform a clustering analysis and detect communities having common features and sharing strong relations within the SIoT network.

As the goal of the embedding step is to retrieve the SIoT graph semantic and latent complex structure and transform it into a representative 2-D vector space, several approaches with different focuses have been proposed in the literature [47], [48]. Some of them have studied embedding the nodes in the graph [49], where each vertex is encoded with its vector representation, while other approaches designed mechanisms to embed the entire graph [50], where the whole graph is transformed into a single vector. A more figurative explanation is included in Fig. 3, where the difference between node embedding and graph embedding is highlighted. Indeed, each of these embedding types has its specific application.

In our proposed approach, we are interested in extracting information from the SIoT network structure and the IoT devices' specifications (i.e., location, computational characteristics, etc.). Therefore, we proceed with embedding the nodes in the graph. In order to achieve this, we need to explore three stages: 1 mainly) define an encoder F (i.e., a mapping from nodes to embeddings), 2) define a node similarity function (i.e., a measure of similarity in the original network), it specifies how the relationships in vector space map to the relationships in the original network, and 3) optimize the parameters of the encoder, so that similarity of two nodes in the original 3-D network is the same as the one in the resultant 2-D embedding space. Throughout the literature [47], [49], the encoder F was represented as a simple embedding-lookup with $F(a) = y_a = Y \times e_a$ where each column in matrix Y indicates a node embedding. The total number of rows in Y equals the dimension/size of embeddings. e_a is the indicator vector with all zeros except one in the column indicating node a.

The second stage is to define nodes' similarity functions for the original 3-D network, denoted by S_{3D}, and the embedding space, denoted by S_{2D}. The similarity between two nodes a and b in the embedding space can been defined as, $S_{2D}(a, b) = \mathbf{y}_a^\top \mathbf{y}_b$, the dot product between the vectors. Our goal is to define an appropriate S_{3D} and optimize the embeddings such that $S_{3D} \approx S_{2D}$. In other words, the degree of similarity of two nodes in the original 3-D network must be kept in the 2-D embedding network. Deepwalk was among the first proposed mechanisms for node embedding [51]. It proposes a strategy R that runs a fixed-length, unbiased random walk starting from each node in the graph to define S_{3D} and produces the embeddings. However, this technique was proven to have several limitations, including omitting information in the embedded node's neighborhood. The authors of the Node2vec approach have surpassed this issue by proposing a flexible, biased random walk strategy R that can trade-off between local and global views of the network [49]. In fact, Node2vec introduces two parameters p and q. Parameter q defines how probable the random walk would discover the undiscovered part of the graph, while parameter p defines how probable the random walk would return to the previous node. An example of computing the probability to transition from one node to another can be further explained in Fig. 4. The resultant node similarity function is computed as the likelihood of visiting node b on a random walk starting from node a using the random walk strategy R. To determine the embeddings y_a of node a, we optimize and minimize the loss function written as:

$$\mathcal{L} = \sum_{a\in V} \sum_{b\in N_R(a)} -\log \mathrm{P}\left(\mathrm{b}/\mathbf{z_a}\right), \tag{1}$$

where V is the overall set of nodes and $N_R(a)$ is the multiset of nodes visited on random walks starting from a. $\mathrm{P}\left(\mathrm{b}/\mathbf{z_a}\right)$ represents the likelihood of random walk co-occurrences and it is expressed as follows: $\mathrm{P}\left(\mathrm{b}/\mathbf{z_a}\right) = \frac{\exp(\mathbf{z_a^\top z_b})}{\exp(\sum_{\mathbf{n}\in\mathbf{V}} \mathbf{z_a^\top z_n})}$. Note that $N_R(a)$ could have repeated elements since nodes can be visited multiple times on random walks. Two classic strategies to define a neighborhood $N_R(a)$ of a given node a are BFS and DFS. The nested sum over nodes yileds $O(|V|^2)$ complexity. Therefore, we rely on the negative sampling technique where we introduce the sigmoid function σ and P_v

means random distribution over all nodes such that:

$$\begin{aligned}\log\left(\frac{\exp\left(\mathbf{z_a}^{\top}\mathbf{z_b}\right)}{\exp\left(\sum_{\mathbf{n}\in\mathbf{V}}\mathbf{z_a}^{\top}\mathbf{z_n}\right)}\right) &\approx \log\left(\sigma\left(\mathbf{z_a}^{\top}\mathbf{z_b}\right)\right)\\ &\quad-\sum_{\mathrm{i=1}}^{\mathrm{k}}\log(\mathbf{z_a}^{\top}\mathbf{z_{n_i}}),\quad \text{where } \mathbf{n_i}\sim\mathbf{P_b},\end{aligned} \tag{2}$$

with $\mathbf{P_b}$ a random distribution over all nodes. In fact, instead of normalizing with respect to all nodes, we normalize against k random ‘‘negative samples’’ $\mathbf{n_i}$. In this way, we need to sample k negative nodes proportional to the degree to compute the loss function.

Because this embedding procedure captures only the SIoT graph topology, vertex-to-vertex relationship, and sometimes other relevant information about graphs, subgraphs, and vertices, we also propose to design another service discovery approach with an embedding level capable of capturing the nodes' characteristics and features while including them in the resultant output embedding vectors. Hence, the proposed GNN community detection algorithm is more generic and robust.

To this end, the function $F()$ must be carefully defined so that the resultant vector value y_a can include either the SIoT vertex-to-vertex relationship or the devices' features. The encoder is a computational graph with multiple encoding layers in this case. It takes a vector, for example, X_a, 0 for node a, which involves the SIoT relationships and attributes of the node a at layer 0. $F(\mathbf{I}_u^0) = \mathbf{z}_u = \mathbf{I}_u^K$ where $\mathbf{I}_u^0$ is the first layer input and represents the nodes' features and $\mathbf{I}_u^K$ is the optimized embedding and the final layer K output. The neural layers corresponding to layer k for a node u are defined as I_u^k and can be written as follows:

$$F(\mathbf{I}_u^0) = \sigma\left(\mathbf{W}_k\sum_{a\in N(u)}\frac{\mathbf{I}_a^{k-1}}{|N(u)|}+\mathbf{B}_k\mathbf{I}_u^{k-1}\right), \tag{3}$$

$$\forall k\in\{1,\ldots,K\} \tag{4}$$

The initial 0-th layer embedding $\mathbf{I}_u^0$ are equal to the node features $\mathbf{x}_u$ and the optimized embedding $\mathbf{z}_u$ are equal to the final layer embedding $\mathbf{I}_u^K$. The function σ denotes the non-linearity (e.g., relu), and $\mathbf{W}_k$ and $\mathbf{B}_k$ represent the trainable weight matrices that will be adjusted with the loss function. The term $\sum_{a\in N(u)}\frac{\mathbf{I}_a^{k-1}}{|N(u)|}$ represents the average of neighbors's previous layer embedding. A more illustrative pseudo-code for the GNN embedding procedure is included in Algorithm. 1. The outer loop indicates the number of update iteration, while $\mathbf{I}_{N(v)}^k$ denotes the latent vector of node v at update iteration k. At each update iteration, $\mathbf{I}_{N(v)}^k$ is updated based on an aggregation function, the latent vectors of v and v's neighborhood in the previous iteration, and a weight matrix W^k. There are many aggregation function such as mean aggregator, LSTM aggregator, and pooling aggregator. In our paper, we choose the mean aggregator as it takes the average of the latent vector of a node and all its neighborhood and consequently, preserving the passed information from all the nodes' surroundings.

