# Graph Neural Networks-based Clustering for Social Internet of Things

Abdullah Khanfor[1], Amal Nammouchi[1], Hakim Ghazzai[1], Ye Yang[1], Mohammad R. Haider[2], and Yehia Massoud[1]
[1]School of Systems & Enterprises, Stevens Institute of Technology, Hoboken, NJ, USA
[2]University of Alabama at Birmingham, AL, USA

***Abstract*—In this paper, we propose a machine learning process for clustering large-scale social Internet-of-things (SIoT) devices into several groups of related devices sharing strong relations. To this end, we generate undirected weighted graphs based on the historical dataset of IoT devices and their social relations. Using the adjacency matrices of these graphs and the IoT devices' features, we embed the graphs' nodes using a Graph Neural Network (GNN) to obtain numerical vector representations of the IoT devices. The vector representation does not only reflect the characteristics of the device but also its relations with its peers. The obtained node embeddings are then fed to a conventional unsupervised learning algorithm to determine the clusters accordingly. We showcase the obtained IoT groups using two well-known clustering algorithms, specifically the $K$-means and the density-based algorithm for discovering clusters (DBSCAN). Finally, we compare the performances of the proposed GNN-based clustering approach in terms of coverage and modularity to those of the deterministic Louvain community detection algorithm applied solely on the graphs created from the different relations. It is shown that the framework achieves promising preliminary results in clustering large-scale IoT systems.**

***Index Terms*—Internet of Things (IoT), clustering, deep learning, graph neural networks.**

## I. Introduction

Internet-of-things (IoT) becomes essential in a variety of civil, public, and military applications, which makes their complexity and size perpetually increasing [1]. The growing number of connected devices requires advanced forms of collaboration to exploit their heterogeneity and improve their services effectively. The Social Internet-of-things (SIoT) concept has been emerged by allowing IoT devices to establish their own social networks [2]. The paradigm aims to aid the smart objects to establish and maintain relations with their peers. The relationships in the network are not exclusive to machine-to-machine but can be extended between the users of the SIoT system, such as machine-to-human or even human-to-human relations. The social relations help assure trustworthiness between devices as the basis to share resources or collaborate on different services such as the share of computational needs. In fact, the relations between IoT devices may reflect their ownership, location, or past collaboration. However, understanding the structure of such complex and ubiquitous networks composed of diverse communicating nodes remains a challenging task. Novel data and graph analysis techniques can constitute an appealing solution to discern the SIoT network patterns and correlations among IoT devices [3].

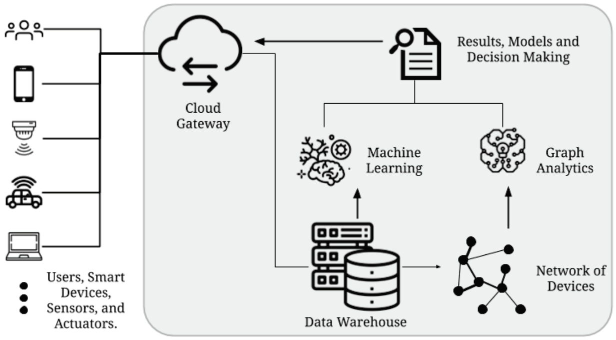

Results, Models and Decision Making
Cloud Gateway
Machine Learning
Graph Analytics
Users, Smart Devices, Sensors, and Actuators.
Network of Devices
Data Warehouse

Fig. 1: Data and network analytic architecture of IoT system.

Machine learning techniques can be used for this purpose to help in designing predictive analytics' techniques for SIoT networks and make well-informed decisions accordingly. For example, data analytics can help process, understand, and enhance the data generated by the devices [4]. It can also help understand the structure of IoT systems using unsupervised machine learning approaches such as classification and clustering methods to group IoT 1) infrastructures, e.g., by clustering devices to reduce the complexity of the vast IoT network or 2) services, e.g., by assigning IoT devices to tasks/services [4]. For example, the study of [5] employed machine learning to identify suspicious network activities by analyzing the transmission paths between the nodes. Another example is presented in [6] where clustering algorithms are used as a first stage to reduce the complexity of a dynamic network of IoT devices.

