

SIoTPredict: A Framework for Predicting Relationships in the Social Internet of Things

Abdulwahab Aljubairy¹, Wei Emma Zhang², Quan Z. Sheng¹, and
Ahoud Alhazmi¹

¹ Department of Computing, Macquarie University, NSW 2109, Australia
{abdulwahab.aljubairy, ahoud.alhazmi}@hdr.mq.edu.au
michael.sheng@mq.edu.au

² School of Computer Science, The University of Adelaide, SA 5001, Australia
wei.e.zhang@adelaide.edu.au

Abstract. The Social Internet of Things (SIoT) is a new paradigm that integrates social network concepts with the Internet of Things (IoT). It boosts the discovery, selection and composition of services and information provided by distributed objects. In SIoT, searching for services is based on the utilization of the social structure resulted from the formed relationships. However, current approaches lack modelling and effective analysis of SIoT. In this work, we address this problem and specifically focus on modelling the SIoT's evolvement. As the growing number of IoT objects with heterogeneous attributes join the social network, there is an urgent need for identifying the mechanisms by which SIoT structures evolve. We model the SIoT over time and address the suitability of traditional analytical procedures to predict future relationships (links) in the dynamic and heterogeneous SIoT. Specifically, we propose a framework, namely SIoTPredict, which includes three stages: i) collection of raw movement data of IoT devices, ii) generating temporal sequence networks of the SIoT, and iii) predicting relationships among IoT devices which are likely to occur. We have conducted extensive experimental studies to evaluate the proposed framework using real SIoT datasets and the results show the better performance of our framework.

Keywords: Social Internet of Things (SIoT) · Link prediction · Dynamic Networks · Edge Exchangeability.

1 Introduction

Crawling the Internet of Things (IoT) to discover services and information in a trusted-oriented way remains a prolonged challenge [21]. Many solutions have been introduced to overcome the challenge. However, due to the increasing number of IoT objects in a tremendous rate, these solutions do not scale up. Integrating social networking features into the Internet of Things (IoT) paradigm has received an unprecedented amount of attention for the purpose of overcoming issues related to IoT. There have been many attempts to integrate IoT devices in social loops such as Smart-Its friend procedure [8], Blog-jects [3], Things that

Twitter [9], and Ericson Project³. A new paradigm has emerged from this, called *Social Internet of Things* (SIoT), and the key idea of this paradigm is to allow IoT objects to establish relationships with each other independently with respect to the heuristics set by the owners of these objects [1, 2, 15, 18]. The perspective of SIoT is to incorporate the social behaviour of intelligent IoT objects and allow them to have their own social networks autonomously.

There are several benefits to the SIoT paradigm. First, SIoT can foster resource availability and enhance services discovery easily in a distributed manner using friends and friends of friends [1], unlike traditional IoT where search engines are employed to find services in a centralized way. Second, the centralized manner of searching IoT objects raises scalability issue, and SIoT overcomes the issue because each IoT object can navigate the network structure of SIoT to reach other objects in a distributed way [2, 20, 21]. Third, based on the social structure established among IoT objects, things can inquire local neighbourhood for other objects to assess the reputation of these objects. Fourth, SIoT enables objects to start new acquaintance where they can exchange information and experience.

Many research efforts have been devoted to realizing the SIoT paradigm. However, the majority of the research activities focused on identifying possible policies, methods and techniques for establishing relationships between smart devices autonomously and without any human intervention [1, 2]. In addition, several SIoT architectures have been proposed [2, 5, 17]. In spite of the intensive research attempts on SIoT, there are insufficient considerations to model and analyze the resulted SIoT networks. The nature of SIoT is dynamic because it can grow and change quickly over time where nodes (IoT objects) and edges (relationships) appear or disappear. Therefore, there is a growing interest in developing models that allow studying and understanding this evolving network, in particular, predicting the establishment of future links (relationships) [1, 15]. Predicting future relationships among IoT objects can be utilized for several applications such as service recommendation and service discovery. Thus, there is a need for identifying the mechanisms by which SIoT structures evolve. This is a fundamental research question that has not been addressed in SIoT yet, and it forms the motivation for this work.

However, the size and complexity of the SIoT network create a number of technical challenges. Firstly, the nature of the resulted network structure is *dynamic* because smart devices can appear and disappear overtime and the existed relationships may vanish and new relationships may establish. Secondly, SIoT is naturally structured as a *heterogeneous* graph with different types of entities and various relationships [2, 18]. Finally, the size of SIoT network is massive, and hence, it requires efficient and scalable methods. Therefore, this paper focuses on modelling the SIoT network and study, in particular, the problem of predicting future relationships among IoT objects. We study the possibility of relationship establishment among IoT objects when there is co-occurrence meeting in time and space. Our research question centers on how likely two IoT objects could

³ <https://www.ericsson.com/en/blog/2012/4/a-social-web-of-things>

create a relationship between each other when they have been approximately on the same geographical location at the same time on multiple occasions. In our work, we develop the SIoTPredict framework, which includes three stages: i) collecting the raw movement data of IoT devices, ii) generating temporal sequence networks of SIoT, and iii) predicting future relationships that may be established among things. The salient contributions of our study are summarized as follows:

