Automated Service Discovery for Social Internet-of-Things Systems

Abdullah Khanfor§, Hakim Ghazzai§, Ye Yang§, Mohammad Rafiqul Haider¶, and Yehia Massoud§

§School of Systems & Enterprises, Stevens Institute of Technology, Hoboken, NJ, USA

¶University of Alabama at Birmingham, AL, USA

Abstract—In this paper, we propose to design an automated service discovery process to allow mobile crowdsourcing task requesters select a small set of devices out of a large-scale Internet-of-things (IoT) network to execute their tasks. To this end, we proceed by dividing the large-scale IoT network into several virtual communities whose members share strong social IoT relations. Two community detection algorithms, namely Louvain and order statistics local method (OSLOM) algorithms, are investigated and applied to a real-world IoT dataset to form non-overlapping and overlapping IoT devices groups. Afterwards, a natural language process (NLP)-based approach is executed to handle crowdsourcing textual requests and accordingly find the list of IoT devices capable of effectively accomplishing the tasks. This is performed by matching the NLP outputs, e.g., type of application, location, required trustworthiness level, with the different detected communities. The proposed approach effectively helps in automating and reducing the service discovery procedure and recruitment process for mobile crowdsourcing applications.

Index Terms—Internet of Things (IoT), community detection, natural language processing, mobile crowdsourcing.

I. INTRODUCTION

The rapid advancement in communication and sensor technologies have significantly raised the number of smart and connected devices. A forecast estimates that 15 smart devices will be owned and maintained by one person with a total of 500 billion devices connected to the Internet by 2030, according to CISCO [1]. Typically, IoT is defined as a set of smart physical objects that are able to sense, report, and interact with the surrounding environment and hence, communicate and cooperate with each other and potentially establish "social" relations. Thus, the concept of social IoT (SIoT) has emerged to model the different relations that IoT objects can have and analyze their impacts on the IoT ecosystem and the application performance. The relationships between the different entities in IoT can be categorized into devices-to-devices, devices-to-people, and people-to-people [2], [3]. Nevertheless, the heterogeneity and complexity of such network entities, structures, and social interactions pose challenges for practitioners to leverage the benefits of such a socially connected networks [4].

One crucial aspect is to understand the different relations between these devices and employ them efficiently for better exploitation of the IoT network omnipresence and diversity [5] [6]. Indeed, individuals with social relationships and mobile computing capabilities are most likely to connect their devices with each other and initiate cooperative applications [7]. Hence, in IoT systems, social relationships is not only limited to human interaction but can also be extended to machine-to-machine (M2M) interactions. According to [8]–[10], there are generally four types of relations that can be established among IoT devices: 1) parental-child relation, 2) co-location/co-work-based relation, 3) object ownership relation, and 4) social object relation. The existence of various social relationships in SIoT opens many research issues and thrust areas. For instance, it is mandatory to enhance the service discovery mechanisms in IoT networks in order to broaden the knowledge of human users about the different potential services. Moreover, when social relations are established, privacy and trustworthiness become important factors that impact the security aspects of the device's interaction. In this context and as an emerging application, mobile crowdsourcing can benefit from this broad SIoT system. In fact, by overlapping the mobile crowdsourcing and SIoT, mobile devices, e.g., smartphones or sensors, can be recruited as workers and be effectively exploited to meet the interests of requesters [11], [12].