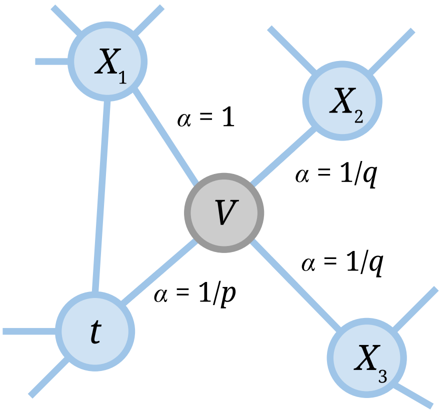


FIGURE 4. Illustration of the graph embedding procedure where the probability to transition from a node v to any one of its neighbors is illustrated.

After determining the vector representation for each node in the SIoT graph, a clustering analysis approach is conducted. The resultant embedding vectors go through a dimensional reduction algorithm (e.g., T-SNE or PCA) to reduce the number of variables in the data and extract only the most important ones from the pool. This step is important as it alleviates the clustering problem, speeds up the computation process, and presents a visualization method in the 2-dimensional space for the clusters. This step is achieved before running an unsupervised machine learning process to group the IoT devices with common features and attributes into SIoT clusters or communities. Throughout the literature, there have been a wide variety of possible machine learning approaches for clustering [52]. Since the task of clustering is subjective, the means that can be used for achieving this goal are plenty. Every methodology follows a different set of rules for defining the similarity among data points. In fact, there are more than 100 clustering algorithms known. But few of the algorithms are used popularly, such as Agglomerative Clustering, DBSCAN, K-Means, OPTICS, Spectral Clustering, and Mixture of Gaussians [53]. The psuedo-code for the community detection included in Algorithm 1 illustrates the clustering process with K-means. The convergence condition for the k-means is where the centroid positions no longer change. The result of the clustering analysis phase gives multiple communities where each community contain devices that are more close and similar to each others in terms of features (i.e., social network and/or characteristics) than to the devices in the other communities. Such clusters, consequently, by design gives context to the devices as each device in a specific community is socially connected to other devices and have similar characteristics.

We should note that in this paper, the mobility of the devices is captured by the CLOR SIoT relation, which reflects the relative positions of the service requesters and providers.

Algorithm 1: Pseudo-Code of the Graph Neural Network Algorithm for Community Detection Using K-Means

```
Input: A SIoT Graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, input features
  $\{\mathbf{x}_u, \forall u \in \mathcal{V}\}$, depth $K$, weight matrices
  $\{\mathbf{W}_k, \forall k \in \{1, \ldots, K\}\}$; non-linearity $\sigma$, neighborhood
  function $N : v \rightarrow 2^{\mathcal{V}}$, number of clusters $k$.
Output: A set of $k$ clusters of devices $\{\mathbf{C_1}, \cdots, \mathbf{C_k}\}$
  and the vector representations $\mathbf{z}_v$ for all devices $v \in \mathcal{V}$.
$\mathbf{I}_u^0 \leftarrow \mathbf{x}_u, \forall u \in \mathcal{V}$.
for $k$=1,...,$K$ do
    for $v \in \mathcal{V}$ do
        $\mathbf{I}_{N(v)}^k \leftarrow \text{AGGREGATE}_k\left(\{\mathbf{I}_u^{k-1}, \forall u \in N(v)\}\right)$
        $F(\mathbf{I}_v^k) \leftarrow$
          $\sigma\left(\mathbf{W}^k \cdot \text{CONCAT}\left(\mathbf{I}_v^{k-1}, \mathbf{I}_{N(v)}^k\right) + \mathbf{B}_k \mathbf{I}_u^{k-1}\right)$
    end
    $F(\mathbf{I}_v^k) \leftarrow F(\mathbf{I}_v^k) \left\|F(\mathbf{I}_v^k)\right\|_2, \forall v \in \mathcal{V},$
end
$\mathbf{z}_v \leftarrow F(\mathbf{I}_v^K), \forall v \in \mathcal{V}$.
Initialize $k$ centroids randomly from $\mathbf{z}_v$.
while not convergence do
    Associate each data point from all the $\mathbf{z}_v$ with the
      nearest centroid, this will divide the data points into
      $k$ clusters in $\{\mathbf{C_1}, \cdots, \mathbf{C_k}\}$.
    Recalculate the position of centroids as mean over
      all assigned points.
end
```

In other words, it takes into account the mobility of devices. To better capture the mobility changes over time in the IoT network, the proposed approach requires updating its relation in each time step and re-predict the new communities to find suitable service providers.

V. RESULTS AND DISCUSSION

In this section, we investigate the performances of the proposed GNN technique for large-scale service discovery. At first, we start by presenting the used dataset and environmental setup. Then, we define some of the evaluation metrics and analyze the behavior of the GNN approach by investigating the embedding and clustering performances to, finally, close the loop by comparing the behavior of the proposed approach with those of the Louvain and Leiden community detection approaches.

A. DATASET AND EXPERIMENTAL SETUP

We use a dataset provided by Marche et al. [54]. The dataset includes real IoT objects in Santander, Spain, mixed with simulated objects such as smartphones, tablets, and personal computer devices. The total number of objects is 16, 216 devices, of which 14, 600 are from private users and 1, 616 from public services. Fig. 5 presents the available information for each device in the dataset, such as the type, the brand, and the model. In addition, the dataset indicates whether a device is static or mobile and provides its geographical locations in the Cartesian coordinate system. A more detailed information about the dataset can be found online.[1] For private devices, additional information about the owner identification is stored in the *user_id*. Moreover, a detailed description of the dataset features is contained in Table 2. In our simulations, we considered two types of SIoT relations: CLOR to capture the location changes between devices and SFOR to consider the devices' trustworthiness and acceptability.

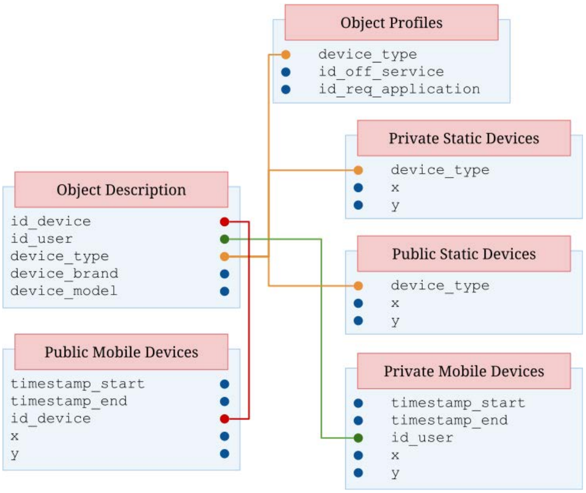


FIGURE 5. High-level entity relationship of the SIoT dataset.