- **Designing and implementing the SIoTPredict framework for studying the SIoT network.** The SIoTPredict framework consists of three main stages for i) collecting raw movement data of IoT devices, ii) generating temporal sequence networks, and iii) predicting future relationships among things. To the best of our knowledge, our framework is the first on SIoT relationship prediction.
- **Generating temporal sequence networks of SIoT.** We develop two novel algorithms in the second stage of our framework. The first algorithm identifies the stays of IoT objects and extracts the corresponding locations. The second algorithm, named Sweep Line Time Overlap, discovers when and where any two IoT objects have met.
- **Developing a Bayesian nonparametric prediction model.** We adopt the Bayesian nonparametric learning to build our prediction model. This model can adapt the new incoming observations due to the power representation and flexibility of Bayesian nonparametric learning.
- **Conducting comprehensive experiments to assess our framework.** SIoTPredict has been evaluated by extensive experiments using real-world SIoT datasets [12]. The results demonstrate that our framework outperforms the existing methods.

The rest of this paper is organized as follows. Section 2 discusses the related works. Section 3 presents heterogeneous graph modeling for Social IoT and introduces the SIoTPredict framework. The experimental results on real SIoT datasets are presented in Section 4, and finally Section 5 concludes the paper.

2 Related Work

SIoT is still in the infancy stage, and several efforts have been devoted to realizing the SIoT paradigm. Most of the current research activities focused on identifying possible policies, methods and techniques for establishing relationships between smart devices autonomously and without any human intervention [2]. Atzori et al. [2] proposed several relationships that can be established between IoT objects as shown in Table 1. Some of these relationships are static such as POR and OOR, which can usually be defined in advance. Other relationships are dynamic and can be established when the conditions of the relationship are met. Roopa et al. [18] defined more relationships that may be established among IoT objects.

Nevertheless, current SIoT research lacks effective modelling and analysis of SIoT networks. However, in the context of IoT, there are a few attempts to exploiting the relationships among smart devices and users for recommending

things to users. Yao et al. [23] proposed a hyper-graph based on users' social networks. They used existing relationships among users and their things to infer relationships among IoT objects. They leveraged this resulted network for recommending things of interest to users. Mashal et al. [14] modelled the relationships among users, objects, and services as a tripartite graph with hyper-edges between them. Then they explored existing recommendation algorithms to recommend third-party services. Nevertheless, these works are mainly based on users' existing relationships and the things they own. Atzori et al. [1] emphasized to modelling and analyzing the resulted social graphs (uncorrelated to human social networks) among smart objects in order to introduce proper network analysis algorithms. Therefore, our work aims to model the SIoT network in order to allow studying relationships prediction (link prediction) among IoT objects that may form in the future.

Link prediction is considered as one of the most essential problems that have received much attention in network analysis, and in particular, when anticipating the network structure at a future time. A large body of work has investigated link prediction with various aspects including similarity-based measures, algorithmic methods, and probabilistic and statistical methods [11, 13]. Recently, there is a growing interest in developing probabilistic network models using Bayesian nonparametric learning. Bayesian nonparametric is capable to capture the network evolution in different time steps by finding latent structure in observed data. Latent class models such as Stochastic Blockmodels (SBs) [7] and Mixed Membership Stochastic Blockmodels (MMSB) [19] depend on the vertex-exchangeability perspective where nodes are the target unit to assign into clusters. However, these models suffer from generating dense networks while most of the real-world networks tend to be sparse. To overcome this limitation, edge-exchangeable models have been proposed to deal with sparse networks [4, 22]. In this perspective, edges are the main units to assign into clusters.

Table 1: Some main SIoT relationships [2].

SIoT Relationships	Description	Static/ Dynamic
Parental object relationship (POR)	This relationship is established among same generation objects which are made by the same manufacturer.	Static
Co-location object relationship (CLOR)	This relationship is established among objects that are always existed in the same place.	Dynamic
Co-work object relationship (CWOR)	This relationship is established among objects when they collaborate to achieve a common goal or provide a particular IoT application.	Dynamic
Ownership object relationship (OOR)	This relationship is established among objects which belong to the same user.	Static
Social object relationship (SOR)	This relationship is established among objects when they come into contact, sporadically or continuously, because they or their owners come in touch with each other during daily routine.	Dynamic

3 Bayesian Non-parametric Approach

In this section, we describe the dynamic heterogeneous SIoT graph modelling and then present the details of our SIoTPredict framework.

3.1 Dynamic Heterogeneous SIoT Graph Modelling

A dynamic, heterogeneous SIoT graph is composed of nodes and edges where nodes represent IoT devices, and edges represent relationships that could be of multiple different types. The formal definition is as follows.

