Received 3 November 2022, accepted 21 November 2022, date of publication 30 November 2022,
date of current version 5 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3225734

RESEARCH ARTICLE

A Predictive Paradigm for Event Popularity in Event-Based Social Networks

THANH TRINH[1,2] **AND NHUNG VUONGTHI**[3]
[1]Faculty of Computer Science, Phenikaa University, Ha Dong, Hanoi 12116, Vietnam
[2]Phenikaa Research and Technology Institute (PRATI), A&A Green Phoenix Group JSC, Cau Giay, Hanoi 11313, Vietnam
[3]Hanoi School of Business and Management, Vietnam National University, Hanoi 11310, Vietnam

Corresponding author: Thanh Trinh (thanh.trinh@phenikaa-uni.edu.vn)

ABSTRACT Recently, event-based social networks (EBSNs) have been used as flexible online platforms that create online groups and make offline events for people. The success of popular offline events depends much on a participant number factor, which contributes to the growth of online groups and social networks. In this paper, we study a research problem of event popularity, where the popularity of an event is relevant to the number of participants of the event. In this work, we propose a predictive paradigm which consists of the procedure of generating features and training regression methods to estimate the popularity of events. We first crawled datasets and then generated features from the datasets. Finally, three famous regression methods, i.e., support vector machine, random forest, and decision tree, were used to predict the popularity of events. Extensive experiments were conducted on three city datasets with two different contexts of using these three datasets. In the city context, each city dataset was converted into a data table. Three regression methods used the data table to build predictive models and estimate the popularity of events. In the other context, each group in one city dataset was transformed into one group data table, and regression models were built on the group data table. Overall, the proposed paradigm with random forest is the best in terms of MAE and RMSE metrics. Moreover, this study has shown that for the city context, the event content is the best contributing factor that pushes people to engage in events. Furthermore, with the group context, the event time factor is very crucial to assist users in planning to join events.

INDEX TERMS Social networks, EBSNs, event popularity.

I. INTRODUCTION

Online social networks shape the way people work and communicate with each other. Moreover, with the rapid growth of online social networks, people have many choices to attend online events and offline activities. To combine online and offline events in one framework, event-based social networks (EBSNs) [1] are emerging, for example, Meetup, Douban, and Facebook. Hence, people are able to create and distribute events in these networks. Users are able to take part in any event that they are interested in it. Many groups are created with similar themes, and events of the groups are published with similar topics. For instance, groups have a start-up theme and events that are issued by those groups often have business topics.

Since one event is announced, this event's invitation is sent to users. There are a lot of works [2], [3], [4] about finding a list of users who are willing to attend this event. Recommending this event to users has been investigated by many researchers [5], [6], [7]. However, when creating a new event, this event's organizers always want to estimate the number of participants in order to prepare the event for participants as good as possible and save costs for this event. Obviously, the success of events depends on the participant number. In other words, the more people come, the more successful events are. Thus, the participant number of this event is the key factor that evaluates the sustainability of a group's event and even the growth of a social network. Hence, predicting participants in an event is also a challenging problem in social networks. Moreover, in EBSNs, there are no online tools to assist event organizers in estimating the number of participants when organizers create a new event.

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

The initial concept of event popularity is measured as the participant number. In this paper, we study many diversified events with very different participant numbers, for example, from social events to sportive activities. Therefore, we define a new metric based on the participant number to represent the popularity of events.

Predicting event popularity provides valuable information for administrators of social networks to deploy more services for users. Thus, it is highly demanded to develop an advanced technique for event popularity prediction over online social network platforms. In addition, the problem of event popularity is not studied thoroughly. These realities lead to open a new research problem: event popularity in these social networks.

In this paper, we study the problem of event popularity over event-based social networks. Furthermore, we provide a further understanding of online social networks through the problem of event popularity. This problem is formulated as follows: *Given a new event e^* published by a group g within an EBSN dataset, the objective is to predict the popularity of this event based on the historical events in the EBSN dataset.*

In this work, we propose a predictive paradigm which consists of four parts to estimate the popularity of events. Part 1 stores an EBSN dataset crawled from Meetup. Part 2 represents the three main groups of features based on three main factors of events, i.e., venue, time and content factors. Part 3 is implemented with three regression methods to estimate the popularity of events, i.e., random forest, support vector machine and decision tree. The event popularity is sent to event organizers in Part 4. In experiments, we carry out the proposed paradigm in two different contexts of using three crawled city datasets. For the first context, we first consider each city as one EBSN dataset. The three groups of features of all events are generated based on this EBSN dataset. Then, each regression method uses the generated features to build a predictive model. Next, a new coming event e^* is created by a group g and published in this EBSN. The proposed paradigm generates features of e^* with respect to all past events in the EBSN dataset, and the paradigm provides the generated features for predictive models in order to estimate the popularity of event e^*. In the other context, each group in one city is treated as one group dataset. Similar to the first context, features of only events in the group dataset are first created, and then predictive models are built on these features. Next, features of e^* are generated and used in the predictive models to forecast the popularity of e^*. To summarize, the contributions of our work are:

- The problem of event popularity in event-based social networks is defined.
- We propose a predictive paradigm to address the problem.
- In the proposed paradigm, we generate features from a dataset and train regression methods based on the features to predict the event popularity.
- We conduct extensive experiments on Meetup datasets consisting of three famous cities in the world to illustrate the accuracy and efficiency of the proposed paradigm.
- This work can be implemented as an online tool for event organizers.

The remainder of this paper is organized as follows. Section II briefly reviews related works. EBSNs terminologies and the problem are explained in Section III. Event popularity paradigm is offered in Section IV. Section V performs the empirical study. Conclusions are given in Section VI.

II. RELATED WORK

The concept of popularity and social trend predictions have been studied in many works [8], [9], [10], [11], [12]. Zhao et el. [8] recently studied event popularity over microblogs [8]. They addressed the problem of social trends popularity, which was measured as existing time of social trends in this work. Yin et al. [13] studied the problem of topic reading dynamics that was expressed by a set of keywords in Weibos. They proposed a model that can predict those who were interested in specific topics in Weibos. In another work [14], they investigated the behaviors of users within the context of Covid-19. In addition to this, they proposed SRFI model to predict the opinions of users about the pandemic through Chinese Sina blogs. Prediction of social trends about the vaccine was investigated in work [15], and they used rough set theory to evaluate the network of public opinions.

Another study on popularity in work [16] defined a research problem of online news popularity, which could be expressed by the number of shares, likes and comments. They first generated a list o features from articles and then used the boosting method to predict whether users shared a new article or not. Gao et al. [12] investigated the problem of future message popularity over the Weibo social network. The process of resending new messages was studied and they predicted the popularity by an extension of Poison model involving the time mapping process. The lifetime of online stories was presented in work [17], in which they provided an extensive analysis of the quality and the quantity of online articles in order to model social media interactions among readers. Lee et al. [11] illustrated a study on the popularity of online content. They aimed to predict the likelihood of a lifetime of online content by using a hazard regression model. They used two datasets with rich contents, i.e., forum.dpreview.com and forums.myspace.com, in their study. Almed et al. [18] defined the problem of popularity in user-generated content throughout YouTube, Digg and Vimeo. They proposed a two-stage method to predict content popularity. In the first stage, they analyzed content behaviors and generated features. In the second stage, they used a regression model to predict the values of content popularity. Moreover, to study the problem of online content popularity, another work has been investigated over Digg and Youtube [19].

Shang et al. [20] integrated social influence with homophily into a model to predict online content popularity. Dou et al. [21] also predicted online content popularity with rich information. In their work, they first selected contexts, then represented these contexts as a unified form, and finally utilized the form to predict the popularity. They proposed a knowledge-based method to enhance the accuracy of the popularity of online items. Lymperopoulos [22] clarified the online contents into two patterns: linear and non-linear growth periods. They modelled the popularity of those contents as a sequence of linear and non-linear phases and used these phases to predict popularity.

Liu et al. [1] had investigated Meetup and defined it as event-based social networks (EBSNs). Research problems of EBSNs have been defined by researchers, such as event recommendation [2], group recommendation [4] and active-friend recommendation [23], [24]. The problem of event attendees recommendation is expressed by selecting top N users who are likely to attend events. However, the problem of event popularity needs to be explored in EBSNs.

To predict a list of attendees at events, Wang et al. [25] proposed a model which was formed by a combination of a weak tie theory and a linear regression method. This study was conducted on data crawled from Facebook. In another work [10], Mehmood et al. analyzed the contents of events that were gathered from Twitter. They proposed a model which was based on LSTM in order to predict the participant number of events. Bhowmick et al. [26] defined a new concept of topical micro-categories in the context of EBSNs. This work designed a new methodology to explore micro-categories, which was clarified by the popularity profile of Meetup events. Chen et al. [27] studied the event popularity problem through Twitter. In their work, they first considered an event as a set of messages which involved hashtags. Then, they designed a new model based on hashtag-based and influence-based to predict popularity. Madisetty et al. [28] designed a study to investigate the problem of social media popularity of events. To do that, they proposed a model based on a deep learning method to estimate event popularity. Li et al. [9] studied the problem of group popularity. They proposed a deep neural network model that was constructed based on group-based, time-based features to predict group popularity.