In order to implement the GNN service discovery approach, we employ the node2vec approach as a single layer GNN and set the number of node embedding dimensions to 64. The walking strategy R is set as a 10 random walk from the source with a length of 10 nodes. The parameters P and Q are set to 1. For the multi-layer GNN, we implement the Attributed Network Representation Learning (ANRL) via deep neural networks by Zhang et al. [55]. We set the number of feature embedding dimensions to 256, and increased the walk length and the number of walks to 15 while keeping the other parameters the same. Also, we propose to include a coefficient $w \in [0, 1]$ to reflect the degree of the IoT devices' features that will be taken into consideration. The value of $\omega = 0$ means no attributes were embedded (i.e., single layer GNN), and $\omega = 1$ means all the available attributes were embedded (i.e., multi-layer GNN). In the following experiments, all algorithms are implemented in a Python 3.6 environment and run on a 32 socket Intel(R) Xeon (R) E5-2698 v3 @2.30GHz CPU with 72G of RAM.

B. BENCHMARKING SCHEMES

The Louvain method [45] was proposed in 2008 for detecting non-overlapping communities in a graph by researchers from the University of Louvain, which is how it acquired its name. The technique maximizes the modularity score for each community. Modularity represents the quantification of

[1] http://www.social-iot.org/index.php?p=downloads

TABLE 2. Description of the SIoT dataset features.

Object Feature	Description
Mobility	In this feature, the devices labeled on the ability to operates or functions in different locations. Each devices are labeled as mobile or static.
Type	For each private device, we have further classification where the private devices are Smartphones, Cars, Tablets, Smart Fitness, Smartwatch, Personal Computers, Printers, or Home Sensors. For public devices, the following classification for each device is a point of interest, environment and weather, transportation, garbage Truck, street light, parking, or alarms.
Ownership	This feature used to identify the owner ID of the device. For municipal-owned devices, it will owner ID is 0.
Privileges	Identify one of the following two types either *private* or *public* devices.
Locations	(x, y) coordinate in the Cartesian coordinate system. All of the devices are located in Santander, Spain.
Geotemporal	The location and time of the objects will be identify as quadruple of $(x, y, t_{start}, t_{end})$ where x and y is for the location and t_{start} and t_{end} to indicate the period device where the device is located.

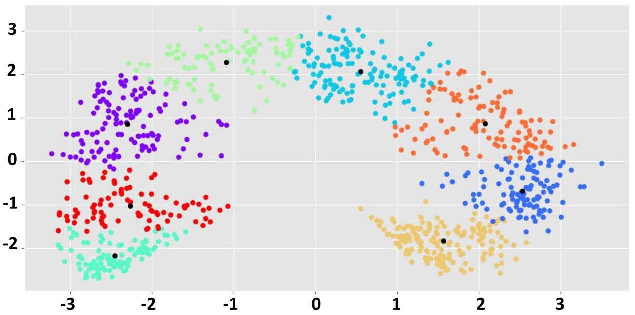


FIGURE 6. 2-D representation showing the IoT devices' embeddings after performing node embedding with $\omega = 0$ and k-means for clustering using CLOR relation. IoT devices belonging to the same cluster have similar colors in the figure as they are highly inter-connected.

the quality of the assignment of nodes to communities by examining the density of the edges of a set of nodes compared to how the set would be connected in a random network. It is a hierarchical algorithm where in every stage, it clusters a set of nodes and converts them into one node with a self-loop representing the edges between them. Then, the condensed graph with the new resultant nodes from the previous step is used for the next level of clustering until the process of grouping becomes stable without new nodes.

Unlike the Louvain algorithm that merges the communities in each level, the Leiden algorithm [46], introduced in 2019, mainly splits and merges the clusters in each level. Therefore, it guarantees that it leads to more well-connected clusters. Compared to the Louvain algorithm, the Leiden algorithm can conduct a fast local move approach. The approach allows the movement of one or more nodes from one cluster to another to improve the clusters' quality in each iteration on finding communities. The selection of the nodes to be moved if and only if the nodes are unstable. This difference improves the running time of the Leiden algorithm compared to the Louvain method. Moreover, the Leiden algorithm can overcome one of the main inefficiencies of the Louvain method. In fact, the latter, in some cases, can generate poorly connected nodes as a community and might lead to a discounted network community [46].

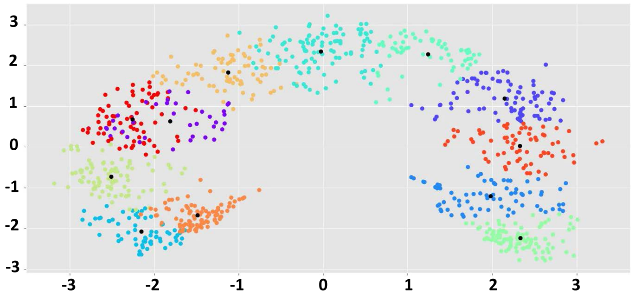


FIGURE 7. 2-D representation showing the IoT devices' embeddings after performing GNN with $\omega = 0.5$ and k-means for clustering using the CLOR relation. IoT devices belonging to the same cluster have similar colors in the figure as they are highly inter-connected and have partially similar characteristics.

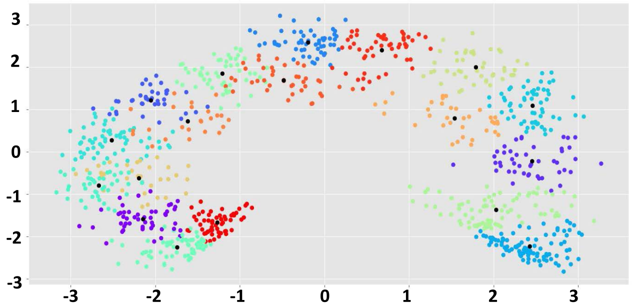


FIGURE 8. 2-D representation showing the IoT devices' embedding after performing GNN with ω =1 and k-means for clustering using CLOR network. IoT devices belonging to the same cluster have similar colors in the figure as they are highly inter-connected and have similar characteristics.