In this paper, we develop a novel clustering approach based on Graph Neural Network (GNN), a deep learning algorithm, to discern SIoT structure. We aim to embed the features of devices as well as their connections from a real-world dataset using GNN and then apply an unsupervised learning algorithm to determine clusters of IoT devices sharing strong social relations. Results of GNN-based clustering are compared to the deterministic community detection approach, namely the Louvain method [7]. In Fig. 1, we illustrate a general SIoT data analytic framework where graph analysis and machine learning techniques are used to perceive the structure of SIoT system and the relations among its nodes. The IoT devices connect through a cloud gateway to exchange necessary data such as the location and specification of the devices. From this information and other IoT devices' features, graphs modeling the different social relations between the devices can be es-

This work was patially supported by the National Science Foundation (NSF) under the Award no. ECCS-1813949. However, any opinions, findings, and conclusions expressed herein are those of the authors and do not necessarily reflect the views of NSF.

978-1-7281-8058-8/20/$31.00 ©2020 IEEE

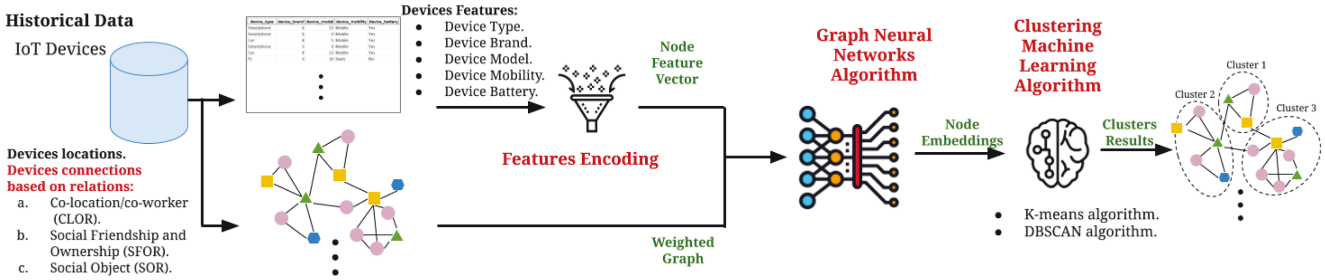


Fig. 2: Graph Neural Network and clustering framework of SIoT devices.

tablished. GNN and unsupervised learning techniques are then employed to determine the clusters of devices sharing strong social relations, which can help better understand the structure of the network and use this extra level of knowledge for more effective service discovery or mobile crowdsourcing tasks.

## II. Proposed GNN-based Clustering Framework

In Fig. 2, we present the different steps of our proposed GNN-based clustering framework for social IoT systems. Starting from a dataset of $N$ IoT devices that includes several features such as Device Type, User ID owner, Device Brand, Device Mobility, Device Battery, and Device Geo-location, a pre-processing step is executed to create multiple weighted graphs of social relations connecting the devices. Afterwards, a GNN algorithm is applied to embed the nodes and their connections with numerical vector representations [8]. The GNN approach is enhanced such as it is capable of handling weighted graphs, i.e., the strength level of the social relations. Finally, an unsupervised clustering algorithm is applied on the vector presentations of the nodes to determine the different clusters of the IoT network.

### A. Social Relations and Data Pre-processing

*1) Devices Relations:* There are different social relations between SIoT devices [2]. These relations are based on the information about the devices such as ownership and geographical locations. In this study, we consider the following three social relations:

- *Co-location/co-work based relation (CLOR):* This relation is inferred from the spatial features of the devices. Therefore, if there is a set of devices within a certain location, there are CLOR relations between these devices. The devices can be stationary or moving to different places. Therefore, mobile devices can dynamically change their CLOR links with other devices.
- *Social object relation (SOR):* The SOR relation is created when two devices collaborate in a continuous or sporadic form. The criteria for setting the links are based on the owners' policies. For example, if two devices are co-located and exchange data for a certain period, then a SOR relationship can be established between them.
- *Social friendship and ownership relation (SFOR):* This relation is based on the social network of the owners and the ownership of the devices. Thus, we create high-weight links between devices that have common owners. The social network of owners can be then used to establish less weighted links among devices based on the number of friends to reach each owner (i.e., ”friend” or ”friend of a friend”) and then project that on the SIoT network.