An SIoT network can be considered as a temporal sequence of networks (as depicted in Figure 1a) $G = \{G^{t_1}, G^{t_2}, \dots, G^{t-n}\}$ where G^t is a snapshot of the network at time t and $t \in \mathbb{Z}^+$. Each snapshot of the network contains a set of nodes and edges $G^t = \{V^t, E^t, X^t\}$ where the set of edges $E^t = \{e_1^t, \dots, e_n^t\}$ contains N edges observed at time t and the set of vertices V^t is the set of vertices that have at least participated in one edge up to t such that $V^{t-1} \subseteq V^t$. An edge $e_n^t = (v_i^t, v_j^t)$ is a tuple of two interacting nodes. $X^t = [x_1^t, x_2^t, \dots, x_i^t]$ represents the feature matrix at time t where x_i is the attribute vector of node v_n .

Throughout the paper, we consider the dynamic relationship establishment using SIoT data as a case study. Figure 1b shows an example of the SIoT heterogeneous network. For this application, we assume that a heterogeneous SIoT graph has been obtained at time t from the SIoT data. Given these data, we will predict the likelihood of a relationship (edge) creation between any two IoT devices (nodes).

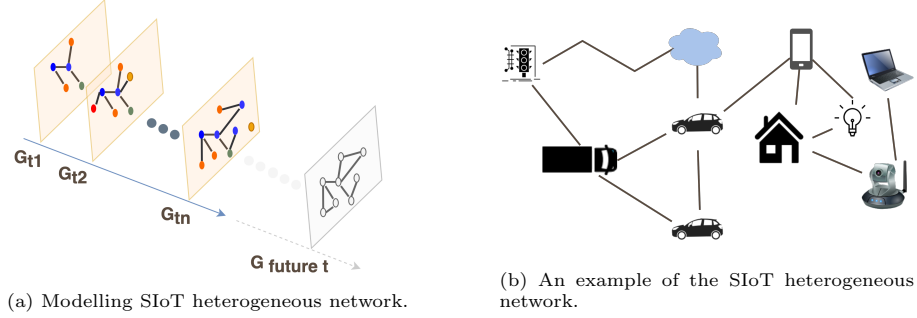


Fig. 1: The SIoT heterogeneous network.

3.2 The Proposed SIoTPredict Framework

This section explains our SIoTPredict framework for predicting future relationships in SIoT. Figure 2 gives an overview of the SIoTPredict framework. The framework includes three stages namely: *Stage 1*: collection of the raw movement data of IoT devices, *Stage 2*: generating the temporal sequence networks of SIoT, and *Stage 3*: prediction future relationships of the SIoT. In the following, we will provide more details on these three stages.

Stage 1. Collection of the Raw Movement Data of IoT Devices. In the first stage of our framework, we collect the raw movement data of IoT devices. We distinguish two types of IoT devices: *mobile* and *static*. The coordinates of a

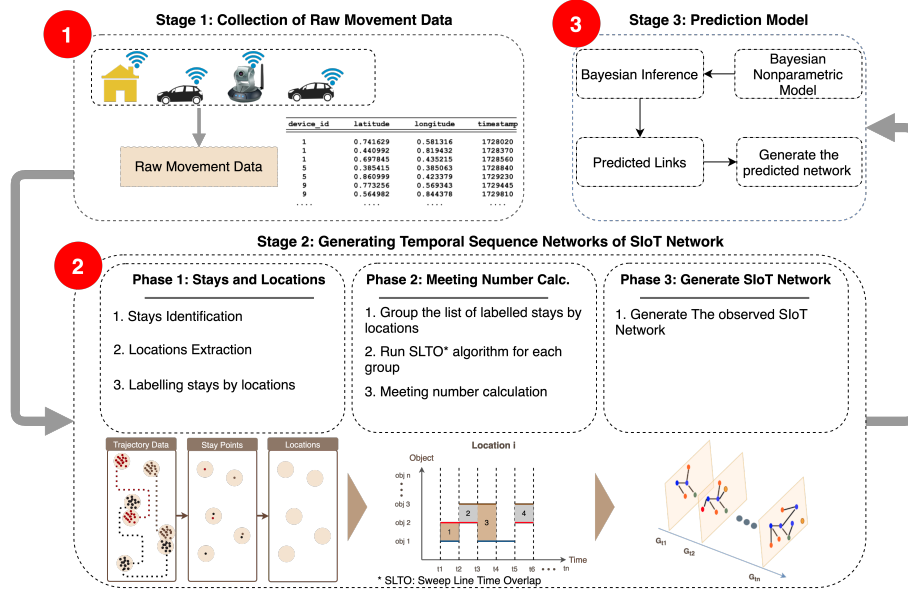


Fig. 2: An overview of our SIOTPredict framework.

static device (e.g., a light pole) are stationary and known whereas the coordinates of a mobile device (e.g., a bus) is dynamic and changing while the device is moving. We assume that mobile devices include GPS technology which provides the location coordinates of these devices along with the timestamp. We also assume mobile IoT devices send their location history records continuously (e.g., every 60 seconds). Each record contains some important fields: (*device_id*, *latitude*, *longitude*, and *timestamp*).

Definition 1. A **location history record** is represented by a point on the earth (*latitude*, *longitude*) and a *timestamp*. This record tells where an IoT object is at a specific time (as illustrated in Figure 3).

Definition 2. A **trajectory** is a sequence of location history records of an IoT object (as illustrated in Figure 3).