C. EVALUATION METRICS

To assess the quality of the different clustering results, we use three of the standard cluster quality metrics in our study: modularity, coverage and performance. Graph modularity analyzes the presence of each intra-cluster edge of the graph with the probability that that edge would exist in a random graph. It is expressed as follows:

$$Q = \frac{1}{2m} \sum_{vw} \left(A_{vw} - \frac{k_v k_w}{2m} \right) \delta\left(v, w\right),$$

where δ is the Kronecker delta, it equals to one if v and w belong to the same community and 0 otherwise, k_v is the degree of node v, m is the number of edges in the graph, and A_{vw} is the element located at row v and column w of the adjacency matrix $\boldsymbol{A}$. As for coverage metric, it compares the fraction of intra-cluster edges in the graph to the total number of edges in the graph. It is given by:

$$\text{Cov} = \frac{\sum_{i,j} A_{ij} \delta\left(\mathbf{C_i}, \mathbf{C_j}\right)}{\sum_{i,j} A_{ij}},$$

TABLE 3. **Quality metrics illustrating the clustering performances of Leiden algorithm, Louvain algorithm, GNN (Node Embedding only), GNN (Partial node and relation embedding with $\omega = 0.5$ and Full node and relation embedding with $\omega = 1$.**

		Louvain	Lieden	Node Emb. ($\omega = 0$)	Node Emb. ($\omega = 0.5$)	Node Emb. ($\omega = 1$)
CLOR	Number of Communities	6	5	8	12	18
	Largest Community	204	245	100	56	33
	Average community size	153.33	131	114.66	76.58	51.05
	Modularity	0.61	0.6	0.12	0.02	0.01
	Coverage	0.79	0.84	0.8	0.44	0.21
	Performance	0.88	0.87	0.3	0.23	0.12
	Average St. deviation	0.47	0.39	0.13	0.1	0.04
	Max St. Deviation	0.7	0.66	0.25	0.23	0.06
SFOR	Number of Communities	28	26	33	41	72
	Largest Community	72	67	52	28	17
	Average community size	32.75	35.34	45.85	22.3	12.5
	Modularity	0.73	0.8	0.1	0.05	0.03
	Coverage	0.71	0.78	0.9	0.62	0.52
	Performance	0.96	0.95	0.4	0.22	0.21
	Average St. deviation	0.8	0.72	0.2	0.14	0.09
	Max St. Deviation	0.9	0.75	0.25	0.2	0.09

where $\mathbf{C_i}$ is the cluster to which node i is assigned and $\delta(a, b)$ is 1 if $a = b$, otherwise is equal to 0. Coverage falls in the range 0 to 1, and 1 is the highest score that indicates that a graph topology is well-clustered.

The performance P, is another quality function that counts correct vertices also known as ''interpreted'' pairs of vertices, i.e., if there are two nodes in the same community and has an edge, or two nodes in different communities and not connected by an edge. The definition of performance, for a each partition P, is presented as follows:

$$P(\mathcal{P}) = \frac{\left|\{(i,j) \in E, \mathbf{C_i} = \mathbf{C_j}\}\right| + \left|\{(i,j) \notin E, \mathbf{C_i} \neq \mathbf{C_j}\}\right|}{n(n-1)/2},$$

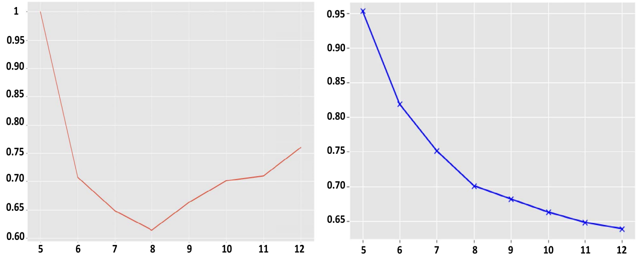
where n is the number of vertices within the cluster and E represents the set of edges of the total graph.

D. PERFORMANCE ANALYSIS OF THE PROPOSED SOLUTION

After performing the embeddings, we obtain the results illustrated in Figs. 6-8. Each figure contains the resultant 2-D vector of the IoT devices in the embedding space (i.e., real vectors). Each point represents an IoT device, and the color reflects the community to which that device belongs. The distance between two nodes represents the distance between the IoT devices in the social network. Moreover, the more two nodes are similar and close in the social network, the closer they are in the embedding space. The clustering process was achieved using a vector quantization technique, namely k-means, which it aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. The cluster centers, *aka* clusters' centroid, are colored in black. The nodes having similar colors are assigned to the same community. For k-means, the number of clusters must be determined a priori. Therefore, we use several clustering validation methods, such as Davies Bloudin and Elbow Distortion. An example of these simulations using node2vec to embed the CLOR network is shown in Fig. 9. The optimal number of clusters corresponds to the Davies Boudin index's minimum value and nearly 20% of the maximum distortion values. Before performing the clustering, we applied a dimensional reduction algorithm, namely Principal Component Analysis (PCA), to reduce the number of variables in the resultant embedding vector by extracting the most important ones.

E. COMPARISON WITH BENCHMARKING APPROACHES

The quality metrics after performing a Monte-Carlo simulation for 1000 community detection simulation using these approaches, along with the Louvain and Leiden approaches for CLOR, and SFOR relationships, are included in Table 3. We notice that the highest number of communities, 18 communities, is outputted by the GNN approach when embedding the whole available features (i.e., GNN when $\omega = 1$) while the Louvain algorithm results in only 6 clusters. The effect of including IoT devices' attributes in the community detection framework results in more communities as the number of resultant communities increases while increasing the number of attributes in the embedding. The GNN embedding for both $\omega = 0.5$ and $\omega = 1$ can be seen as a form of clustering aggregation. In fact, these approaches tend to create clusters inside the Louvain clusters with precise community detection. However, we notice the modularity decreases when adding the attributes in the clustering. This can be explained by the fact that the clustering no longer regroups close IoT devices in the network but rather close and similar IoT devices having relative attributes leading to a decrease in modularity. To furthermore evaluate the behavior of the approaches towards the variation of ω and with respect to Louvain and Leiden algorithms, we have computed the average standard deviation and max standard deviation of the obtained clusters. We can notice that the proposed GNN embedding approach achieves the lowest mean standard deviation with the value of $\omega = 1$. This shows that the formed clusters with the fully embedding approach are very tight and condensed, and hence reflects that the selected nodes in the clusters are highly



(a) Davies Bouldin method

(b) Elbow method

FIGURE 9. Evaluating the distortion of clustering with respect to the number of clusters using (a) Davies Bouldin and (b) Elbow methods after applying GNN to the CLOR graph.

similar. The maximum standard deviation is also minimum for $\omega = 1$ and increases when embedding lower information about the devices. Leiden and Louvain algorithms achieves higher mean standard deviations, which signifies that they induce higher dispersion in the formed clusters.

VI. PRACTICAL APPLICATIONS

In this section, we discuss selected practical smart city and IoT applications that can be leveraged via fast and effective service discovery procedures. We also present high-level graph-based mechanisms applied to the SIoT in order to devise effective service discovery in large-scale networks.

A. MOBILE CROWDSOURCING

Mobile Crowdsourcing (MCS) utilizes the power of mobile devices to accomplish specific sensing and data collection tasks without requiring pre-deployed dedicated infrastructure. MCS and crowdsensing concepts are intertwined with many applications of the IoT systems, such as ride-sharing, transportation, disaster management, agriculture, and community healthcare [56], [57], [58]. The IoT devices that could play the role of service providers and/or service requesters in the SIoT network can be captured and utilized in MCS. As such, the tasks requested by IoT end-users or devices can be crowdsourced to other entities connected to the network, such as mobile devices, users, and vehicles [58], [59]. In MCS, we aim to find trustworthy and capable devices or groups of devices that can share information from their personal sensing activities to satisfy the need of the requested task [60]. Fig. 10 demonstrates a high-level architecture that can be used in MCS applications. The IoT devices will be selected based on the task requirements and the social relationships that may connect the task requester with the existing devices. Hence, using community detection techniques or GNN, the SIoT can be reduced to a manageable set of relevant devices. Then, different recruitment and selection approaches can be applied to the small set of relevant devices to assign the devices that will handle the crowdsourced request.