In this paper, we design an automated service discovery process for mobile crowdsourcing applications while exploiting the benefits of SIoT infrastructure. The objective is to allow mobile crowdsourcing task requesters, usually human, to find a small set of devices out of a large-scale IoT network to execute their tasks. This will help in automating and reducing the service discovery procedure and recruitment process for many mobile crowdsourcing applications. In such a scenario, the crowdsourcing platform, instead for browsing its full dataset to find potential workers/devices, will extract information from the textual requests of the submitted tasks and accordingly find the list of IoT devices meeting the requirements of the crowdsourcers, e.g., application, location, trustworthiness level, etc. To this end, we divide the large-scale IoT network into several virtual communities whose members share strong social IoT relations. Two community detection algorithms, namely Louvain and OSLOM algorithms, are investigated and applied for this purpose to determine non-overlapping and overlapping IoT devices groups. Then, a NLP-based approach is executed to handle the textual requests and map them with different detected communities. The final list of IoT devices capable of responding the crowdsourcing task will correspond to ones obtained from the intersection of the identified communities. Selected simulation results are applied to a real-world IoT dataset and showcase some examples of the proposed service discovery process.
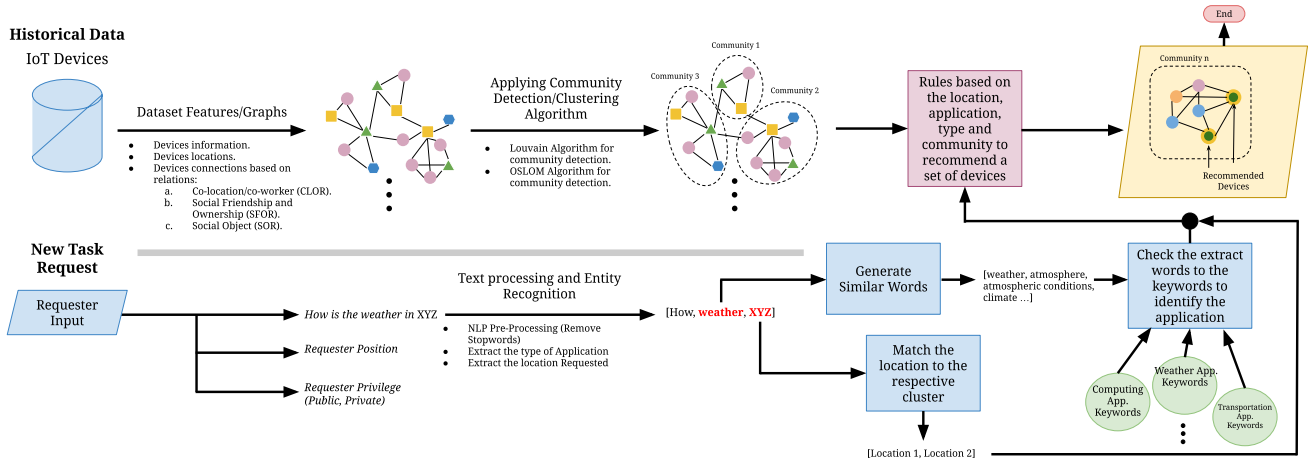
Fig. 1: Proposed service discovery framework to map textual request to the identified communities in SIoT graph.

II. METHODOLOGY

In the proposed framework presented in Fig. 1, we focus on investigating the SIoT relations to help in the service discovery and crowdsourcing of mobile tasks. The framework includes two components: First, we split the IoT network into multiple communities sharing common characteristics using the community detection algorithm. Second, we propose a process to handle a request in natural language, e.g., "What is the humidity level near the beach?". The framework can extract valuable information from the textual request, such as the type of service and the location of the desired information. The requested service is then mapped to the classified communities of different types of relations such as location, friendship, and ownership from the first stage of the framework to determine a small set of IoT devices meeting the requirement of the crowdsourcing task requester.

A. IoT Devices and Social Relationships

We consider a large-scale IoT network located in a wide geographical area. The IoT objects have different types that vary from smartphones, smartwatches, weather sensors, and personal computers devices, etc. In order to build the social IoT network, we introduce the following social relationships:

• *Co-location/co-work based relation (CLOR):* This relationship is established based on the geospatial information of the devices. Hence, we assume that two IoT devices have a CLOR if the geographical distance separating them is less than a certain threshold. In other words, the devices are geographically co-located.

• *Social friendship and ownership relation (SFOR):* This relationship is established by considering the social relationship of the owners of the IoT devices. For example, two devices having the same owner are assumed to have a SFOR relationship. Another example could be the case of two devices owned by two different entities. But, if these owners are having any sort of social relationship (friends, cooperating), then they have some privilege to access their respective devices. Lastly, public devices are assumed to be owned by one entity, e.g., local authorities, that can be establishing a link with any entity in the network.