Stage 2. Generating Temporal Sequence Networks of SIOt. In the second stage, we generate temporal sequence networks of SIOt based on the observed raw movement data of IoT devices from *Stage 1*. Generating the temporal sequence networks of SIOt is performed in three phases as follows: *Phase 1*: identifying stays from raw movement data, extracting locations, and labelling each stay by the extracted locations, *Phase 2*: calculating the number of meetings among IoT devices using the Sweep Line Time Overlap (SLTO) algorithm, and *Phase 3*: generating temporal networks of SIOt.

Phase 1) Identifying “stays” from the raw movement data and extracting “locations”. We are interested in knowing where objects meet. Therefore, we first

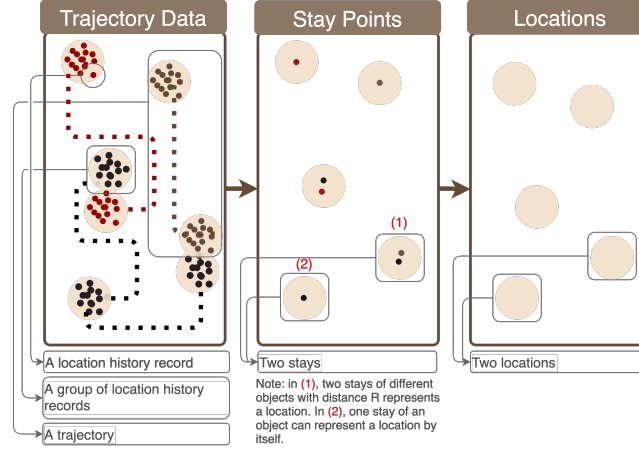


Fig. 3: Illustration on IoT object location, stay point, and movement trajectory need to identify the stays for all objects using their raw movement data. Then, we extract locations from these stays. This enables us to identify where and when IoT objects have stayed.

Definition 3. A **stay** is a sequence of N location history records, which can be represented by (**longitude**, **latitude**, **Start-Time**, **End-Time**). Longitude and latitude represent the average of longitude and latitude values in the sequence. Start-Time indicates the smallest timestamp in the sequence, and End-Time represents the largest timestamp (see Figure 3).

Definition 4. A **location** can be the latitude and longitude of one stay, or it can be the average of longitude and latitude of a group of stays. This group of stays are separated by less than or equal to a distance R (see Figure 3).

We develop Algorithm 1 for identifying stays, extracting locations and, then labelling identified stays by the extracted locations. The *input* of this algorithm is the raw movement data of IoT objects. The *output* is a list of identified stays labelled by extracted locations. The time complexity of this algorithm is quadratic since it is required to calculate the distances between each two observations in the raw movement data.

The first step focuses on stay identification (line 6-27). This step is to identify the stays for each IoT object from the given raw movement data. First, we define the time period of the stay (for example, when the value of stay period = 10, it means that we need to identify if an object stays at a place for 10 minutes). According to our assumption, each record in the raw movement data is sent every one minute, so 10 records represent 10 minutes. The algorithm calculates the distance among the raw movement data records according to Eq 1, where, d is the distance between the two location history records, r is the radius of the sphere (earth), θ_1, θ_2 are the latitude of the two location history records, λ_1, λ_2 are longitude of the two location history records. Then, it groups them if

their distances are less than or equal to a threshold R . From line (18-27), the algorithm checks each group. If a group has a number of records larger than or equal to the value of the stay period, the algorithm takes the average of the latitude and longitude of the group. It also takes the smallest timestamp of the group to be the starting time of the stay, and the largest timestamp to be the ending time of the stay.

$$distance = 2r.sin^{-1}\left(\sqrt{\sin^2\left(\frac{(\theta_2 - \theta_1)}{2}\right) + \cos(\theta_1).\cos(\theta_2).\sin^2\left(\frac{(\lambda_2 - \lambda_1)}{2}\right)}\right) \quad (1)$$

The second step targets location extraction (line 28-40). The algorithm extracts the list of locations out of the identified stays. Since stays are represented by latitude and longitude, the algorithm calculates the distance among these stays, and groups them using a threshold R . The algorithm takes the average of the latitude and longitude of these stays to represent one location. If there is a stay which has not been grouped with any other stays, this stay can represent a location. Finally the third step focuses on labelling stays with locations. In this step, we label the identified stays by one of the extracted locations.

Phase 2) Calculating the number of meetings between objects using Sweep Line Time Overlap Algorithm. We develop Algorithm 2 to detect and report all overlapped periods occurred among the given set of stays produced by Algorithm 1. The purpose is to determine if any two IoT objects have met in a location at a particular time. This novel algorithm named as Sweep Line Time Overlap (SLTO). The SLTO algorithm is inspired by a sweep-line algorithm in geometry which finds intersections between a group of line segments. However, the SLTO algorithm identifies if there are overlapping periods among stays of the objects in a location. The idea of this algorithm is to run a virtual sweep-line parallel to y-axis and move from left to right in order to scan intervals on x-axis. When this sweep-line detects overlaps among stays (they look like line segments which represent the stay periods of objects), it starts calculating if there is an overlap and reporting these overlaps. There are two main steps.