B. EDGE COMPUTING

The IoT devices possess distinct resource and computational capabilities. Many of them, especially sensors, have minimal computing power. Therefore, this category of devices can seek support from their surrounding peers to share part of their computational and storage resources, if available, to process some of the collected data and hence, relax the load for the cloud servers [61], [62]. To scale the computational resource sharing to a large IoT network scenario, context-aware service discovery in SIoT can look into the extensive network and cluster it accordingly, e.g., the latency that can be reflected via the co-location relations and the devices' trustworthiness based on ownerships and privileges. GNN can also incorporate the edge computers' features in the search, allowing a more accurate search and clustering of the devices. As an example, in [63] and as illustrated in Fig. 11, we present a graph-based community detection on SIoT devices to determine trustworthy edge computers sharing strong social relations with the device requiring data processing. Afterward, a machine learning algorithm is applied to predict the time that will be required by a machine to process the task. Finally, the fastest and most available computing devices are selected.

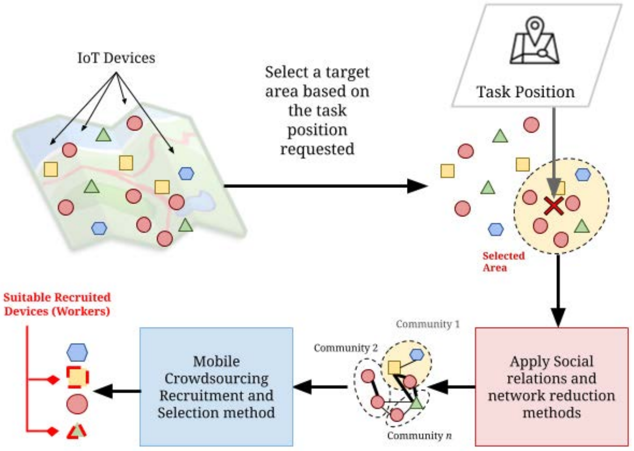

IoT Devices
Select a target area based on the task position requested
Task Position
Selected Area
Apply Social relations and network reduction methods
Community 1
Community 2
Community n
Mobile Crowdsourcing Recruitment and Selection method
Suitable Recruited Devices (Workers)

FIGURE 10. A high-level architecture of a smart SIoT community detection allowing the search of workers in mobile crowdsourcing context.

C. REAL-TIME NAVIGATION

For modern smart cities, interconnected IoT devices can be exploited for a variety of navigation problems such as autonomous unmanned vehicle navigation and pedestrian routing [64]. Several applications can be considered in this context including pedestrians and cyclists in path segments with better air quality [65], [66], [67] or employing the IoT data to assist in practicing social distancing and avoiding crowded areas based on mobile IoT devices [68], [69]. Social network information can also help avoid contact with friends during an outbreak such as COVID-19 [70]. In Fig. 12, we provide a high-level framework showcasing how the SIoT data can be used to determine crowded areas and avoid meeting with friends to help practice social distancing. A weighted graph has the map intersections as nodes, and as edges, the path segments connecting them (e.g., streets) are developed. The weights of the graph correspond to the safety levels of the path segments. They are computed based on the social

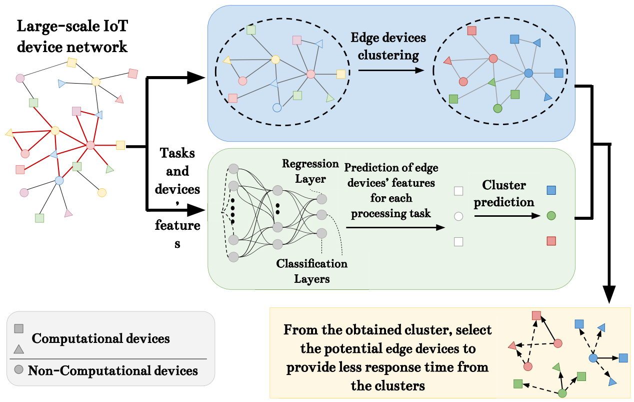

Large-scale IoT device network
Edge devices clustering
Tasks and devices, features
Regression Layer
Prediction of edge devices' features for each processing task
Cluster prediction
Classification Layers
Computational devices
Non-Computational devices
From the obtained cluster, select the potential edge devices to provide less response time from the clusters

FIGURE 11. High-level architecture for edge computer recommendation. The SIoT data and relations are exploited to determine edge computers available to execute the offloaded tasks.

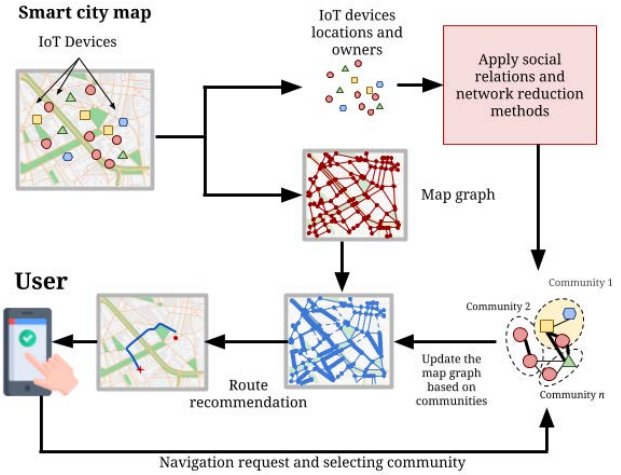

Smart city map
IoT Devices
IoT devices locations and owners
Apply social relations and network reduction methods
Map graph
User
Community 1
Community 2
Community n
Update the map graph based on communities
Route recommendation
Navigation request and selecting community

FIGURE 12. A high-level architecture for safe pedestrian navigation. The SIoT data and relations are used to identify crowded areas and safe path segments where social distancing can be practiced.

relations connecting the devices. By detecting their communities, the CLOR relation helps in determining the crowdedness level in each path segment, while the SFOR relation is used to identify users that may have a close relationship with the navigating user. Finally, the shortest path algorithm is applied to the updated graph to recommend a safe route to the user.

VII. CONCLUSION

In this paper, we investigated graph-based techniques for context-aware service discovery in IoT networks. We discussed the role of social relations among the connected devices in better understanding the network's structure. We have also showcased the role of community detection in shrinking the search space and speeding up the service discovery process. We have proposed to employ the GNN algorithm as a novel tool for community detection in social IoT. We have shown that GNN enables the achievement of a higher quality clustering compared to conventional community detection algorithms due to its ability to embed the nodes' features and relations simultaneously. GNN will play an instrumental role in fostering novel IoT-enabled applications requiring deep network understanding and rapid service discovery search by representing the devices' characteristics with simple numerical vectors. This being said, the quality of the embedding is critical to our approach as the clustering analysis is highly correlated with the embedding efficiency, and consequently, creating a better representation of the nodes results in better performances. Also, the dynamic nature of the network where nodes frequently appear and disappear makes the model susceptible to an overall re-embedding from time to time. In future works, we will investigate using several GNN-based tools such as link prediction and node classification on large-scale IoT networks and study their impact on enhancing the service discovery process while avoiding re-embedding the whole graph when a change happens to the network. We will also focus on leveraging smart city applications by exploiting graph analytic techniques.