In our work, we focus on the popularity of events within the context of event-based social networks. In the following section, we illustrate the structure of EBSNs and define the event popularity problem within these networks.

III. DEFINITIONS AND PROBLEM

A. EBSN TERMINOLOGIES

Event-based social networks (EBSNs) are one of the most active social networks currently. Meetup[1] is a famous example of EBSNs, and the social network is widely used in 190 countries. This network has 300000 groups, which create 10000 online and offline events per week and has more than 52 million users. Meetup only provides information of events about time, location, contents, and the list of participants of each event. And it does not provide information about reasons why events are canceled or delayed, such as weather conditions. Moreover, there are no direct links between users in Meetup network. EBSNs are constructed by four main entities, which are illustrated in Figure 1 and described as follows:

1) GROUPS

A group is initially created by only one user and organized by several users. The group founder can offer a short description of the group's theme in order to gain more users. The group stores happened events; moreover, upcoming events of this group are informed to this group's users and the whole users of an EBSN.

2) EVENTS

Any user in a specific group is allowed to creating an event, and the user is defined as the event organizer. Moreover, the event is published by the group. This created event is described by a detailed content. In addition, time and location factors are also involved in helping users to make a plan to engage in this event. Users will send a RSVP with YES to confirm attending this event; otherwise, they will reply with a RSVP with NO. Hence, each event has a list of participants

3) VENUES

Venue is a special entity in event-based social networks. A particular venue is demonstrated by a physical address with a specific location containing latitude and longitude. In EBSNs, people first join online groups and then create offline events, which are hosted in several venues where they meet each other in. Thus, a venue stores a list of hosted events. Choosing a suitable venue to host events is crucial to attracting more users to join.

4) USERS

When a user joins in EBSNs, he/she can be a member of one or several groups relevant to his/her interests. Even the user can create his/her own group. Since one event is sent to the user, this user will decide to engage in this event or refuse it. In EBSNs, there are no connections that indicate whether users are friends or not.

Figure 1 also describes the procedure of creating events and hosting events for users. For example, event Meeting is first created by user u^3, and issued by group g^2. Then, this event is described by a content, and hosted in venue v^2. In addition, users u^3 and u^5 engage in this event. In this figure, it is aware that there is no an online tool or a model to help events organizers forecast participants numbers.

B. PROBLEM STATEMENT

To hold a new event, forecasting how many users who want to take part in the event is a contributing factor to the success of it. The participant number is measured by RSVPs with YES in the event. In EBSNs, there are many different groups with diverse topics so the number of users who want to take part in different events is different. For example,

[1] meetup.com

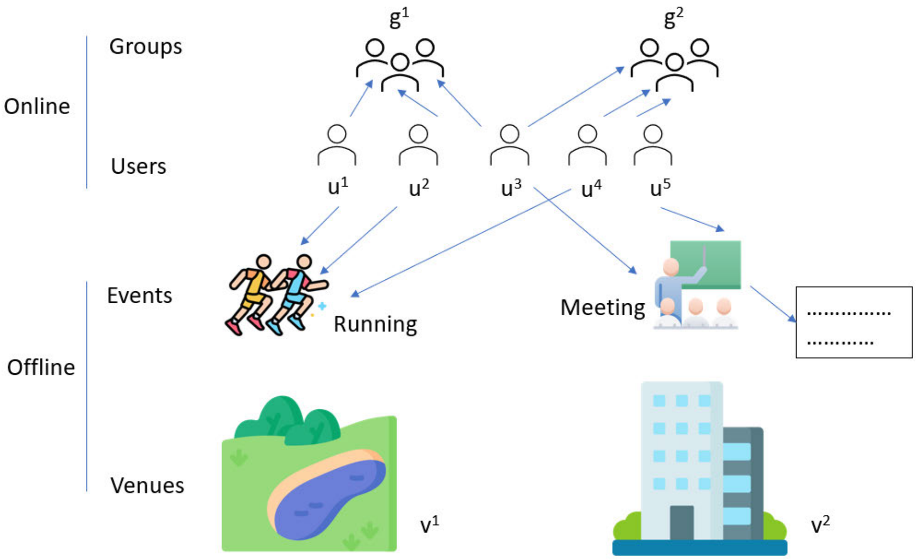


FIGURE 1. Example of an event-based social network.

group ID "15817402" about Web3 in Sydney changed its topics of events many times from blockchain topics to start-up topics in its events. And this group published 25 events in the period of 2017-2018. The participants in this group's events were very fluctuated from 5 participants to more than 200 participants. Hence, in this study, we propose a new metric to study the event popularity as follows:

$$p^i = \frac{\Sigma_1^N |e^i|}{N \times |e^i|} \tag{1}$$

where p^i is defined as the popularity of event e^i. N is the number of events issued by group g, $|e^i|$ is the number of participants in e^i.

Event Popularity Problem: Given a new event e^* issued by group g in an EBSN dataset, we aim to predict the popularity p^* of event e^* based on past events in this dataset. To address this problem, we propose a predictive paradigm in the following section.

IV. EVENT POPULARITY PARADIGM

This section presents our paradigm. We first discuss the architecture of the paradigm, and we then present a feature generation. Finally, we build regression models based on the generated features.

A. ARCHITECTURE OF THE PROPOSED PARADIGM

Figure 2 presents the architecture of the paradigm, which consists of four parts. The process of the proposed paradigm works as follows: Since a given EBSN dataset is stored in Part 1, we model them as relationships between entities in the EBSN model. Part 2 describes methods to yield features. Specifically, three major factors are selected to generate features of all events in this dataset. Three regression methods in Part 3 are chosen to build predictive models based on the generated features. Since a new event e^* is given, features of e^* are generated with respect to the dataset. The features of e^* are provided for predictive models to achieve the popularity p^* of this event and sent it to an event organizer in Part 4.

B. FEATURE GENERATION

Given an EBSN dataset, we make features based on the four main entities and the structure of this dataset. Specifically, given event e^* in group g; we leverage the information of three factors: venue, time and content of event e^* to make features of e^*. The features are grouped into three main categories, i.e., venue-based, time-based, and content-based features. To make a further clarification of the presentation, Table 1 describes notations and Table 2 illustrates generated features.

1) VENUE-BASED FEATURES

People prefer to engage in a new event due to several reasons. The new event is hosted in a popular venue that is convenient to go there. Moreover, the location of this event is close to previously attended events. Thus, choosing a suitable location or a convenient venue is very important to gain more users to attend this event. To generate a list of features from a new event e^* in group g with a physical location, we first collect events relevant to e^* as follows:

$$E^* = \{E | dis(e^*, e^i) < r\} \tag{2}$$

where E is the list of events extracted from a given EBSN dataset and $dis(e^*, e^i)$ is the Euclid distance in kilometer. A given threshold r of a radius is set to collect a list of events, denoted by E^*, each of which is in the radius of event e^*.

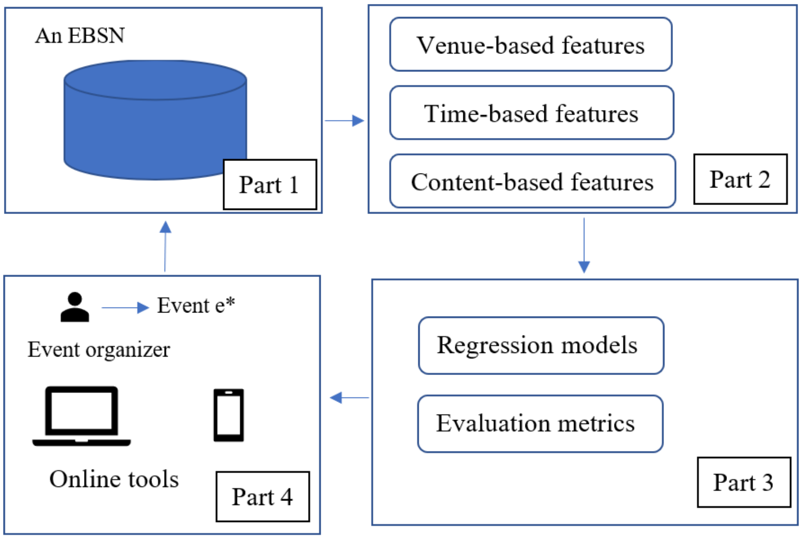


FIGURE 2. Architecture of event popularity paradigm.