The first step focuses on storing all the intervals of the stays (line 2-3). Since our goal is to find the overlapped periods among the set of stays, the algorithm initializes two data structures: i) a priority queue Q to store all the intervals of the stays we got from Algorithm 1 in sorting order, and ii) a sweep-line status as S to scan the stays from left to right.

The second step runs the sweep line S (line 4-10). We get the interval end points from Q one by one to allow the sweep line S to scan it. The sweep line detects the start of the stay in the space, and adds it to the S . Also, when it detects the end of the stay (in this case, the SLTO finished from scanning the stay), the algorithm checks the last element in S . If it is the start of this stay, then the algorithm removes the stay from S with no action. If the last element in S is not the start of the finished stay, then an overlap or more have been detected between the this stay and other active stays in the S . Algorithm 3 reports the

Algorithm 1: Stay identification and location extraction.

Input : The raw movement data of IoT objects
Output: Stays Labelled with extracted locations

```

1 data = raw movement data;
2 StayPeriod = N;
3 ASSIGN EMPTY LIST to Stay;
4 ASSIGN EMPTY LIST to IdentifiedStays;
5 ASSIGN EMPTY LIST to Locations;
6 for  $i \leftarrow 1$  to  $data.length - 1$  do
7   Stay.append( $data[i]$ );
8   for  $j \leftarrow i + 1$  to  $data.length$  do
9      $distance \leftarrow distance(data[i], data[j])$  ; ▷ Eq1
10    if ( $distance \leq R$ ) then
11      Stay[i].append( $data[j]$ );
12    else
13       $i \leftarrow j$ ;
14      Break;
15    end
16  end
17 end
18 for  $i \leftarrow 1$  to  $Stay.length$  do
19   if  $Stay[i].length \geq StayPeriod$  then
20     objID  $\leftarrow$  Stay[i].objID;
21     lat  $\leftarrow$  Compute avg lat of the stay;
22     long  $\leftarrow$  Compute avg long of the stay;
23     Start-Time  $\leftarrow$  min(Stay[i].timestamp);
24     End-Time  $\leftarrow$  max(Stay[i].timestamp);
25     IdentifiedStays.append([objID,lat,long,Start-Time,End-Time]);
26   end
27 end
28 for  $i \leftarrow 1$  to  $IdentifiedStays.length - 1$  do
29   coords=[ ];
30   coords.append([IdentifiedStays[i].lat,IdentifiedStays[i].long]);
31   for  $j \leftarrow i + 1$  to  $IdentifiedStays.length$  do
32      $distance \leftarrow distance(IdentifiedStays[i], IdentifiedStays[j])$  ; ▷ Eq1
33     if ( $distance \leq R$ ) then
34       coords.append([IdentifiedStays[j].lat,IdentifiedStays[j].long]);
35     end
36   end
37   lat  $\leftarrow$  Compute avg lat of coords;
38   long  $\leftarrow$  Compute avg long coords;
39   Locations.append([lat,long]);
40 end

```

overlaps discovered by SLTO. It calculates the amount of the detected overlaps, and the results are reported. The time complexity of SLTO is $O(N \log N + L)$ since the stay time is only calculated when overlaps between objects exist.

Algorithm 2: SLTO to find the time overlaps among the stays of objects.

Input : A list of stays
Output: A list of pair of objects with the overlapped period among them

```

1 begin
2   Q = The list of stays from Algorithm 1
    as [[stay endpoint value, status,
    interval, objID]];
3   S = Initialize the Sweep line for storing
    detected stays;
4   while Q is not empty do
5     currentEvent  $\leftarrow$  Q.get();
6     if (currentEvent.Status == Start)
7       then
8         - Insert currentEvent into S;
9       else
10        - CheckOverlap(S ,
            currentEvent);
        - Delete currentEvent from S;

```

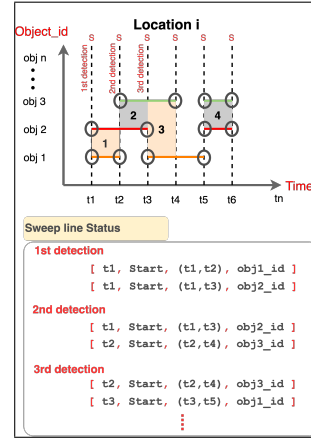


Fig. 4: The sweep line *S* moves from left to right to scan stay intervals. The 1st detection is occurred at time *t1*.

Algorithm 3: CheckOverlap

Input : *S*, *currentEvent*
Output: Report the overlapping among stays

```

1 begin
2   L2 = currentEvent
3   for (L1 in S) do
4     if L1.start < L2.end & (L2.start < L1.end) then
5       OverlapPeriod  $\leftarrow$  min(L1.end, L2.end) - max(L1.start, L2.start)
6       - Report the overlap (ids of objects and OverlapPeriod)

```

Phase 3) Generating the temporal networks of SIoT. After obtaining the time overlapping periods of stays among IoT objects, we are able to know the count of meetings occurred between any two IoT objects. According to this, we check the rules of the targeted relationship such as how many times they have met and the length of the interval period. If the rules are met, then we build the temporal sequence of the SIoT composed from this relationship to be used in our prediction model stage.