REFERENCES

[1] A. Lardieri, "Report: Two-thirds of world's population will live in cities by 2050," U.S. NEWS, Washington, DC, USA, Tech. Rep., May 2018. [Online]. Available: https://www.theguardian.com/world/2018/may/17/two-thirds-of-world-population-will-live-in-cities-by-2050-says-un

[2] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[3] C. Forrest. (2015). *The Power of IoT and Big Data Home Business Internet of Things Ten Examples of IoT and Big Data Working Well Together*. Accessed: Sep. 9, 2022. [Online]. Available: https://www.zdnet.com/article/ten-examples-of-iot-and-big-data-working-well-together/

[4] S. S. Albouq, A. A. A. Sen, N. Almashf, M. Yamin, A. Alshanqiti, and N. M. Bahbouh, "A survey of interoperability challenges and solutions for dealing with them in IoT environment," *IEEE Access*, vol. 10, pp. 36416–36428, 2022.

[5] Y. Chen, S. He, B. Wang, P. Duan, B. Zhang, Z. Hong, and Y. Ping, "Cryptanalysis and improvement of DeepPAR: Privacy-preserving and asynchronous deep learning for industrial IoT," *IEEE Internet Things J.*, early access, Jun. 9, 2022, doi: 10.1109/JIOT.2022.3181665.

[6] *Number of Internet of Things (IoT) Connected Devices Worldwide in 2018, 2025 and 2030(in Billions)*. Accessed: Mar. 5, 2021. [Online]. Available: https://www.statista.com/statistics/802690/worldwide-connected-devices-by-access-technology/

[7] H. F. Atlam and G. B. Wills, "IoT security, privacy, safety and ethics," in *Digital Twin Technologies and Smart Cities*. Cham, Switzerland: Springer, 2020, pp. 123–149.

[8] L. C. Camara Gradim, M. Archanjo Jose, D. Marinho Cezar da Cruz, and R. de Deus Lopes, "IoT services and applications in rehabilitation: An interdisciplinary and meta-analysis review," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 9, pp. 2043–2052, Sep. 2020.

[9] W. Z. Khan, Q.-U.-A. Arshad, S. Hakak, and M. K. Khan, "Trust management in social Internet of Things: Architectures, recent advancements, and future challenges," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 7768–7788, May 2021.

[10] I.-R. Chen, F. Bao, and J. Guo, "Trust-based service management for social Internet of Things systems," *IEEE Trans. Dependable Secure Comput.*, vol. 13, no. 6, pp. 684–696, Nov./Dec. 2016.

[11] J. Bleecker, "A manifesto for networked objects-cohabiting with pigeons, arphids and aibos in the Internet of Things," in *Proc. 13th Int. Conf. Hum.–Comput. Interact. Mobile Devices Services (MobileHCI)*, 2006, pp. 1–17.

[12] M. Kranz, L. Roalter, and F. Michahelles, "Things that Twitter: Social networks and the Internet of Things," in *Proc. What Can Internet Things Citizen (CIoT) Workshop 8th Int. Conf. Pervasive Comput. (Pervasive)*, 2010, pp. 1–10.

[13] A. Hamrouni, T. Alelyani, H. Ghazzai, and Y. Massoud, "Toward collaborative mobile crowdsourcing," *IEEE Internet Things Mag.*, vol. 4, no. 2, pp. 88–94, Jun. 2021.

[14] A. Hamrouni, H. Ghazzai, T. Alelyani, and Y. Massoud, "Low-complexity recruitment for collaborative mobile crowdsourcing using graph neural networks," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 813–829, Jan. 2022.

[15] H. Zorgati, R. B. Djemaa, and I. A. B. Amor, "Service discovery techniques in Internet of Things: A survey," in *Proc. IEEE Int. Conf. Syst., Man Cybern. (SMC)*, Oct. 2019, pp. 1720–1725.

[16] F. Marino, C. Moiso, and M. Petracca, "Automatic contract negotiation, service discovery and mutual authentication solutions: A survey on the enabling technologies of the forthcoming IoT ecosystems," *Comput. Netw.*, vol. 148, pp. 176–195, Jan. 2019.

[17] E. Medina, D. Lopez, R. Meseguer, S. Ochoa, D. Royo, and R. Santos, "Mobile autonomous sensing unit (MASU): A framework that supports distributed pervasive data sensing," *Sensors*, vol. 16, no. 7, p. 1062, Jul. 2016.

[18] S. Sim and H. Choi, "A study on the service discovery support method in the IoT environments," *Int. J. Electr. Eng. Educ.*, vol. 57, no. 1, pp. 85–96, Jan. 2020.

[19] F. Amin, A. Ahmad, and G. S. Choi, "Community detection and mining using complex networks tools in social Internet of Things," in *Proc. IEEE Region 10 Conf. (TENCON)*, Oct. 2018, pp. 2086–2091.

[20] U. S. Premarathne, "MAG-SIoT: A multiplicative attributes graph model based trust computation method for social Internet of Things," in *Proc. IEEE Int. Conf. Ind. Inf. Syst. (ICIIS)*, Dec. 2017, pp. 1–6.

[21] A. Khanfor, A. Hamrouni, H. Ghazzai, Y. Yang, and Y. Massoud, "A trustworthy recruitment process for spatial mobile crowdsourcing in large-scale social IoT," in *Proc. IEEE Technol. Eng. Manage. Conf. (TEMSCON)*, Jun. 2020, pp. 78–83.

[22] M. S. Roopa, S. Pattar, R. Buyya, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, "Social Internet of Things (SIoT): Foundations, thrust areas, systematic review and future directions," *Comput. Commun.*, vol. 139, pp. 32–57, May 2019.

[23] L. Atzori, A. Iera, G. Morabito, and M. Nitti, "The social Internet of Things (SIoT)–when social networks meet the Internet of Things: Concept, architecture and network characterization," *Comput. Netw.*, vol. 56, no. 16, pp. 3594–3608, Nov. 2012.

[24] F. Amin, A. Ahmad, and G. S. Choi, "Towards trust and friendliness approaches in the social Internet of Things," *Appl. Sci.*, vol. 9, no. 1, p. 166, 2019.

[25] A. Aljubairy, W. E. Zhang, Q. Z. Sheng, and A. Alhazmi, "Siotpredict: A framework for predicting relationships in the social internet of Things," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.*, 2020, pp. 101–116.

[26] L. Yao, Q. Z. Sheng, A. H. H. Ngu, and X. Li, "Things of interest recommendation by leveraging heterogeneous relations in the Internet of Things," *ACM Trans. Internet Technol.*, vol. 16, no. 2, pp. 1–25, Apr. 2016.