We generate features of e^* with respect to E^* as follows:

$$V_{av} = \frac{\sum_{i=1}^{|E^*|} |e^i|}{|E^*|} \tag{3}$$

where V_{av} represents the average of events participants in list E^*. $|E^*|$ is the number of events in E^*, and it is considered one feature of event e^*. The three different features are also derived, i.e., $V_{min} = \text{argmin}\{|e^i|,\ e^i \text{ in } E^*\}$, $V_{max} = \text{argmax}\{|e^j| \mid e^j \text{ in } E^*\}$, and V_{sd}. In other words, V_{min} and V_{max} represent the smallest number of participants and the largest number of participants in the list of events E^*, respectively. And, V_{sd} is the standard deviation of events participants in E^*.

To understand more the relationship between events venues in the EBSN dataset, we first compute the distance similarity between each event e^i in E^* and e^* as the following equation:

$$S_V^i = e^{\frac{1}{dis(e^*,e^i)}} \tag{4}$$

where S_V^i is the distance similarity between the venue of e^* and the venue of event e^i. Then, we achieve list ES in the following equation:

$$ES = \{es^i |\ es^i = |e^i| \times S_V^i \text{ and } e^i \in E^* \} \tag{5}$$

Finally, the features of event e^* relevant to ES are created as:

$$V_{av}^{ES} = \frac{\sum_{i=1}^{|ES|} es^i}{|ES|} \tag{6}$$

where feature V_{av}^{ES} is the number average of es in list ES. Similar to list E^*, we also have features V_{min}^{ES}, V_{max}^{ES}, and V_{sd}^{ES} of event e^* based on list ES.

Event e^* is published by group g, with this, we make other features of e^* that are only relevant to group g. We first select a list of events in E that are only issued by g, denoted by E^g.

$$E^g = \{e^j | e^j \text{ published in group } g \text{ and } e^j \in E\} \tag{7}$$

Equation 4 is also taken to compute the distance similarity between the venue of e^* and the venue of e^j in E^g. As a result, we have a list $ES^g = \{\ es^j \mid es^j = |e^j| \times S_V^j \text{ and } e^j \in E^g\ \}$ with $|E^g|$ elements.

Similar to E^* and ES lists, we create five features of e^* referred to E^g and four other features of e^* relevant ES^g. Those nine features are described in Table 2.

2) TIME-BASED FEATURES

People often make a plan to take part in events in a specific day of the week and at a particular time, for instance, at 5 pm on Saturday. Moreover, if they suddenly have free time during one day, they will look for a suitable event and join in it. Hence, we separate the time-based factor into Day of Week and Hour of Day factors and generate features based on these two factors.

a: DAY OF WEEK

To make features based on this factor, we first only select events in E^* that those events, denoted by E_D, are hosted on the same day of the week with event e^*, such as Saturday. Then, we obtain a list of events $ES_D = \{\ es^d \mid es^d = |e^d| \times S_V^d$ and $e^d \in E_D\,\}$. Hereby, E_D and ES_D lists generate nine features of e^* by the same way of creating features based on E^* and ES lists. Finally, we collect events in E that those events are issued by group g and held in the same day of the week with event e^*, denoted by E_D^g. As a result, we achieve the list of events $ES_D^g = \{\ es^i \mid es^i = |e^i| \times S_V^i \text{ and } e^i \in E_D^g\}$. Thus, these two lists, E_D^g and ES_D^g, result in nine features as E^g and ES^g do. All features based on this factor are described in Table 2.

b: HOUR OF DAY

Similar to Day of Week factor, we achieve four event lists as follows:

$$E_H = \{e^h | e^h \text{ created in the same Hour of Day of } e^*$$

$$\text{and } e^h \in E^*\} \tag{8}$$

$$ES_H = \{es^i \mid es^i = |e^i| \times S_V^i \text{ and } e^i \in E_H\} \tag{9}$$

$$E_H^g = \{e^t \mid e^t \text{ created in the same Hour of Day of } e^* \text{ and } e^t \in E, \text{ and } e^t \text{ issued by } g\} \tag{10}$$

$$ES_H^g = \{es^j \mid es^j = |e^j| \times S_V^j \text{ and } e^j \in E_H^g\} \tag{11}$$

The two lists, E_H and ES_H, create nine features for e^*. Likewise, we obtain nine other features for e^* from E_H^g and ES_H^g. Those features are presented in Table 2.

3) CONTENT-BASED FEATURES

Event e^* that is announced in an EBSN often offers an explicit content, which includes a title and a description of this event. This content has an impact on users' decisions about whether to go or not. Therefore, we create features based on the content similarity.

The content of each event can be represented as a vector of terms. Hence, given two events e^* and e^i with two vectors of terms T^* and T^i, respectively, the content similarity between two events is computed as Equation 12:

$$t^i(e^*, e^i) = \frac{T^* \cdot T^i}{\|T^*\|\|T^i\|} \tag{12}$$

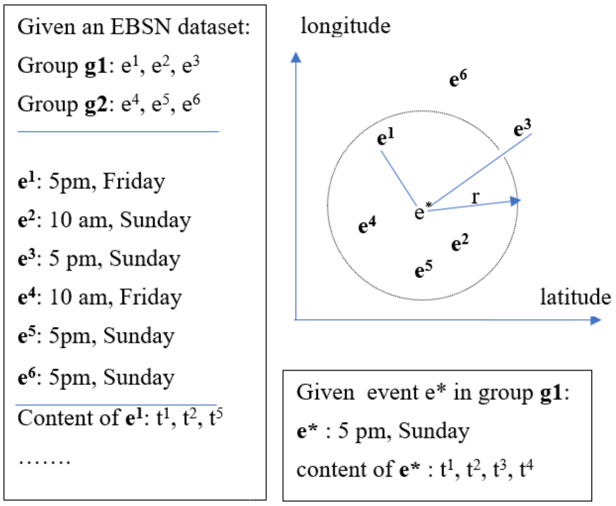
where $t(.,.)$ is the cosine similarity score between two events, the value of t is from [0, 1]. The higher value of t indicates that the two events are more relevant in content. In addition, we obtain two new lists of events that are relevant to e^* as follows:

$$ES_C = \{es^i \mid es^i = |e^i| \times t^i(e^*, e^i) \text{ and } e^i \in E^*\} \tag{13}$$

$$ES_C^g = \{es^j \mid es^j = |e^j| \times t^j(e^*, e^j); e^j \in E; \text{ and } e^j \text{ published by } g\} \tag{14}$$

These two lists, ES_C and ES_C^g, yield eights features for e^* as list ES does. Those eight features are also described in Table 2.

Example of Obtaining Lists of Events: Figure 3 shows an example of an EBSN dataset including two groups $g1$ and $g2$, and a set E containing six events. Since an upcoming event e^* is published by g1, we gain lists of events relevant to e^* as follows: Given a threshold r, we obtain a list of events $E^* = \{e^1, e^2, e^4, e^5\}$ as shown in the circle in Figure 3. The distance similarity between e^* and each event in E^* is computed by Equation 4; therefore, we have a list of elements ES. E^{g1} contains events e^1,e^2, and e^3. Moreover, ES^{g1} is also obtained. Events in list E_D are e^2 and e^5 due to hold on Sunday as event e^*. ES_D are to be created. E_D^{g1} consists of e^2 and e^3, thus, list ES_D^{g1} is also achieved. Similarly, list E_H stores e^1 and e^5, and list E_H^{g1} includes e^1 and e^3. Easily, we obtain two lists, ES_H and ES_H^{g1}. To generate two lists ES_C and ES_C^{g1} we need to compute the similarity based on the content of event e^* and contents of events in E^* and E^{g1}. For example, the content of e^* and the content of e^1 are presented by terms $\{t^1, t^2, t^3, t^4\}$ and $\{t^1, t^2, t^5\}$, respectively. The content similarity which is calculated by Equation 12 is 0.58. Finally, we have all lists of events, which are used to generate all features of e^* with respect to the given EBSN dataset. All features are listed in Table 2.


Given an EBSN dataset:
Group **g1**: e^1, e^2, e^3
Group **g2**: e^4, e^5, e^6

e^1: 5pm, Friday
e^2: 10 am, Sunday
e^3: 5 pm, Sunday
e^4: 10 am, Friday
e^5: 5pm, Sunday
e^6: 5pm, Sunday
Content of **e^1**: t^1, t^2, t^5
…….

Given event e* in group **g1**:
e* : 5 pm, Sunday
content of **e*** : t^1, t^2, t^3, t^4

FIGURE 3. Example of EBSN dataset.

TABLE 1. Notations.