Stage 3. Predicting Future Relationships of the SIoT Network. After generating the temporal sequence networks in the second stage, our next step is to model each one of them using the Bayesian non-parametric model [22]. This model allows combining structure elucidation with a predictive performance by clustering links (edges) rather than the nodes. Our aim here is to predict links (relationships) between IoT objects that are likely to occur in the subsequent snapshot of the network. Therefore, the SIoT network is modelled as an exchangeable sequence of observed links (relationships) and that allows adapting the growth of the network over time. We assume that the SIoT network clusters into groups, and for this, we model each community using a mixture of Dirichlet network distributions. The description of the model as follows:

$$\begin{aligned} D &:= (dk, k \in N) \sim GEM(\alpha) \\ G &:= \sum_{i=1}^{\infty} \pi_i \delta_{\Theta_i} \sim DP(\gamma, \theta) \\ c_n &\sim D \\ u_n, v_n &\sim G \end{aligned} \tag{2}$$

We model the relationships of the SIoT network using Dirichlet distribution G , where δ_{Θ_i} is a delta function centered on Θ_i , and π_i is the corresponding probability of an edge to exist at Θ_i , with $\sum_{i=1}^{\infty} \pi_i = 1$. The parameter γ controls the total number of nodes in the network. To model the size and number of clusters, we use a stick-breaking distribution $GEM(\alpha)$ with concentration parameter α that controls the number of the clusters. The model places a distribution over all clusters, and it places per-cluster distribution over the nodes. To generate an edge, first, a cluster will be picked according to D . Then, two nodes (devices) will be sampled according to G . The probability of predicting a link between any two objects is proportional to the product of the degree of these two devices.

$$P(\theta|X, model) = \frac{P(X|\theta, model)P(\theta|model)}{\int P(\theta, X|model) d\theta} \tag{3}$$

For the inference part that is based on the Bayes' rule (Eq. 3), we follow the same steps conducted in [22] to compute the distribution over the cluster assignment using the Chinese restaurant process and evaluate the predictive distribution over the n^{th} link, given the previous $n-1$ links. We perform inference using an Markov chain Monte Carlo (MCMC) scheme [22].

4 Experiments

We evaluated the effectiveness and efficiency of the SIoTPredict framework based on comprehensive experiments. In this section, we discuss the experimental design and report the results.

4.1 Dataset

We used the SIoT datasets⁴ to evaluate the SIoTPredict framework. These datasets are based on real IoT objects available in the city of Santander and contain a description of IoT objects. Each object is represented by fields such

⁴ <http://www.social-iot.org/index.php?p=downloads>

as (`device_id`, `id_user`, `device_type`, `device_brand`, `device_model`). The total number of IoT objects is 16,216. 14,600 objects are from private users and 1,616 are from public services. The dataset includes the raw movement data of devices that are owned by users and the smart city. There are two kinds of devices: static devices and mobile devices. Static devices are represented by fixed latitudes and longitudes. Mobile devices are represented by latitudes, longitudes, and timestamps. The latitude and longitude values of mobile devices are dynamic. In addition, the dataset includes an adjacency matrix for SIoT relationship produced with some defined parameters. In Table 2, we only depict SOR and SOR2 relationships and their parameters to be used in our experiments.

Table 2: Parameter setting for SOR and SOR2 relationships.

Relationship	Potential Devices	Parameters
SOR	Private mobile devices	This relationship is based on three parameters: <ol style="list-style-type: none"> 1. Number of meetings= 3 2. Meeting duration= 30 minutes 3. Interval between two consecutive meetings= 6 hours
SOR2	Public devices (static and mobile) and mobile private devices	This relationship is based on three parameters: <ol style="list-style-type: none"> 1. Number of meetings= 3 2. Meeting duration= 1 minute 3. Interval between two consecutive meetings= 1 hour

4.2 Performance Metrics and Comparison Methods

In this section, we explain the common metrics and the comparison methods.

Performance Metrics. Our performance metrics used in the experiments include Accuracy, Precision, Recall, and $F1$ score. Following the work of information diffusion in [6], we define the Accuracy as the ratio of correctly predicted edges to the total edges in the true network, Precision as the fraction of edges in the predicted network that are also present in the true network, Recall as the fraction of edges of the true network that are also presented in the predicted network, and finally $F1$ score as the weight average of precision and recall.