All the above relations in SIoT can be modeled by undirected and weighted graphs. The nodes are the devices of the IoT system and the edges are the social relations between these devices. The corresponding weights indicate the strength of social relations. The graphs do not include self-looped links on the objects.

*2) Features Encoding:* To ensure that the features of the devices in the dataset are suitable for the machine learner, we encode the categorical attributes such as Device Type, Brand, Mobility, and Battery using a one-hot encoder in Scikit pre-processing. The one-hot encoder transforms nominal data points to integer representation with consideration to limit the natural ordering comparing to the label encoding method. Moreover, the categorical textual values are encoded to integer values that will distinguish the data points, in which many clustering algorithms can handle. For example, in device type, there are a number of classes such as Smartphone, Smart Fitness, Pc, Car, etc. These types will be represented in integer values rather than a string. The resulting feature vector of each device $j$ of size $d \times 1$ is denoted by $\boldsymbol{X}_j$ where $j = 1, \ldots, N$ and $d$ is the number of features per node.

### B. Graph Neural Network Algorithm

With the increase of computational power, many problems are represented by graphs. There is an emergence of adopting neural networks for graph classification, in general. The GNN surpasses that with the ability to handle a graph representation of nodes and edges to classify these nodes [9]. This allows a better representation of the nodes and their relations by jointly embedding their features and their relationships with other IoT devices. The model follows a recursive neighborhood aggregation scheme, where each node aggregates feature vectors of its neighbors to compute its new feature vector. Thus, the node is represented by its transformed feature vector, which captures the structural information within its neighborhood and uses the nodes' different attributes as latent feature representations to enhance the learned representation. Given the weight matrix $\boldsymbol{A}$ of a social relation graph of $n$ nodes, we first normalize it to obtain the matrix $\tilde{\boldsymbol{A}}$ as follows:

$$\tilde{\boldsymbol{A}} = \hat{\boldsymbol{D}}^{-\frac{1}{2}} \hat{\boldsymbol{A}} \hat{\boldsymbol{D}}^{-\frac{1}{2}}, \tag{1}$$

where $\hat{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ and $\hat{\boldsymbol{D}}$ is a diagonal matrix such that $\hat{D}_{ii} = \sum_j \hat{A}_{ij}$ and $\boldsymbol{I}$ is the diagonal matrix. The formula given in (4) adds self-loops to the graph and normalizes each row of the emerging matrix $\boldsymbol{A}$. This normalization addresses numerical instabilities which may lead to exploding/vanishing gradients when used in a deep neural network model. Our GNN model consists of two message passing hidden layers, where the first hidden layer has $h_1 = 64$ units and the second hidden layer has $h_2 = 32$ units such as:

$$\boldsymbol{Z}^t = f(\tilde{\boldsymbol{A}}\boldsymbol{Z}^{t-1}\boldsymbol{W}^t), \tag{2}$$

where $\tilde{\boldsymbol{A}}$ is the normalized weight matrix of the graph given in (1), $\boldsymbol{W}^t$ is a matrix of trainable weights at layer $t$ such as $\boldsymbol{W^1} \in \mathbb{R}^{\boldsymbol{d}\times\boldsymbol{h_1}}$ and $\boldsymbol{W^2} \in \mathbb{R}^{\boldsymbol{h_1}\times\boldsymbol{h_2}}$ , $f$ is the rectified linear activation function (ReLU), and $\boldsymbol{Z}^t$ is the learned embeddings of the graph in the $t^{th}$ layer. As an initialization, $\boldsymbol{Z}^0 = \boldsymbol{X}$ where $\boldsymbol{X} \in \mathbb{R}^{\boldsymbol{n}\times\boldsymbol{d}}$ is a matrix whose $j^{th}$ row contains the feature vector $\boldsymbol{X}_j$ of the IoT node $j$. The two message passing layers are followed by a fully-connected layer which makes use of the softmax function to produce a probability distribution over the class labels.