$EBSN$	An event-based social network
$RSVP$	Please reply
YES	A number of participants in each event
E	The list of events in the given EBSN
g	Group g
e^*	An upcoming event
r	Given threshold r in km
E^*	The list of events obtained in Equation 2
ES	Obtained in Equation 5
E^g	Events that are in the same group with e^*
ES^g	Each element of this list is defined by $es^j = \|e^j\| \times S_V^j$.
E_D	The list of events that are created in the same day of week with e^*
ES_D	Each element of this list is defined by $es^j = \|e^j\| \times S_V^j$ and $e^j \in E_D$
E_D^g	The list of events that are created in the same day of week with e^* and issued by g
ES_D^g	Each element of this list is defined by $es^j = \|e^j\| \times S_V^j$ and $e^j \in E_D^g$
E_H	Expressed in Equation 8
ES_H	Expressed in Equation 9
E_H^g	Described in Equation 10
ES_H^g	Described in Equation 11
ES_C	Described in Equation 13
ES_C^g	Described in Equation 14

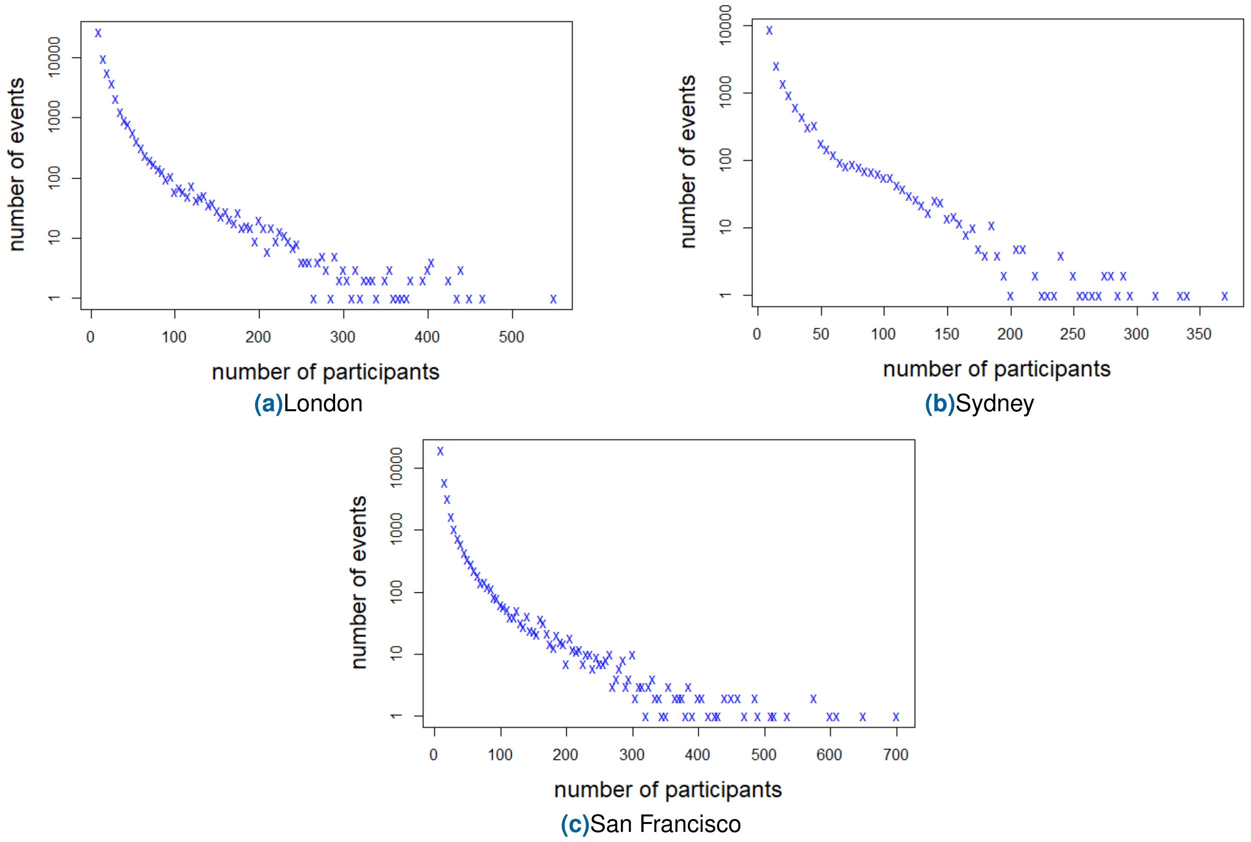
C. REGRESSION METHODS

Based on the feature generation stage, we achieve a list of generated features of all events in E. In other words,

TABLE 2. The 64 features derived from datasets.

Feature and description	
Venue-based features	**Time-based features**
	Day of Week
V_N The number of events in E^*	$T_N^{E_D}$ The number of events in E_D
V_{av} Average number of events participants (YES) in E^*	$T_{av}^{E_D}$ Average number of events participants in E_D
V_{min} The minimum of YES in all events in E^*	$T_{min}^{E_D}$ The minimum of YES in all events in E_D
V_{max} The maximum of YES in all events in E^*	$T_{max}^{E_D}$ The maximum of YES in all events in E_D
V_{sd} The standard deviation of YES in E^*	$T_{sd}^{E_D}$ The standard deviation of YES in E_D
V_{av}^{ES} Average number of YES in ES	$T_{av}^{ES_D}$ Average number of events participants in ES_D
V_{min}^{ES} The minimum of YES in all events in ES	$T_{min}^{ES_D}$ The minimum of YES in all events in ES_D
V_{max}^{ES} The maximum of YES in all events in ES	$T_{max}^{ES_D}$ The maximum of YES in all events in ES_D
V_{sd}^{ES} The standard deviation of YES in ES	$T_{sd}^{ES_D}$ The standard deviation of YES in ES_D
$V_N^{E^g}$ The number of events in E^g	$T_N^{E^g_D}$ The number of events in E^g_D
$V_{av}^{E^g}$ Average number of events participants (YES) in E^g	$T_{av}^{E^g_D}$ Average number of events participants in E^g_D
$V_{min}^{E^g}$ The minimum of YES in all events in E^g	$T_{min}^{E^g_D}$ The minimum of YES in all events in E^g_D
$V_{max}^{E^g}$ The maximum of YES in all events in E^g	$T_{max}^{E^g_D}$ The maximum of YES in all events in E^g_D
$V_{sd}^{E^g}$ The standard deviation of YES in E^g	$T_{sd}^{E^g_D}$ The standard deviation of YES in E^g_D
$V_{av}^{ES^g}$ Average number of events participants in ES^g	$T_{av}^{ES^g_D}$ Average number of events participants in ES^g_D
$V_{min}^{ES^g}$ The minimum of YES in all events in ES^g	$T_{min}^{ES^g_D}$ The minimum of YES in all events in ES^g_D
$V_{max}^{ES^g}$ The maximum of YES in all events in ES^g	$T_{max}^{ES^g_D}$ The maximum of YES in all events in ES^g_D
$V_{sd}^{ES^g}$ The standard deviation of YES in ESs^g	$T_{sd}^{ES^g_D}$ The standard deviation of YES in ES^g_D
Content-based features	**Hour of day**
$C_{min}^{ES_C}$ The minimum of YES in all events in ES_C	$T_N^{E_H}$ The number of events in E_H
$C_{av}^{ES_C}$ Average number of events participants in ES_C	$T_{av}^{E_H}$ Average number of events participants in E_H
$C_{max}^{ES_C}$ The maximum of YES in all events in ES_C	$T_{min}^{E_H}$ The minimum of YES in all events in E_H
$C_{sd}^{ES_C}$ The standard deviation of YES in ES_C	$T_{max}^{E_H}$ The maximum of YES in all events in E_H
$C_{min}^{ES^g_C}$ The minimum of YES in all events in ES^g_C	$T_{sd}^{E_H}$ The standard deviation of YES in E_H
$C_{av}^{ES^g_C}$ Average number of events participants in ES^g_C	$T_{av}^{ES_H}$ Average number of events participants in ES_H
$C_{max}^{ES^g_C}$ The maximum of YES in all events in ES^g_C	$T_{min}^{ES_H}$ The minimum of YES in all events in ES_H
$C_{sd}^{ES^g_C}$ The standard deviation of YES in ES^g_C	$T_{max}^{ES_H}$ The maximum of YES in all events in ES_H
Other features	$T_{sd}^{ES_H}$ The standard deviation of YES in ES_H
$Weekday$ A day in a week that event is held in	$T_N^{E^g_H}$ The number of events in E^g_H
$Hour$ Time to host events	$T_{av}^{E^g_H}$ Average number of events participants in E^g_H
	$T_{min}^{E^g_H}$ The minimum of YES in all events in E^g_H
	$T_{max}^{E^g_H}$ The maximum of YES in all events in E^g_H
	$T_{sd}^{E^g_H}$ The standard deviation of YES in E^g_H
	$T_{av}^{ES^g_H}$ Average number of events participants in ES^g_H
	$T_{min}^{ES^g_H}$ The minimum of YES in all events in ES^g_H
	$T_{max}^{ES^g_H}$ The maximum of YES in all events in ES^g_H
	$T_{sd}^{ES^g_H}$ The standard deviation of YES in ES^g_H

we transform the given EBSN dataset into a data table $D = \{F, P\}$, each D^i represents a list of generated features F^i, which is shown in Table 2, and the popularity p^i of event e^i. We use D to train regression models. For a new event e^*, we obtain generated features of e^*, denoted by F^*, which is used in the trained models to predict the popularity p^* of e^*. In this work, we select decision tree (DT) [29], support vector machine (SVM) [30], and random forest (RF) [31] methods to predict the popularity of events.