• *Social object relationship (SOR):* The relationship is established when two IoT objects come into contact when operating. This relationship can be sporadic or continuous based on the needs of the devices or owner policies. For instance, we assume that if two devices have cooperated or communicated together in the past to complete a specific task, then they have a SOR relation. For example, they shared information together or cooperated in data routing, etc. This relationship is built based on the historical operation of the devices and their levels of interactions.

The three relationships presented earlier can be modeled by three different undirected and weighted graphs. The IoT devices constitute the vertices of the graphs while the edges represent the social IoT relationships between them. Some nodes may not be connected to one another for certain social relations. In that case, an edge will not be established. Moreover, the graph have no self-loop edges for the nodes. Finally, the weights on each edge indicate the strength of a relationship. In Section III, we elaborate how social relations are established for real-world IoT system.

B. Community Detection Algorithms

In this section, we investigate community detection algorithms to split the IoT network on different virtual groups based on their social relationship. Flake et al. [13] defined communities as a set of nodes that are connected with more edges compared to the rest of the network. There are different benefits of community detection applications, such as understanding the structure of a graph or improving information retrieval, which is one of our research aims. We apply two community detection algorithms to find non-overlapping and overlapping communities. Moreover, using communities as a method to reduce the search space instead of browsing the whole graph nodes by narrowing the search space to the respective community. As mentioned earlier, the objective is to simplify the service discovery process in large-scale IoT network and thus, the recruitment process in mobile crowdsourcing. To this end, we apply two community detection algorithms to find non-overlapping and overlapping communities.

1) Louvain Algorithm: Louvain method [14] proposed in 2008 to detect non-overlapping, *aka* disjoint, communities in a graph. The technique maximizes the modularity score for each community, where the modularity represents the quality of nodes assignment to communities by examining the density of edges with a set of nodes compared to how it would be connected in a random network. The Louvain method is a greedy optimization and one of the fastest community detection algorithms having a running time of $O(n \log n)$.

2) OSLOM Algorithm: OSLOM is an overlapping clustering algorithm for graphs that conducts a local optimization of a fitness function by considering the statistical significance of clusters with respect to random fluctuations [15]. Even though the worst-case complexity of the algorithm is $O(n^2)$, the method shows a high accuracy in labeling the communities compared to similar methods in artificial network benchmarks.

C. Task Assignment Workflow

Once IoT virtual communities are identified, the mobile crowdsourcing platform can map textual requests to matching IoT groups and find a small set of devices capable of accomplishing the task. To do so, we adopt the process illustrated in Fig. 1 when a new task request is received. The text request is written in a natural language assumed to be fed by a human user. It includes generic information about the task such as type of requested service and task location. Moreover, with the submitted text, the request may include metadata such as the location of the requester and requester's trustworthiness level. The text is processed in the NLP pipeline to remove the stop words such as "a", "an", "in", "the", etc. Next, the words are tokenized to tag it with a part-of-speech (POS) labels. The POS labeling helps the framework extract the information needed to identify the type of application and the location the request needs to process the tasks. For example, the words that are proper nouns may indicate an area name. Other words are selected to generate a set of similar words. The generated words will be measured with the similarity of keywords list for each application such that each application have a set of pre-defined words. Then, we took the highest similarity score between the extracted words from the request and each application keywords. Finally, the framework will be able to locate the necessary information like application type, location, based on the highest similarity score. The new generated words will be compared with a list of pre-defined keywords and/or synonyms corresponding to each application.

In this study, we examine three application types about weather monitoring, transportation, and computation services. The application types can be expanded to other types. As an example, if the input text mentions the words weather, sunny, feel, rain, humidity, etc. then, the desired application will be about the weather, and therefore, it will require sensors with the capability of collecting weather-related data. Then, we extract the required location where the requester needs to check the weather and hence, the platform must identify the corresponding CLOR community. Indeed, using the community detection algorithm, the platform may determine multiple CLOR communities associated to a specific area on the map. Finally, the intersections of application type, CLOR, and SFOR communities will narrow down the list of devices and therefore provide a small set of devices that can process the request.