Generally, the GNN model can be used as node classifier for either supervised or semi-supervised classification. But, given the unlabeled data in our case, we tend to use the GNN model as an embedder where we extract the feature representations of IoT devices in a forward pass using the propagation rule. We then label few nodes of our data and we train the model in a semi-supervised way in order to learn better representations. In fact, the model is trained as a classifier yet we tend to only make use of the nodes' hidden representations $\boldsymbol{Z^2} \in \mathbb{R}^{\boldsymbol{n}\times\boldsymbol{h_2}}$ that are produced from the second message passing layer. We use fixed hyper-parameters for all the graph sizes: learning rate equal to $10^{-2}$ and dropout rate equal to 0.5. We train the model over 100 epochs. Finally, we feed the extracted embeddings to unsupervised machine learning clustering algorithm to determine the communities and discover more clusters.

### C. Clustering Algorithms

Once a vector representation for each node in a social relation graph is determined using GNN, an unsupervised machine learning technique can be utilized to group the IoT devices with common features and attributes into clusters or communities. The similar IoT devices sharing strong social relations will be labeled in a cluster, while devices in different groups will have dissimilar features. In our study, we examine two clustering algorithms, namely the $K$-means and Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithms.

*1) $K$-means:* It is one of the wide used unsupervised clustering algorithms [10]. It achieves the clustering by pre-specifying the number of clusters, $K$. In general, $K$-means iterate to determine $K$ virtual centroids to which it associates the closest data points using the sum of the squared distance separating the vectors. It converges when no further enhancement is achieved. The $K$-means algorithm aims to choose centroids that minimize the inertia or sum-of-squares within-cluster criterion.

*2) DBSCAN:* The main objective of DBSCAN is to identify the clusters based on the density of data points [11], where the most similar data points are dense together and form a distinguished group where the different clusters are separated with less density data points. The algorithm starts with an arbitrary point and converges when all the data points are visited. It uses a distance threshold to decide whether a nearby point belongs to the same cluster or not. If not, it will assigned as a noise, which can be part to another cluster. The previous process will be repeated over all the data points until the density-connected cluster is achieved.

Once the clustering algorithm is run for each SIoT relation, groups of devices sharing strong social relations are determined. The IoT devices may then cooperate together in a trustworthy manner.

## III. Results & Discussions

To examine the proposed framework, we use a data set of real-world IoT devices from a smart city in Santander, Spain, provided by Marche et al. [12]. The data set includes different types of private and public devices. We select 1000, 1500, and 2000 private devices out of 16216 devices to analyze the possibility of the framework for scalability and applicability in different sizes of the IoT system. Following that, the links between the devices are established based on the various social relations, namely CLOR, SFOR, and SOR, described in Section II-A1.

To assess the quality of the different clustering results, we use two of the standard cluster quality metrics in our study: modularity and coverage. 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:

$$\mathrm{Q} = \frac{1}{2m}\sum_{vw}\left(A_{vw} - \frac{k_v k_w}{2m}\right)\delta(c_v c_w), \tag{3}$$