[27] I. Mashal, O. Alsaryrah, and T.-Y. Chung, "Analysis of recommendation algorithms for Internet of Things," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Apr. 2016, pp. 1–6.

[28] A. Khanfor, H. Ghazzai, Y. Yang, and Y. Massoud, "Application of community detection algorithms on social Internet-of-Things networks," in *Proc. 31st Int. Conf. Microelectron. (ICM)*, Dec. 2019.

[29] G. W. Flake, R. E. Tarjan, and K. Tsioutsiouliklis, "Graph clustering and minimum cut trees," *Internet Math.*, vol. 1, no. 4, pp. 385–408, Jan. 2004.

[30] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi, "Defining and identifying communities in networks," *Proc. Nat. Acad. Sci. USA*, vol. 101, no. 9, pp. 2658–2663, Mar. 2004.

[31] Z. Chen, L. Li, and J. Bruna, "Supervised community detection with line graph neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2019. [Online]. Available: https://openreview.net/forum?id=H1g0Z3A9Fm

[32] A. Khanfor, H. Ghazzai, Y. Yang, M. R. Haider, and Y. Massoud, "Automated service discovery for social Internet-of-Things systems," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Oct. 2020, pp. 1–5.

[33] R. Agrawal, M. Arquam, and A. Singh, "Community detection in networks using graph embedding," *Proc. Comput. Sci.*, vol. 173, pp. 372–381, Jan. 2020.

[34] M. Nitti, L. Atzori, and I. P. Cvijikj, "Network navigability in the social Internet of Things," in *Proc. IEEE World Forum Internet Things (WF-IoT)*, Seoul, South Korea, Mar. 2014, pp. 405–410.

[35] M. Malekshahi Rad, A. M. Rahmani, A. Sahafi, and N. Nasih Qader, "Social Internet of Things: Vision, challenges, and trends," *Hum.-Centric Comput. Inf. Sci.*, vol. 10, no. 1, p. 52, Dec. 2020.

[36] P. Kumaran and R. Sridhar, "Social Internet of Things (SIoT): Techniques, applications and challenges," in *Proc. 4th Int. Conf. Trends Electron. Informat. (ICOEI)(4)*, Jun. 2020, pp. 445–450.

[37] H. Vahdat-Nejad, Z. Mazhar-Farimani, and A. Tavakolifar, "Social Internet of Things and new generation computing—A survey," in *Toward Social Internet of Things (SIoT): Enabling Technologies, Architectures and Applications*. Cham, Switzerland: Springer, Jul. 2019, pp. 139–149.

[38] J. S. Kumar and M. A. Zaveri, "Graph based clustering for two-tier architecture in Internet of Things," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Dec. 2016, pp. 229–233.

[39] J. S. Kumar and M. A. Zaveri, "Clustering approaches for pragmatic two-layer IoT architecture," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–16, 2018.

[40] P. Sun, J. Li, M. Z. A. Bhuiyan, L. Wang, and B. Li, "Modeling and clustering attacker activities in IoT through machine learning techniques," *Inf. Sci.*, vol. 479, pp. 456–471, 2019.

[41] Z. Ren, M. Mukherjee, J. Lloret, and P. Venu, "Multiple kernel driven clustering with locally consistent and selfish graph in industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 17, no. 4, pp. 2956–2963, Apr. 2021.

[42] R. Hamadi, A. Khanfor, H. Ghazzai, and Y. Massoud, "A hybrid artificial neural network for task offloading in mobile edge computing," in *Proc. IEEE 65th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2022, pp. 1–4.

[43] N. Haron, J. Jaafar, I. A. Aziz, M. H. Hassan, and M. I. Shapiai, "Data trustworthiness in Internet of Things: A taxonomy and future directions," in *Proc. IEEE Conf. Big Data Anal. (ICBDA)*, Nov. 2017, pp. 25–30.

[44] B. Afzal, M. Umair, G. A. Shah, and E. Ahmed, "Enabling IoT platforms for social IoT applications: Vision, feature mapping, and challenges," *Future Gener. Comput. Syst.*, vol. 92, pp. 718–731, Mar. 2019.

[45] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech., Theory Exp.*, vol. 2008, no. 10, Oct. 2008, Art. no. P10008.

[46] V. A. Traag, L. Waltman, and N. J. van Eck, "From Louvain to leiden: Guaranteeing well-connected communities," *Sci. Rep.*, vol. 9, no. 1, pp. 1–12, Dec. 2019.

[47] Y. Zhu, Y. Xu, F. Yu, S. Wu, and L. Wang, "CAGNN: Cluster-aware graph neural networks for unsupervised graph representation learning," 2020, *arXiv:2009.01674*.

[48] A. Hamdi, D. Y. Kim, and F. D. Salim, "Flexgrid2vec: Learning efficient visual representations vectors," 2020, *arXiv:2007.15444*.

[49] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 855–864.

[50] A. Narayanan, M. Chandramohan, R. Venkatesan, L. Chen, Y. Liu, and S. Jaiswal, "Graph2vec: Learning distributed representations of graphs," 2017, *arXiv:1707.05005*.

[51] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2014, pp. 701–710.

[52] R. Xu and D. C. Wunsch, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, Jun. 2005.

[53] J. Zhang, X. Hong, S.-U. Guan, X. Zhao, H. Xin, and N. Xue, "Maximum Gaussian mixture model for classification," in *Proc. 8th Int. Conf. Inf. Technol. Med. Educ. (ITME)*, Dec. 2016, pp. 587–591.

[54] C. Marche, L. Atzori, and M. Nitti, "A dataset for performance analysis of the social Internet of Things," in *Proc. IEEE 29th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Bologna, Italy, Sep. 2018, pp. 1–5.

[55] Z. Zhang, H. Yang, J. Bu, S. Zhou, P. Yu, J. Zhang, M. Ester, and C. Wang, "ANRL: Attributed network representation learning via deep neural networks," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3155–3161.

[56] Y. Sun, W. Ding, L. Shu, K. Li, Y. Zhang, Z. Zhou, and G. Han, "On enabling mobile crowd sensing for data collection in smart agriculture: A vision," *IEEE Syst. J.*, vol. 16, no. 1, pp. 132–143, Mar. 2022.

[57] O. Yilmaz, L. Gorgu, M. J. O'grady, and G. M. P. O'hare, "Cloud-assisted mobile crowd sensing for route and congestion monitoring," *IEEE Access*, vol. 9, pp. 157984–157996, 2021.

[58] G. Zhang, D. Lu, and H. Liu, "IoT-based positive emotional contagion for crowd evacuation," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 1057–1070, Jan. 2021.

[59] A. Hamrouni, H. Ghazzai, and Y. Massoud, "Many-to-many recruitment and scheduling in spatial mobile crowdsourcing," *IEEE Access*, vol. 8, pp. 48707–48719, 2020.

[60] A. Hamrouni, H. Ghazzai, T. Alelyani, and Y. Massoud, "An evolutionary algorithm for collaborative mobile crowdsourcing recruitment in socially connected IoT systems," in *Proc. IEEE Global Conf. Artif. Intell. Internet Things (GCAIoT)*, Dec. 2020, pp. 1–6.