III. RESULTS & DISCUSSION

To analyze our proposed framework, we use a real-world dataset of IoT devices that are stationed or projected in the city of Santander, Spain [16]. The total number of objects in the dataset is 16216 devices, of which 14600 are labeled as private users devices and 1616 are public devices. The devices vary from smartphones, smartwatches, weather sensors and personal computers devices. For clarity and tractability, we depict a small area where 2157 devices exist. We then run the community detection algorithms to obtain the virtual communities of the IoT network for the CLOR, SFOR, and SOR relations.

For the CLOR, to establish the links, we compute the edge weight W_{ij} connecting to IoT devices i and j by measuring the normalized distance d_{ij} as given in equations (1):

$$W_{ij} = 1 - \frac{d_{ij}}{d_{max}} \in [0, 1], \tag{1}$$

where d_{max} is the longest distance that may separate two nodes in the map of interest. Then, we remove edges with a weight W_{ij} less than the threshold 0.8.

For the SFOR, we generate a random social network utilizing the Watts-Strogatz model [17] since social relationships between owners are not available in the dataset. The model provides an undirected graph for all the nodes existing in the map representing their friendship levels. Hence, for SFOR, we create the edges between the devices based on the owners' friendship such that if one owner owns two devices, then an edge of weight equals to one is created. Afterwards, we connect all the devices owned by two friends with an edge of weight 0.75. Finally, for a friend of a friend, we create an edge of weight equals to $\frac{1}{n_{hops}}$ between the devices where n_{hops} is the number of hops connecting the owners in the social network.

Finally, for SOR, according to Marche et al. [16], assumes that a SOR relationship is established between two objects if the objects have a three-time meeting of a duration not less than 30 minutes in total, and the interval between the two consecutive meetings is 6 hours.

In Fig. 3, we plot the network degree distribution for each social relation to identify the structure of the obtained graphs. The CLOR and SOR graphs exhibit a scale-free network where the number of nodes with a few degrees exceeds the number of nodes with a large number of connections. However, a small world graph is obtained for the SFOR relation since the network connections between the owners are based on the six degrees of the separation concept.