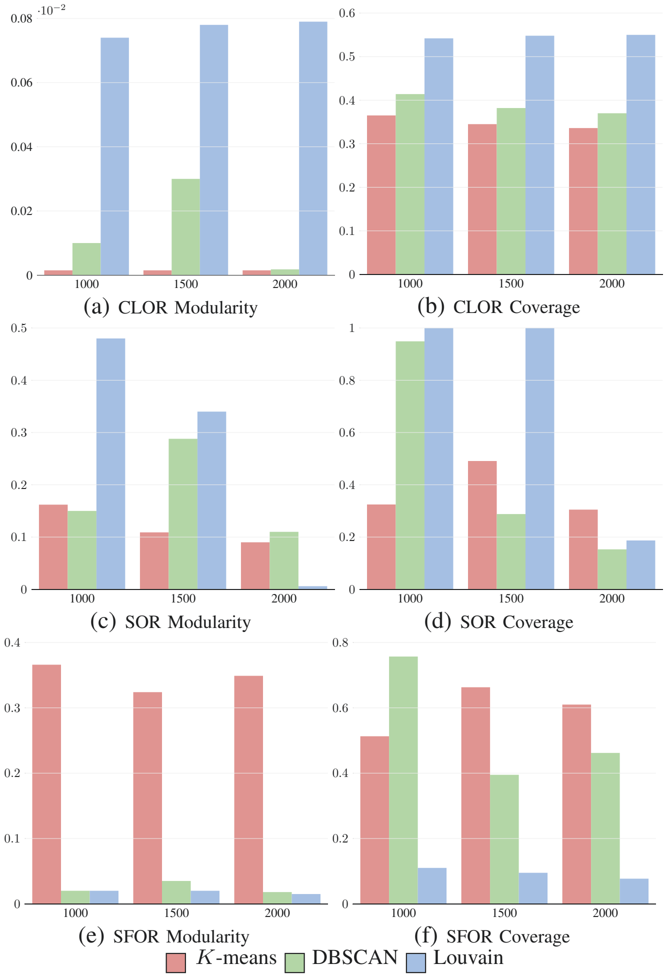
where $\delta$ is the Kronecker delta, it equals to one if $c_u$ and $c_v$ belong to the same community and 0 otherwise, $k_u$ is the degree of node $u$, $m$ is the number of edges in the graph, and $A_{vw}$ is the element located at row $v$ and column $w$ of the adjacency matrix $\boldsymbol{A}$.

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:

$$\mathrm{Cov} = \frac{\sum_{i,j} A_{ij}\delta(S_i, S_j)}{\sum_{i,j} A_{ij}}, \tag{4}$$

where $S_i$ is the cluster to which node $i$ is assigned. Coverage falls in the range 0 to 1, and 1 is the highest score that indicates that a graph topology is well-clustered.

In Fig. 3, we illustrate the clustering results of applying algorithms $K$-means, DBSCAN, and Louvain for the evaluation metrics, modularity, and coverage. We compare the obtained results to one of the deterministic Louvain algorithms. Each sub-figure presents one of three graphs; CLOR, SOR, and SFOR, with three different IoT networks, scale 1000, 1500, and 2000 nodes. The $K$-means is executed using the elbow method to determine the best number of clusters. However, we notice that $K$-means present lower performance when directly applied to the node embeddings of the GNN model, shown in Fig. 3 as red bars. In fact, $K$-means aims to choose centroids that minimize the inertia, which is not a normalized metric.



(a) CLOR Modularity

(b) CLOR Coverage

(c) SOR Modularity

(d) SOR Coverage

(e) SFOR Modularity

(f) SFOR Coverage

Fig. 3: Modularity score and coverage presented for the $K$-means, DBSCAN, and Louvain community detection algorithms with three different social network connectivity scales (1000, 1500, and 2000).

We know that lower values are better, and zero is optimal. But, in very high-dimensional spaces, Euclidean distances tend to become inflated (this is an instance of the so-called “curse of dimensionality”). We run a dimensionality reduction algorithm, i.e., t-distributed Stochastic Neighbor Embedding (T-SNE), before $K$-means clustering, which alleviates this problem, speeds up the computations and presents a visualization method in the 2-dimensional space for the clusters.