[61] P. Zhang, Y. Wang, N. Kumar, C. Jiang, and G. Shi, "A security- and privacy-preserving approach based on data disturbance for collaborative edge computing in social IoT systems," *IEEE Trans. Computat. Social Syst.*, vol. 9, no. 1, pp. 97–108, Feb. 2022.

[62] X. Kong, H. Gao, G. Shen, G. Duan, and S. K. Das, "FedVCP: A federated-learning-based cooperative positioning scheme for social internet of vehicles," *IEEE Trans. Computat. Social Syst.*, vol. 9, no. 1, pp. 197–206, Feb. 2022.

[63] A. Khanfor, A. Nammouchi, H. Ghazzai, Y. Yang, M. R. Haider, and Y. Massoud, "Graph neural networks-based clustering for social Internet of Things," in *Proc. IEEE 63rd Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2020, pp. 1056–1059.

[64] C. Laoudias, A. Moreira, S. Kim, S. Lee, L. Wirola, and C. Fischione, "A survey of enabling technologies for network localization, tracking, and navigation," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3607–3644, 4th Quart., 2018.

[65] A. Pimpinella, A. E. C. Redondi, and M. Cesana, "Walk this way! An IoT-based urban routing system for smart cities," *Comput. Netw.*, vol. 162, Oct. 2019, Art. no. 106857.

[66] V. Matkovic and T. Weis, "Towards enhancing bike navigation safety and experience using sensor enabled devices," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2020, pp. 1–2.

[67] A. Nurminen, A. Malhi, L. Johansson, and K. Framling, "A clean air journey planner for pedestrians using high resolution near real time air quality data," in *Proc. 16th Int. Conf. Intell. Environments (IE)*, Jul. 2020, pp. 44–51.

[68] R. P. Singh, M. Javaid, A. Haleem, and R. Suman, "Internet of Things (IoT) applications to fight against COVID-19 pandemic," *Diabetes Metabolic Syndrome: Clin. Res. Rev.*, vol. 14, no. 4, pp. 521–524, 2020.

[69] M. S. Rahman, N. C. Peeri, N. Shrestha, R. Zaki, U. Haque, and S. H. A. Hamid, "Defending against the novel coronavirus (COVID-19) outbreak: How can the Internet of Things (IoT) help to save the world?" *Health Policy Technol.*, vol. 9, no. 2, pp. 136–138, Jun. 2020.

[70] A. Khanfor, H. Friji, H. Ghazzai, and Y. Massoud, "A social IoT-driven pedestrian routing approach during epidemic time," in *Proc. IEEE Global Conf. Artif. Intell. Internet Things (GCAIoT)*, Dubai, United Arab Emirates, Dec. 2020, pp. 1–6.

AYMEN HAMROUNI (Student Member, IEEE) received the CPGE degree (Hons.) in mathematics and physics from the l'Institut Préparatoire aux Etudes d'Ingénieur de Sfax, in 2016, and the Diplome d'Ingenieur degree *(summa cum laude)* in telecommunication engineering from the Ecole Superieure des Communications de Tunis (SUP'COM), Tunis, Tunisia, in 2019. He is currently pursuing the M.S./Ph.D. degree in electrical and computer engineering with the King Abdullah University of Science and Technology (KAUST). From 2019 to 2021, he was a Research Scholar with the School of Systems and Enterprises, Stevens Institute of Technology, Hoboken, NJ, USA. His research interests include deep learning, optimization, mathematical modeling and algorithm design, graph theory, and the Internet of Things.

ABDULLAH KHANFOR (Member, IEEE) received the master's degree in computer science from the Stevens Institute of Technology, Hoboken, NJ, USA, in 2016, and the Ph.D. degree in systems engineering with a concentration in software engineering, in 2020. He is currently working as an Assistant Professor at the College of Computer Science and Information Systems, Najran University, Saudi Arabia. His research interests include the Internet of Things, graph theory, and machine learning. He received the Exceptional Achievement Award from the School of Systems and Enterprises, Stevens Institute of Technology, in 2020.

HAKIM GHAZZAI (Senior Member, IEEE) received the Diplome d'Ingenieur degree (Hons.) in telecommunication engineering and the master's degree in high-rate transmission systems from the Ecole Superieure des Communications de Tunis (SUP'COM), Tunis, Tunisia, in 2010 and 2011, respectively, and the Ph.D. degree in electrical engineering from the King Abdullah University of Science and Technology (KAUST), Saudi Arabia, in 2015. He was a Researcher Scholar with the Qatar Mobility Innovations Center (QMIC), Qatar; Karlstad University, Sweden; and the Stevens Institute of Technology, Hoboken, NJ, USA. He is currently a Research Scientist with the KAUST. He is the author or coauthor of more than 150 publications. His research interests include artificial intelligence enabled applications, the Internet of Things, intelligent transportation systems (ITS), mobile and wireless networks, and unmanned aerial vehicles (UAVs). He was a recipient of appreciation for an Exemplary Reviewer for IEEE WIRELESS COMMUNICATIONS LETTERS in 2016 and IEEE COMMUNICATIONS LETTERS in 2017. Since 2019, he has been on the Editorial Board of the IEEE COMMUNICATIONS LETTERS and the IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY. Since 2020, he has been an Associate Editor of the Board of IoT and Sensor Networks (speciality section of *Frontiers in Communications and Networks*).

YEHIA MASSOUD (Fellow, IEEE) received the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA. He has held several experiences at leading institutions of higher education and respected industry names, including Rice University; Stevens Institute of Technology, USA; WPI; UAB; the SLAC National Accelerator Laboratory; and Synopsys Inc. From January 2018 to July 2021, he was the Dean of the School of Systems and Enterprises (SSE), Stevens Institute of Technology. Prior to Stevens, he worked as the Head of the Department of Electrical and Computer Engineering (ECE), Worcester Polytechnic Institute (WPI), from 2012 to 2017. In 2003, he joined Rice University, as an Assistant Professor, where he became one of the fastest Rice Faculty to be granted tenure with the Department of Electrical Engineering and the Department of Computer Science, in 2007. He is currently a Professor and the Director of the Innovative Technologies Laboratories (ITL), King Abdullah University of Science and Technology (KAUST), Saudi Arabia. He has published more than 350 papers in leading peer-reviewed journals and conference publications. His research interest includes design of state-of-the-art innovative technological solutions that span over the broad range of technical areas, including smart cities, autonomy, smart health, embedded systems, nanophotonics, and spintronics. He was a recipient of the Rising Star of Texas Medal, the National Science Foundation CAREER Award, the DAC Fellowship, the Synopsys Special Recognition Engineering Award, and several best paper awards. He was selected as one of ten MIT Alumni Featured by the MIT's Electrical Engineering and Computer Science Department, in 2012. He was named a Distinguished Lecturer by the IEEE Circuits and Systems Society, from 2014 to 2015. He has served as the Editor for the *Mixed-Signal Letters-the Americas*, as an Associate Editor for the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS, as well as a Guest Editor for a Special Issue of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: REGULAR PAPERS.