(a)London

(b)Sydney

(c)San Francisco

FIGURE 4. The distribution between events and participants in the three regions.

V. EMPIRICAL STUDY

A. EBSN DATASETS

To gain an overview of event popularity, we select three famous regions, i.e., Sydney, London, and San Francisco, in the world to collect datasets from Meetup. The selected cities provide huge data with various events topics and many users. Each city is treated as an EBSN dataset. The datasets are gathered in the period of two years, 2017-2018. For each city, we selected all groups, and each group published at least 15 events in these two years. Furthermore, each event was hosted in a real physical venue with a specific location and this event had at least 5 participants. Table 3 gives statistics of the three gathered datasets. Based on this table, each user of each EBSN dataset had engaged in an average of five events for the two-year period. The distributions of users in attended events in the three EBSN datasets are depicted in Figure 4. It is observed that the majority of events had less than 50 participants.

B. EXPERIMENTAL SETUP

We use Lucene[2] to make terms, which are used to represent events contents [32]. Specifically, we remove all stop words and only keep terms in each event's content. Moreover, we also keep events with specific locations, which include longitude and latitude. Threshold r is set to 0.5 km to obtain events relevant to event e^*.

TABLE 3. Dataset statistics.

City	#groups	#events	#users	#YES
San Francisco	776	35477	155243	659808
Sydney	361	16937	67834	328356
London	1003	53753	198980	956815

To gain further understanding of how factors affect the decision of users and the popularity of events, experiments are conducted on two contexts of using datasets.

1) THE CITY CONTEXT

Each city (or EBSN) dataset is considered one city dataset, which is used in the proposed paradigm. Specifically, we first sort all events in each city on event time, then we divide the events into two parts: 80% for training and 20% for testing. Training part is defined as a list of events E. We first transform E into a data table $D = \{\text{F}, \text{P}\}$. Then, D is used to train the three selected regression methods. Features of each event e^* in testing part are generated with respect to E, denoted by a vector of features $\{F^*\}$. And, this vector is run into trained models to predict the popularity p^* of event e^*.

[2] https://lucene.apache.org/

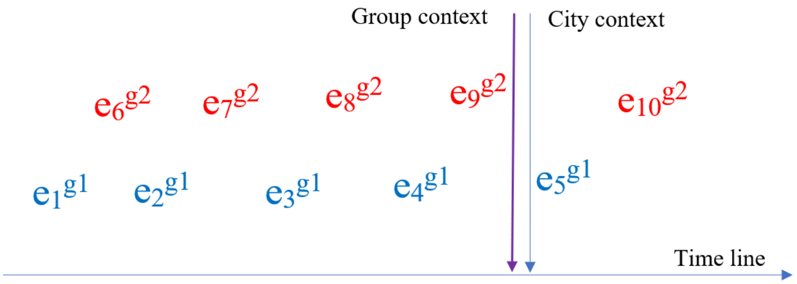


FIGURE 5. Example of splitting events in an EBSN into training and testing parts.

2) THE GROUP CONTEXT

We treat each group in each city (or EBSN) as a group dataset. We first sort all events in a group dataset on event time, then we split the group dataset into two parts: 80% for training and 20% for testing. The procedure of making a data table D for training part and features of each event in testing part is similar to it for the city context. To make further clarification of making features of events within the two contexts, we give the following example.

3) EXAMPLE OF GENERATING FEATURES OF TRAINING AND TESTING PARTS FOR THE TWO CONTEXTS

Figure 5 describes examples of splitting a given EBSN with two groups into training and testing parts. Events are sorted on event time, as shown in Figure 5. For the city context, we split the events datasets into two parts: testing part consists of events e^5 in group g^1 and e^{10} in group g^2; and the rest of the events datasets, denoted by E (8 events), is designed as training part. Each event e^i in E will generate features of it based on $E\backslash e^i$. Therefore, we have a data table D which consists of generated features of all events in E. Then, for each event in testing part, we make features of this event with respect to all events in E.

For the group context, each group is defined as one group dataset. For example, dataset g1 has five events. A given specific time is to split events of $g1$ into two parts: 80% for training and 20% for testing. Testing part of $g1$ only has event e^5, and train part consists of e^2, e^3, e^4 and e^5. To make features of all events in training part, we first collect all events in this EBSN dataset that they are held before the splitting time. Hence, we have list $E = \{e^1$, e^2,e^3,e^4,e^6, e^7,e^8,$e^9\}$. Features of each event e in training part (e^2, e^3, e^4, e^5) are made with respect to $E\backslash e$. Thus, we achieve a training data table D only containing four events and use D to train regression models. Features of e^5 in testing part of $g1$ are yielded with respect to all events in E.

Table 4 describes the time of generating features for each group in each city within both contexts. It can be seen clearly that groups with few events take less time to create features compared to groups with many events.

C. EVALUATION METRICS

These two metrics, MAE and RMSE, are widely used to measure the performance of regression models. Therefore, MAE and RMSE are selected to evaluate the differences between actual values and predicted ones. These two metrics are defined in Equation 15 and Equation 16 respectively.

$$MAE = \frac{1}{M}\sum_{i=1}^{M}|p^i - p^i_{predicted}| \tag{15}$$

$$RMSE = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(p^i - p^i_{predicted})^2} \tag{16}$$

where p^i and $p^i_{predicted}$ are the actual values and the predicted values of event popularity. M is the number of events in each testing part. The two metrics, MAE and RMSE, are used to assess the performance of the three regression models in the city context. For the group context, we use two new metrics that are defined in the following equations:

$$nRMSE = \frac{RMSE}{n} \tag{17}$$

$$nMAE = \frac{MAE}{n} \tag{18}$$

where n is the number of groups in each city.

Platform: All algorithms are implemented in Python and executed in a machine with a dual-core CPU 3.4GHz and 16GB Ram. The number of trees in random forest model is set to 100 trees. CART is used to build the tree model. And, RBF kernel is involved in support vector machine method.

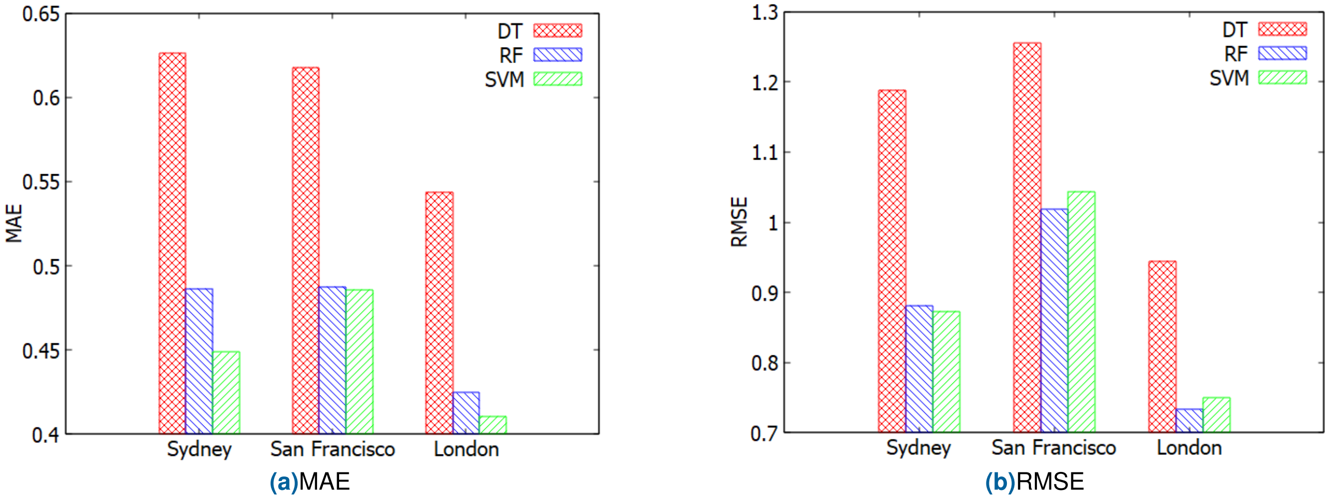
D. RESULT ANALYSIS AND DISCUSSION

1) PERFORMANCE OF PROPOSED PARADIGM IN THE CITY CONTEXT

Figure 6 illustrates the results of MAE and RMSE metrics from the selected three regression methods for the three cities. These three methods use all features (listed in Table 2) to build models based on training parts, then predict the popularity of each event in testing parts. In general, decision tree (DT) yields the worst results of two metrics for three cities. Support vector machine (SVM) gives the best scores of the

TABLE 4. Time (in seconds) of generating features for three cities in both contexts.