In Fig. 2, we present the communities founded by Louvain and OSLOM algorithms. We group the communities having fewer than four devices under the "Others" label. For overlapping communities, the OSLOM algorithm can detect
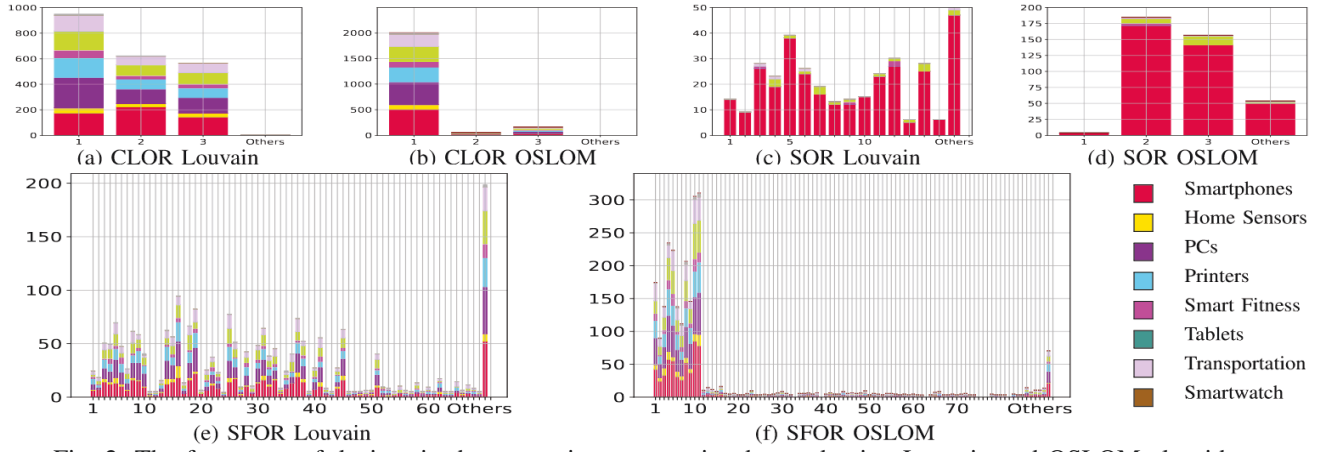
Fig. 2: The frequency of devices in the respective community detected using Louvain and OSLOM algorithms.
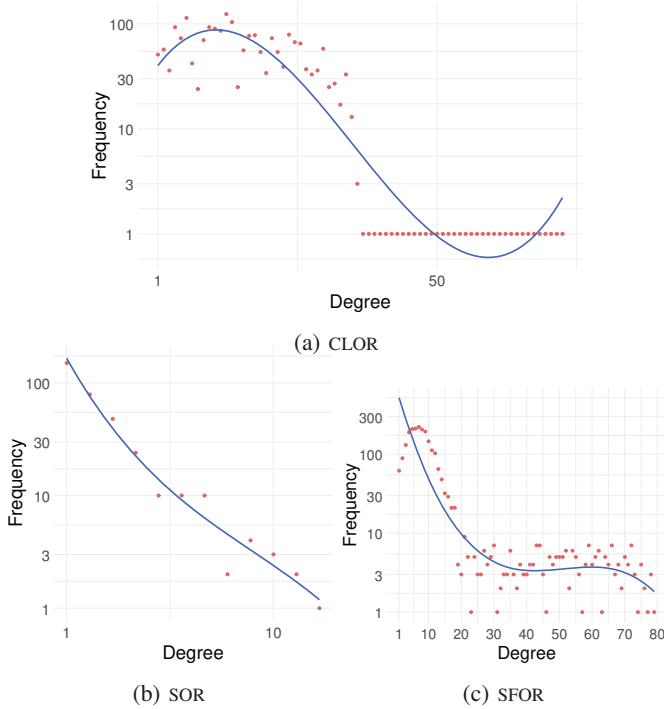

Fig. 3: Degree distribution of graphs associated to each social IoT relations.

Fig. 4: Framework system flow from the requester perspective.

few but vast communities. However, the Louvain algorithm spreads the devices into communities having almost the same frequency of devices. For instance, with CLOR, the OSLM finds three communities, one of them with around 2000 devices while the rest have less than 300 devices. Louvain has three large communities having between 550 and 950 devices. Moreover, the average size of each community is larger in the CLOR relation since we are targeting a small area of the map. Hence, the devices are originally co-located. We also notice that the communities in SOR are only PCs, smartphones, and tablets; since it is based on a simulation that only established relationships among these devices. In summary, the Louvain algorithm provides reasonable virtual community distribution than OSLOM, which can be used to address mobile crowdsourcing requests.
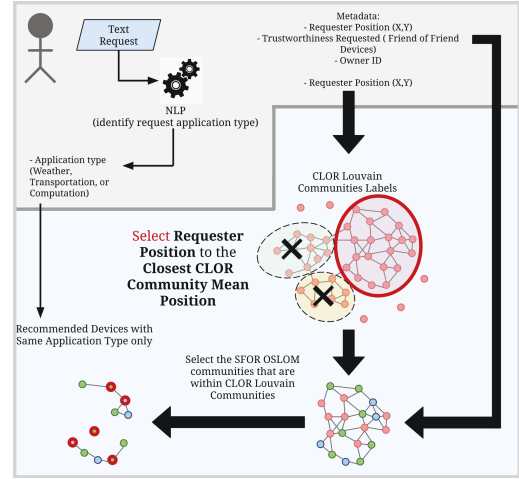
We illustrate the workflow of the framework from the requester perspective in Fig. 4. The gray area can be done on the client-side of the requester, such as smartphones or PCs; the blue part can be deployed on the server-side. The metadata from the requester will be used to recognize the position of the suitable CLOR community. The CLOR community will be labeled by the mean position of the devices of the community. Then, the framework will select the closest CLOR community to the requester position. Next, the SFOR communities within the CLOR will be selected based on the Friendship level requested. Finally, the devices will be recommended after comparing the request application type and the subsets of the CLOR Louvain community and SFOR OSLOM communities.