With DBSCAN, we notice that some nodes are detected as outliers. Therefore, we assign to each of those nodes a new cluster. We observe that the $K$-means gives well-separated clusters but tends to restrict the number of the groups comparing to DBSCAN. We also notice that both K-means and DBSCAN clustering of the GNN embeddings outperform Louvain community detection mainly for the SFOR network in all scales and the SOR network in large size (2000 nodes). Despite its performance with the CLOR network comparing to the two other methods, the Louvain algorithm tends to restrict the discovered communities to two clusters for the three different scales.

TABLE I: Obtained number of clusters

| | **CLOR** | | | **SFOR** | | | **SOR** | | |
|---|---|---|---|---|---|---|---|---|---|
| No. Devices | $K$-means | DBSCAN | Louvain | $K$-means | DBSCAN | Louvain | $K$-means | DBSCAN | Louvain |
| 1000 | 4 | 19 | 2 | 5 | 47 | 9 | 5 | 50 | 18 |
| 1500 | 7 | 105 | 2 | 11 | 181 | 12 | 7 | 73 | 18 |
| 2000 | 8 | 551 | 2 | 15 | 247 | 14 | 10 | 228 | 25 |

In Table I, where a comparison based on the numbers of clusters obtained for each relation with different networks sizes. The number of clusters for DBSCAN tends to be higher than the other methods for all the networks. This characteristic remains the same even when we do not consider the outliers as separated clusters. Finally, the whole process from the embedding and clustering to the dimension reduction is relatively fast compared to the Louvain method, which is an advantage when applying our approach to a vast network of devices.

## IV. Conclusion

In this study, we proposed a novel GNN-based clustering approach for SIoT devices having different social relations. For a real-world dataset, we embedded the features of the nodes as well as their social relations using GNN and then, fed the obtained vector representations to a conventional clustering algorithm to determine communities of socially connected IoT devices. The process allows fast conversions of complex IoT systems into structured groups of devices that can be exploited to enhance the discovery and object identification for various IoT applications. We notice that different clustering algorithms, case by case, can outperform other community detection methods for certain metrics such as modularity and coverage, which represents a promising result to further examine several machine learners.

## References

[1] A. Al-Fuqaha et al., “Internet of things: A survey on enabling technologies, protocols, and applications,” *IEEE communications surveys & tutorials*, vol. 17, pp. 2347–2376, Jun 2015.

[2] L. Atzori et al., “The social internet of things (SIoT)–when social networks meet the internet of things: Concept, architecture and network characterization,” *Computer networks*, vol. 56, pp. 3594–3608, Nov. 2012.

[3] A. Khanfor et al., “Application of community detection algorithms on social internet-of-things networks,” in *IEEE International Conference on Microelectronics (ICM’19), Cairo, Egypt*, Dec. 2019.

[4] M.S. Mahdavinejad et al., “Machine learning for internet of things data analysis: A survey,” *Digital Communications and Networks*, vol. 4, no. 3, pp. 161–175, Oct. 2018.

[5] X. Liu et al., “Identifying malicious nodes in multihop iot networks using diversity and unsupervised learning,” in *2018 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, May 2018.

[6] X. Shao et al., “Dynamic iot device clustering and energy management with hybrid noma systems,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4622–4630, 2018.

[7] A. Khanfor et al., “Automated service discovery for social internet-of-things systems,” to appear in *IEEE International Symposium on Circuits & Systems (ISCAS’20), Sevilla, Spain.* Oct. 2020. Available at arXiv:2003.11524.

[8] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv preprint arXiv:1609.02907*, 2016.

[9] F. Scarselli et al., “The graph neural network model,” *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2008.

[10] D. Arthur and S. Vassilvitskii, “k-means++: The advantages of careful seeding,” tech. rep., Stanford, 2006.

[11] M. Ester et al., “A density-based algorithm for discovering clusters in large spatial databases with noise.,” in *Kdd*, vol. 96, pp. 226–231, 1996.

[12] Marche et al., “A dataset for performance analysis of the social internet of things,” in *IEEE Intl. Symps. Personal, Indoor Mobile Radio Commun. (PIMRC’18)*, Bologna, Italy, Sept. 2018.