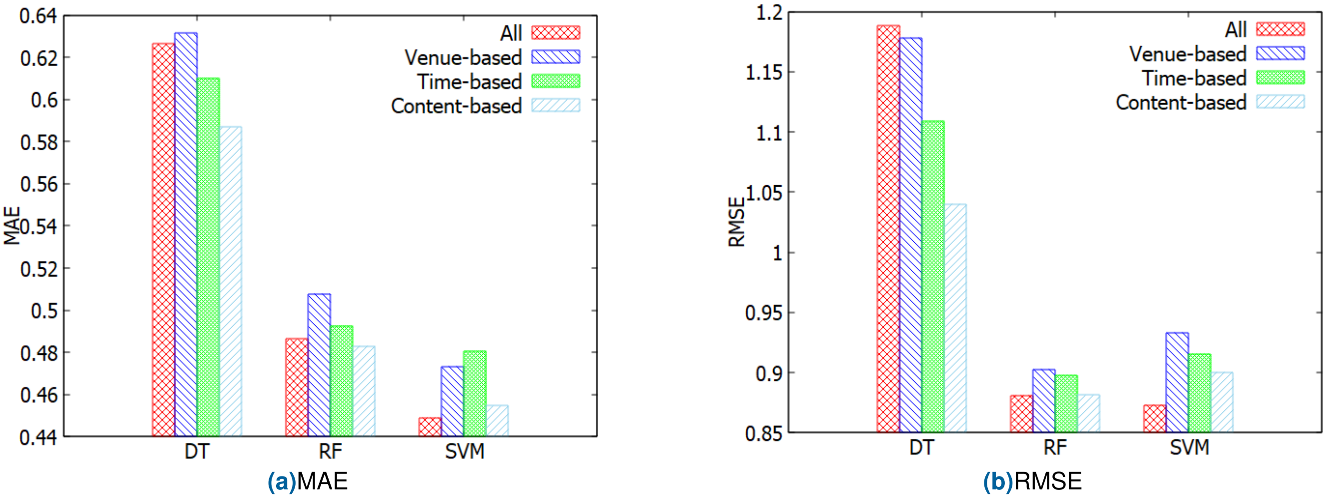
	Group context			City context		
	Shortest time for a group	Longest time for a group	For all groups	Shortest time for a group	Longest time for a group	For all groups
San Francisco	0.12 (15 events)	40.6 (749 events)	1315	0.53 (15 events)	41.1(749 events)	1388
Sydney	0.12 (15 events)	7.5 (228 events)	348	0.28 (15 events)	8.5 (228 events)	377
London	0.28 (15 events)	73.7 (836 events)	3573	0.95 (15 events)	75.2 (836 events)	3750



(a)MAE

(b)RMSE

FIGURE 6. The performance of three methods on the whole three datasets in the city context.



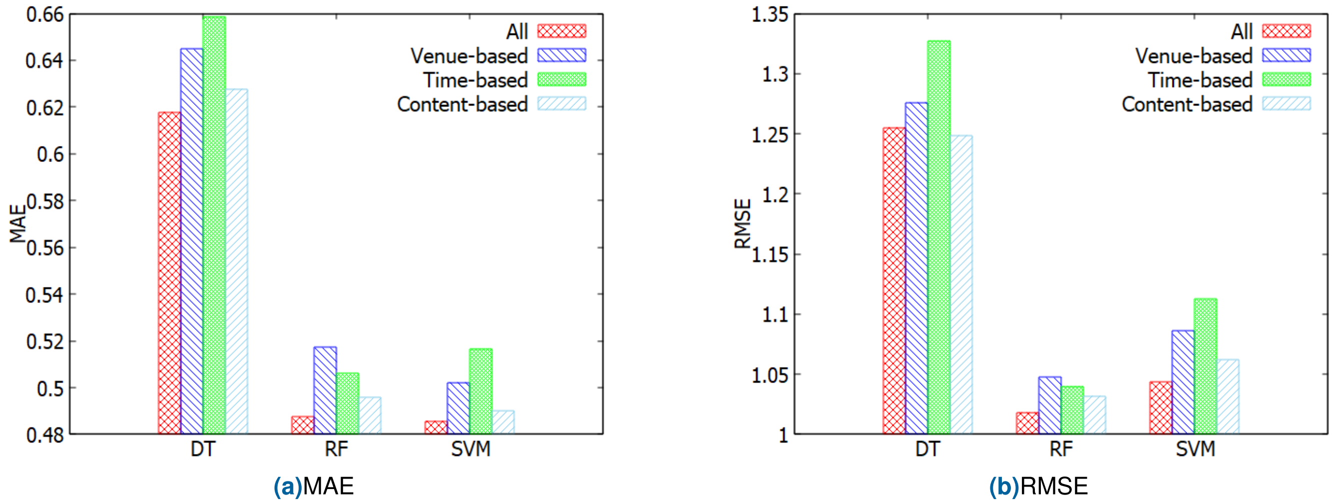
(a)MAE

(b)RMSE

FIGURE 7. Sydney: Performance of three regression methods with different four groups of features in the city context.

three datasets in terms of MAE metric; meanwhile, random forest (RF) is the best model in terms of RMSE metric.

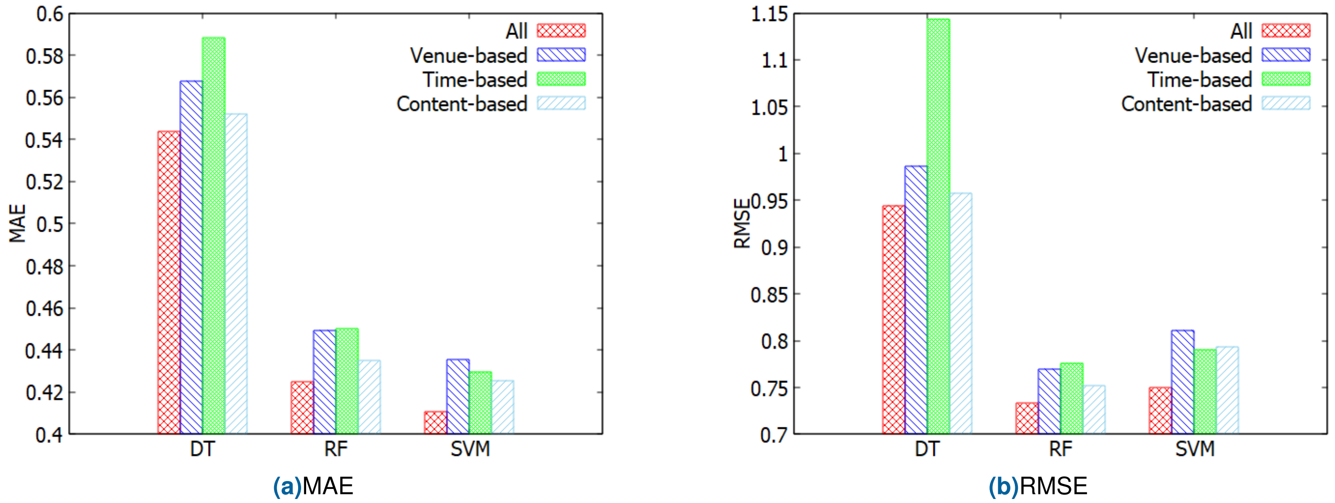
We also compare the performance of these regression models with the four different groups of features, i.e., all, venue-based, time-based, and content-based features. Figures 7, 8, and 9 describe the results of each model corresponding to each group of features for three cities. Overall, models that are built on all features (All) yield the best results. It is observed that the models that are built based on the group of content-based features provide better results than those built on groups of venue-based and time-based features. In addition to this, SVM with the group of content-based



(a)MAE

(b)RMSE

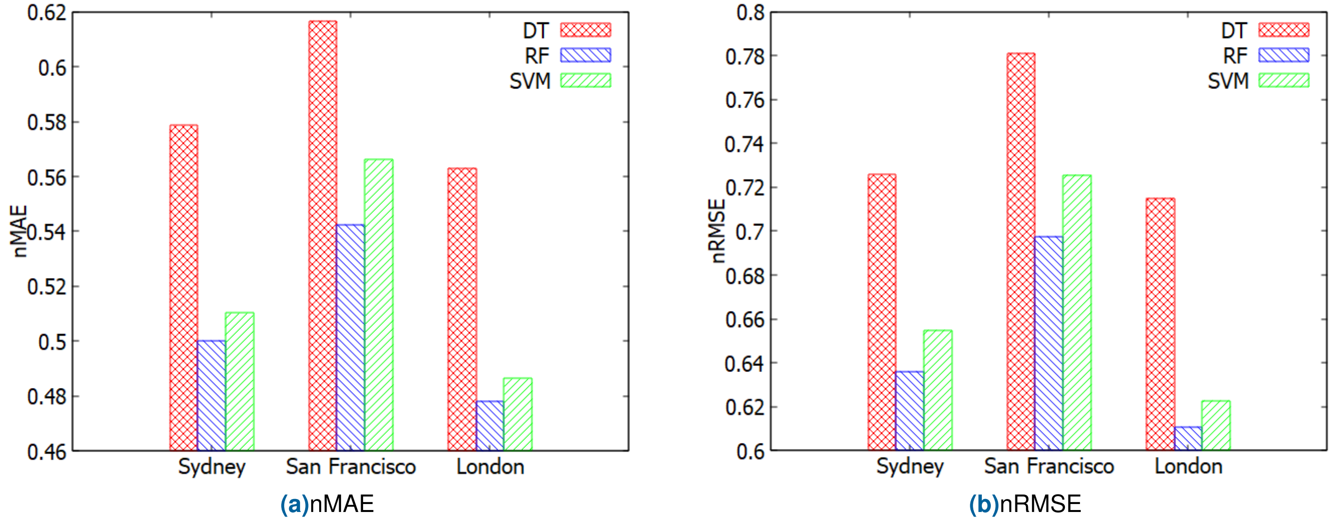
FIGURE 8. San Francisco: Performance of three regression methods with different four groups of features in the city context.