Comparison Methods. To evaluate the effectiveness of the framework, we compared the prediction model in Stage 3 with two well-known non-parametric models, namely Stochastic Blockmodel (SB) [7] and Mixed-Membership Stochastic Blockmodel (MMBS) [19]. Although the aforementioned models are not explicitly designed for link prediction, they can be modified for the prediction task using the above procedure of selecting the N highest probability edges [22]. In addition, these models suffer from the limitation of assuming a fixed number of vertices. Furthermore, we also compared our approach with common link predic-

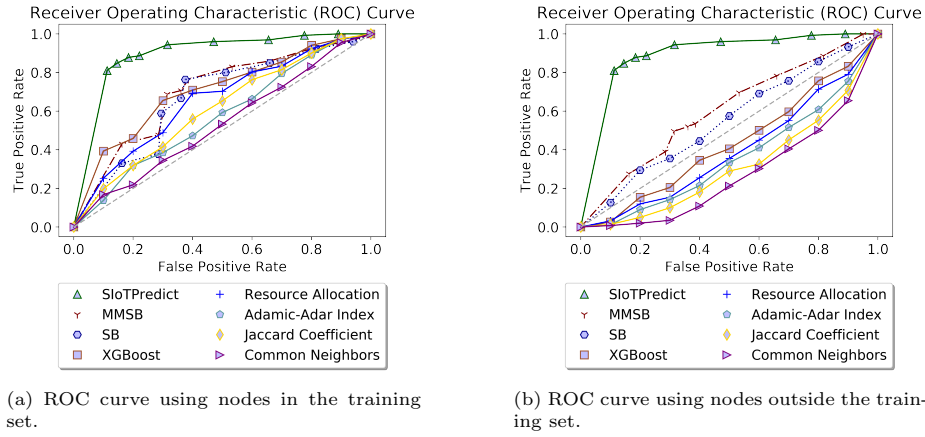


Fig. 5: ROC Curves for the SIoTPredict versus the comparison methods

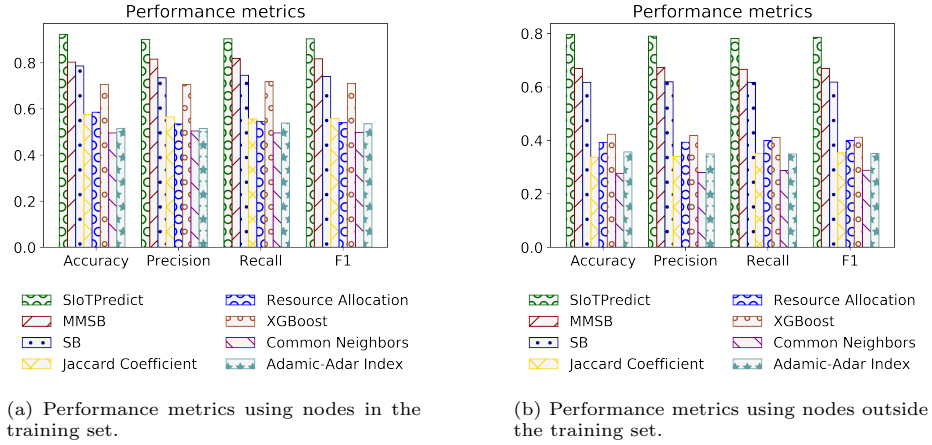


Fig. 6: Performance of the SIoTPredict versus the comparison methods.

tion methods [10]: Resource Allocation, Adamic Adar index, Jaccard Coefficient, XGBoost, and Common Neighbor.

4.3 Results

We modeled the SIoT network to predict future interactions among devices, and that enabled us to have a better understanding on the resulted network. Based on the existing Bayesian models, nodes are assigned to clusters, and these clusters control the way of how these nodes establish relationships. Figure 5 and Figure 6 show the performance of SIoTPredict against other methods. We used a small network, and the reason of that is due to the nature of SB and MMSB, which do not scale very well on large networks [16]. We experimented the performance of these models and methods in two ways. In the first experiment, we used the same nodes (i.e., IoT objects) in the training set and the test set. That means there were no nodes in the test set outside the training set. We performed experiments in this way because SB, MMSB and other methods assume nodes

in the test set are not outside the training set. In the second experiment, the nodes in the test set are outside the training set.

For the overall performance, SB does not perform well against the MMSB and our model. The reason is that the assumption of SB states that nodes can only belong to one cluster whereas we see MMSB performs better than SB because it relaxes this assumption by allowing the nodes to belong to more than one clusters. However, SB, MMSB and other common methods do not perform well compared to our model on both settings as illustrated in Figure 5 and Figure 6. In particular, these methods perform poorly in the second setting (i.e., the nodes in the test set are outside the training set) due to their limitation on dealing with new nodes. In contrast, our model delivers the similar performance.

5 Conclusion

Social Internet of Things (SIoT) can foster and enhance resource availability, discovering services, assessing object reputations, composing services, exchanging information and experience. In addition, SIoT enables establishing new acquaintances, collaborating to achieve common goals, and exploiting other object capabilities. Therefore, instead of relying on centralized search engine, social structure resulted from the created relationships can be utilized in order to find the desired services. In this paper, we take the research line of SIoT to a new dimension by proposing the SIoTPredict framework that addresses the link prediction problem in the SIoT paradigm. This framework contains three stages: i) collecting raw data movement of IoT devices, ii) generating temporal sequence networks of SIoT, and iii) predicting the links that are likely form between IoT objects in the future. Ongoing work includes further assessment of the SIoTPredict framework, and enhancement of the relationship prediction by considering the features of IoT objects (e.g., services offered by the objects).