IV. CONCLUSION

In this study, we proposed an automated framework to handle mobile crowdsourcing requests within a large-scale SIoT network. We first split the IoT network into multiple virtual communities grouping IoT devices sharing common characteristics modeled using the social relations. Two community detection algorithms with non-overlapping and overlapping communities are investigated. Then, with a NLP approach, we successfully capture the information from the textual request to match the with communities to find a small list of devices that would benefit the crowdsourcing process.

REFERENCES

[1] CISCO, "Internet of things at a glance - cisco," 2016.

[2] L. Tan and N. Wang, "Future internet: The internet of things," in *IEEE International Conference on Advanced Computer Theory and Engineering (ICACTE'10)*, IEEE, Chengdu, China, Aug. 2010.

[3] K. Aono, N. Lajnef, F. Faridazar, and S. Chakrabartty, "Infrastructural health monitoring using self-powered internet-of-things," in *IEEE International Symposium on Circuits and Systems (ISCAS'16)*, pp. 2058–2061, IEEE, 2016.

[4] A. Bader, H. Ghazzai, A. Kadri, and M.-S. Alouini, "Front-end intelligence for large-scale application-oriented internet-of-things," *IEEE Access*, vol. 4, pp. 3257–3272, 2016.

[5] D. Hovhannisyan, A. Eltawil, and F. Kurdahi, "Testing topology adaptive irrigation iot with circuits," in *IEEE International Symposium on Circuits and Systems (ISCAS'19)*, IEEE, Sapporo, Hokkaido, Japan, May 2019.

[6] A. Khanfor, H. Ghazzai, Y. Yang, and Y. Massoud, "Application of community detection algorithms on social internet-of-things networks," in *IEEE International Conference on Microelectronics (ICM'19)*, IEEE, Cairo, Egypt, Dec. 2019.

[7] 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," *Computer networks*, vol. 56, no. 16, pp. 3594–3608, 2012.

[8] M. Nitti, L. Atzori, and I. P. Cvijikj, "Network navigability in the social internet of things," in *IEEE world forum on internet of things (WF-IoT'14)*, pp. 405–410, IEEE, Seoul, Korea, Mar. 2014.

[9] B. Afzal, M. Umair, G. A. Shah, and E. Ahmed, "Enabling iot platforms for social iot applications: vision, feature mapping, and challenges," *Future Generation Computer Systems*, vol. 92, pp. 718–731, 2019.

[10] M. Roopa, S. Pattar, R. Buyya, K. R. Venugopal, S. Iyengar, and L. Patnaik, "Social internet of things (siot): Foundations, thrust areas, systematic review and future directions," *Computer Communications*, 2019.

[11] X. Wan, H. Ghazzai, and Y. Massoud, "Mobile crowdsourcing for intelligent transportation systems: Real-time navigation in urban areas," *IEEE Access*, vol. 7, pp. 136995–137009, 2019.

[12] A. Hamrouni, H. Ghazzai, M. Frikha, and Y. Massoud,, "A spatial mobile crowdsourcing framework for event reporting," *to appear in IEEE Transactions on Computational Social Systems*.

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

[14] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

[15] A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato, "Finding statistically significant communities in networks," *PloS one*, vol. 6, no. 4, p. e18961, 2011.

[16] C. Marche, L. Atzori, and M. Nitti, "A dataset for performance analysis of the social internet of things," in *IEEE Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'18)*, Bologna, Italy, Sept. 2018.

[17] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-worldnetworks," *nature*, vol. 393, no. 6684, p. 440, 1998.