(a)MAE

(b)RMSE

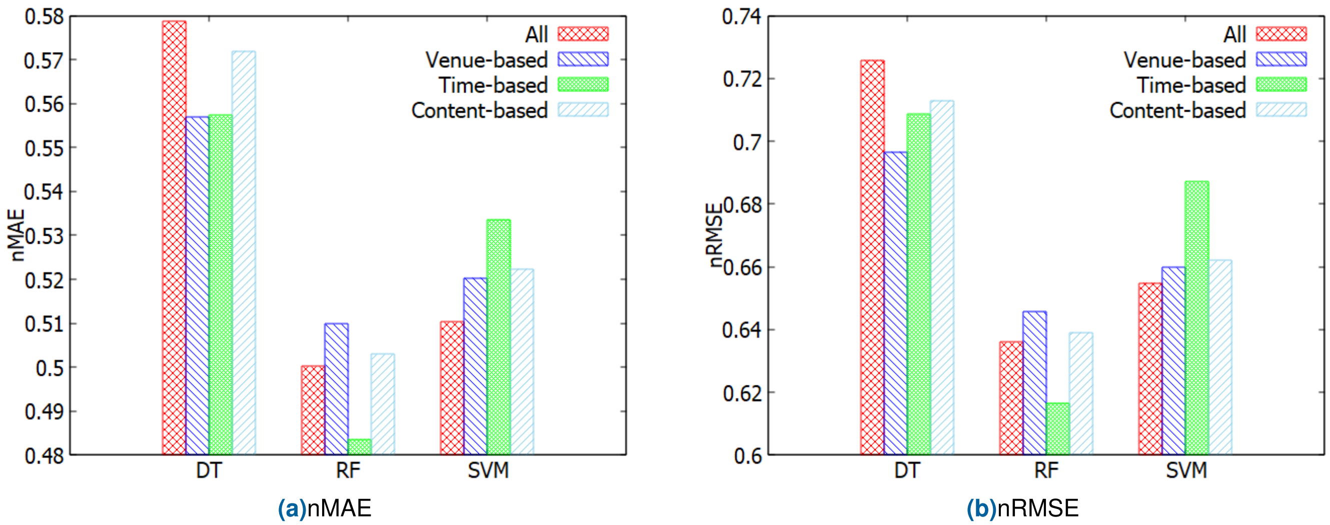
FIGURE 9. London: Performance of three regression methods with different four groups of features in the city context.



(a)nMAE

(b)nRMSE

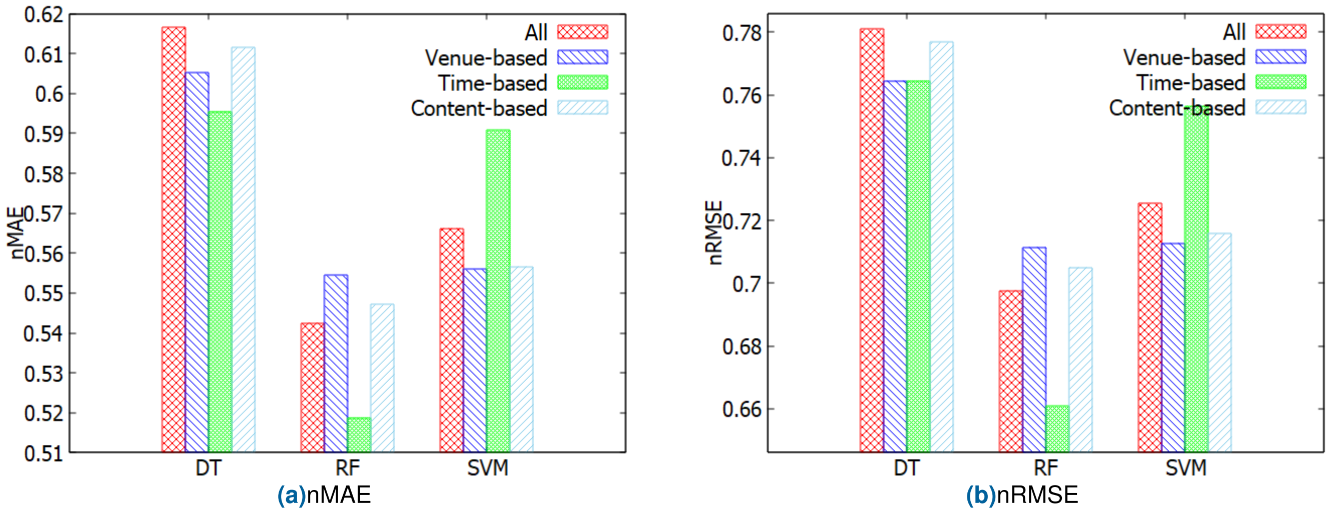
FIGURE 10. The performance of three methods on the whole three datasets in the group context.



(a)nMAE

(b)nRMSE

FIGURE 11. Sydney: Performance of three regression methods with different four groups of features in the group context.



(a)nMAE

(b)nRMSE

FIGURE 12. San Francisco: Performance of three regression methods with different four groups of features in the group context.

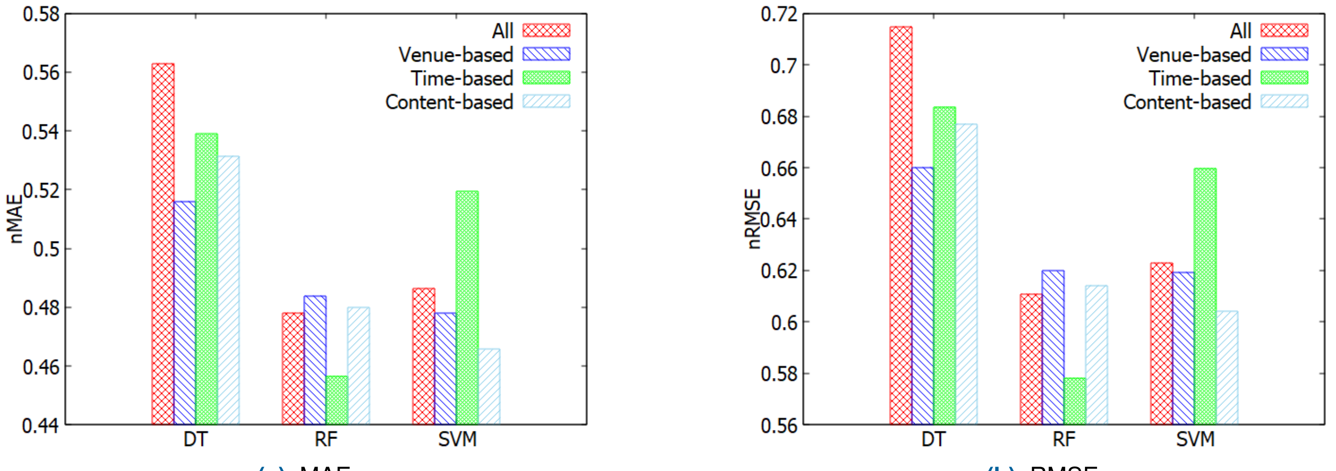
features yields the best results of MAE for three cities, and RF with this group is the best in terms of RMSE. DT with different groups of features is still the weakest method.

The first context (or an EBSN dataset) has many groups with diversified themes. Each group published many events with various topics. In addition, the participant numbers in different events are much dissimilar. Hence, the role of events contents is very critical to attract more people to take part in those events. Based on the results yielded from different groups of features, we can conclude that the contents of offline activities are the most valuable factor in the city context. Obviously, people often come to discuss a certain topic, or they have specific purposes of attending, for example, learning start-up skills. Thus, social network administrators need to improve the contents of events and follow up on social trends in order to keep users stay in their networks.