References

1. Atzori, L., Iera, A., Morabito, G.: From Smart Objects to Social Objects: The Next Evolutionary Step of the Internet of Things. *IEEE Communications Magazine* **52**(1), 97–105 (2014)
2. Atzori, L., Iera, A., Morabito, G., Nitti, M.: The Social Internet of Things (SIoT) - When Social Networks Meet the Internet of Things: Concept, Architecture and Network Characterization. *Computer Networks* **56**(16), 3594–3608 (2012)
3. Bleecker, J.: A Manifesto for Networked Objects – Cohabiting with Pigeons, Arphids and Aibos in the Internet of Things. In: *Proceedings of the 13th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI)*. pp. 1–17 (2006)
4. Cai, D., Campbell, T., Broderick, T.: Edge-Exchangeable Graphs and Sparsity. In: *Proceedings of Advances in Neural Information Processing Systems (NIPS)*. pp. 4249–4257 (2016)
5. Girau, R., Martis, S., Atzori, L.: Lysis: A Platform for IoT Distributed Applications Over Socially Connected Objects. *IEEE Internet of Things Journal* **4**(1), 40–51 (2017)

6. Gomez-Rodriguez, M., Leskovec, J., Krause, A.: Inferring Networks of Diffusion and Influence. *ACM Transactions on Knowledge Discovery from Data (TKDD)* **5**(4), 1–37 (2012)
7. Holland, P.W., Laskey, K.B., Leinhardt, S.: Stochastic Blockmodels: First Steps. *Social Networks* **5**(2), 109–137 (1983)
8. Holmquist, L.E., Mattern, F., Schiele, B., Alahuhta, P., Beigl, M., Gellersen, H.W.: Smart-its friends: A Technique for Users to Easily Establish Connections Between Smart Artefacts. In: *Proceedings of International Conference on Ubiquitous Computing*. pp. 116–122 (2001)
9. Kranz, M., Roalter, L., Michahelles, F.: Things That Twitter: Social Networks and the Internet of Things. In: *Proceedings of What Can the Internet of Things Do for the Citizen (CIoT) Workshop at the 8th International Conference on Pervasive Computing (Pervasive)*. pp. 1–10 (2010)
10. Liben-Nowell, D., Kleinberg, J.: The Link-Prediction Problem for Social Networks. *Journal of the American Society for Information Science and Technology* **58**(7), 1019–1031 (2007)
11. Lü, L., Zhou, T.: Link Prediction in Complex Networks: A Survey. *Physica A: statistical mechanics and its applications* **390**(6), 1150–1170 (2011)
12. Marche, C., Atzori, L., Nitti, M.: A Dataset for Performance Analysis of the Social Internet of Things. In: *Proceedings of the IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications*. pp. 1–5 (2018)
13. Martínez, V., Berzal, F., Cubero, J.C.: A Survey of Link Prediction in Complex Networks. *ACM Computing Surveys (CSUR)* **49**(4), 1–33 (2016)
14. Mashal, I., Alsaryrah, O., Chung, T.Y.: Analysis of Recommendation Algorithms for Internet of Things. In: *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*. pp. 1–6 (2016)
15. Nitti, M., Atzori, L., Cvijikj, I.P.: Friendship Selection in the Social Internet of Things: Challenges and Possible Strategies. *IEEE Internet of Things Journal* **2**(3), 240–247 (2015)
16. Orbanz, P., Roy, D.M.: Bayesian Models of Graphs, Arrays and Other Exchangeable Random Structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **37**(2), 437–461 (2014)
17. Ortiz, A.M., Hussein, D., Park, S., Han, S.N., Crespi, N.: The Cluster Between Internet of Things and Social Networks: Review and Research Challenges. *IEEE Internet of Things Journal* **1**(3), 206–215 (2014)
18. Roopa, M., Pattar, S., Buyya, R., Venugopal, K.R., Iyengar, S., Patnaik, L.: Social Internet of Things (SIoT): Foundations, Thrust Areas, Systematic Review and Future Directions. *Computer Communications* **139**(1), 32–57 (2019)
19. Snijders, T.A., Nowicki, K.: Estimation and Prediction for Stochastic Blockmodels for Graphs with Latent Block Structure. *Journal of Classification* **14**(1), 75–100 (1997)
20. Tran, N.K., Sheng, Q.Z., Babar, M.A., Yao, L.: Searching the Web of Things: State of the Art, Challenges, and Solutions. *ACM Computing Surveys (CSUR)* **50**(4), 1–34 (2017)
21. Tran, N.K., Sheng, Q.Z., Babar, M.A., Yao, L., Zhang, W.E., Dustdar, S.: Internet of Things Search Engine. *Communications of the ACM* **62**(7), 66–73 (2019)
22. Williamson, S.A.: Nonparametric Network Models for Link Prediction. *The Journal of Machine Learning Research* **17**(1), 7102–7121 (2016)
23. Yao, L., Sheng, Q.Z., Ngu, A.H., Li, X.: Things of Interest Recommendation by Leveraging Heterogeneous Relations in the Internet of Things. *ACM Transactions on Internet Technology (TOIT)* **16**(2), 1–25 (2016)