2) PERFORMANCE OF PROPOSED PARADIGM IN THE GROUP CONTEXT

We design each group in one city as a dataset, and split this dataset into two parts. The two metrics, nRMSE and nMAE, in Equation 17 and 18 are used to compare the performance of the three regression models.

The results of nMAE and nRMSE yielded by the three regression methods with all features for the three cities are demonstrated in Figure 10. In general, RF outperforms the two compared methods in terms of the two metrics. Otherwise, DT is still the worst method in all three cities. In this context, each group is treated as one dataset to build predictive models. Many groups in each city do not have many events; therefore, the training data table transformed from one group dataset copes with the problem of high dimensional data. Moreover, RF model is constructed from 100 trees, and



(a)nMAE

(b)nRMSE

FIGURE 13. London: Performance of three regression methods with different four groups of features in the group context.

each node of a tree is built based on the best feature. That are reasons why RF is better than SVM in the group context.

Similar to the first context, we also compare the performance of three predictive models with different groups of all, venue-based, time-based, and content-based features, respectively. The results of the comparisons are shown in Figure 11, Figure 12, and Figure 13. Overall, RF is still the best model for the four different groups of features; meanwhile, DT results in the worst metrics for the four groups of features.

Furthermore, RF built with time-based features yields better results of the two metrics than RF built with all features. In addition, RF trained with content-based features provides better results than it trained with venue-based features. These realities of the group context are different from the results of the city context. They are explained as follows: (1) Each group has only a few topics of events, even some group only has one topic for all events; (2) In EBSNs, event organizers often select the same venue to host offline activities; (3) Since attending previous events, users already know the topics of events and locations of events. Hence, the time factor is the most important character to push users to engage in new events; moreover, they will select events that are suitable for their free time.

We can conclude that in the small context of social networks, such as the group context, the time and content factors are the most contributing factors to the success of events. Hence, organizers need to select a suitable time to hold events and offer attractive contents in order to gain more people coming.

VI. CONCLUSION

In this paper, we present a study on event popularity over event-based social networks. For this objective, we propose a new paradigm to predict the popularity of events by transforming a dataset into a data table that can be used in regression methods. The proposed paradigm first stores an EBSN dataset, and then it makes features from this dataset. Three well-known regression methods are involved in the proposed paradigm to build predictive models based on generated features. Finally, the popularity of events is sent to event organizers. This study is conducted on three cities with two contexts of using datasets. Overall, RF is the best method to yield event popularity in the two contexts. We find that in the context of the whole city, the event content is the best contributing factor to affect people to join events. However, for the group context, event time is very crucial to make users engage in events. This study not only shows the impact of attracting content and suitable hosting time of events when event organizers create offline activities but also helps administrators of social networks to be aware of the importance of events contents. This work opens a new promising direction for future work: time-optimized planning for events and users, in other words, how organizers can catch users.

REFERENCES

[1] X. Liu, Q. He, Y. Tian, W.-C. Lee, J. McPherson, and J. Han, "Event-based social networks," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*. New York, NY, USA: ACM Press, 2012, p. 1032. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2339530.2339693

[2] T. Lan, L. Guo, X. Li, and G. Chen, "Research on the prediction system of event attendance in an event-based social network," *Wireless Commun. Mobile Comput.*, vol. 2022, pp. 1–14, 2022, Art. no. 1701345, doi: 10.1155/2022/1701345.

[3] T. Trinh, N.-T. Nguyen, D. Wu, J. Z. Huang, and T. Z. Emara, "A new location-based topic model for event attendees recommendation," in *Proc. IEEE-RIVF Int. Conf. Comput. Commun. Technol. (RIVF)*, Mar. 2019, pp. 1–6.

[4] Y. Jhamb and Y. Fang, "A dual-perspective latent factor model for group-aware social event recommendation," *Inf. Process. Manage.*, vol. 53, no. 3, pp. 559–576, May 2017, doi: 10.1016/j.ipm.2017.01.001.

[5] T. J. Ogundele, C.-Y. Chow, and J.-D. Zhang, "SoCaST*: Personalized event recommendations for event-based social networks: A multi-criteria decision making approach," *IEEE Access*, vol. 6, pp. 27579–27592, 2018.

[6] C. Xu, "A novel recommendation method based on social network using matrix factorization technique," *Inf. Process. Manage.*, vol. 54, no. 3, pp. 463–474, May 2018.

[7] J. Zhang, X. Tao, L. Tan, J. C.-W. Lin, H. Li, and L. Chang, *On Link Stability Detection for Online Social Networks*, vol. 1. Cham, Switzerland: Springer, 2018, pp. 320–335, doi: 10.1007/978-3-319-98809-2_20.

[8] X. Zhao and W. Li, "Trend prediction of event popularity from microblogs," *Future Internet*, vol. 13, no. 9, p. 220, Aug. 2021.

[9] G. Li, Y. Liu, B. Ribeiro, and H. Ding, "On new group popularity prediction in event-based social networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 3, pp. 1239–1250, Jul. 2020.

[10] U. Mehmood, I. Moser, and N. Ronald, "Event attendance prediction using social media," in *Proc. Australas. Comput. Sci. Week Multiconference*, Feb. 2020, pp. 1–7.

[11] J. G. Lee, S. Moon, and K. Salamatian, "Modeling and predicting the popularity of online contents with Cox proportional hazard regression model," *Neurocomputing*, vol. 76, no. 1, pp. 134–145, 2012.

[12] S. Gao, J. Ma, and Z. Chen, "Modeling and predicting retweeting dynamics on microblogging platforms," in *Proc. 8th ACM Int. Conf. Web Search Data Mining*, Feb. 2015, pp. 107–116.

[13] F. Yin, J. Wu, X. Shao, and J. Wu, "Topic reading dynamics of the Chinese sina-microblog," *Chaos, Solitons Fractals, X*, vol. 5, Mar. 2020, Art. no. 100031. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2590054420300129

[14] F. Yin, H. Pang, X. Xia, X. Shao, and J. Wu, "COVID-19 information contact and participation analysis and dynamic prediction in the Chinese sina-microblog," *Phys. A, Stat. Mech. Appl.*, vol. 570, May 2021, Art. no. 125788, doi: 10.1016/j.physa.2021.125788.

[15] X. G. Chen, S. Duan, and L. D. Wang, "Research on trend prediction and evaluation of network public opinion," *Concurrency Comput., Pract. Exper.*, vol. 29, no. 24, pp. 1–9, 2017.

[16] M. T. Uddin, M. J. A. Patwary, T. Ahsan, and M. S. Alam, "Predicting the popularity of online news from content metadata," in *Proc. Int. Conf. Innov. Sci., Eng. Technol. (ICISET)*, Oct. 2016, pp. 1–5.

[17] C. Castillo, M. El-Haddad, J. Pfeffer, and M. Stempeck, "Characterizing the life cycle of online news stories using social media reactions," in *Proc. 17th ACM Conf. Comput. Supported Cooperat. Work Social Comput.*, Feb. 2014, pp. 211–223.

[18] M. Ahmed, S. Spagna, F. Huici, and S. Niccolini, "A peek into the future: Predicting the evolution of popularity in user generated content," in *Proc. 6th ACM Int. Conf. Web Search Data Mining (WSDM)*, 2013, pp. 607–616.

[19] G. Szabo and B. A. Huberman, "Predicting the popularity of online content," *Commun. ACM*, vol. 53, no. 8, pp. 80–88, 2010.

[20] Y. Shang, B. Zhou, X. Zeng, Y. Wang, H. Yu, and Z. Zhang, "Predicting the popularity of online content by modeling the social influence and homophily features," *Frontiers Phys.*, vol. 10, pp. 1–11, Jul. 2022.

[21] H. Dou, W. X. Zhao, Y. Zhao, D. Dong, J.-R. Wen, and E. Y. Chang, "Predicting the popularity of online content with knowledge-enhanced neural networks," in *Proc. KDD*, 2018. [Online]. Available: https://www.kdd.org/kdd2018/files/deep-learning-day/DLDay18_paper_8.pdf

[22] I. N. Lymperopoulos, "Predicting the popularity growth of online content: Model and algorithm," *Inf. Sci.*, vol. 369, pp. 585–613, Nov. 2016, doi: 10.1016/j.ins.2016.07.043.

[23] T. Trinh, D. Wu, R. Wang, and J. Z. Huang, "An effective content-based event recommendation model," *Multimedia Tools Appl.*, vol. 80, no. 11, pp. 16599–16618, May 2021, doi: 10.1007/s11042-020-08884-9.

[24] H. Yin, L. Zou, Q. V. H. Nguyen, Z. Huang, and X. Zhou, "Joint event-partner recommendation in event-based social networks," in *Proc. IEEE 34th Int. Conf. Data Eng. (ICDE)*, Apr. 2018, pp. 929–940. [Online]. Available: https://ieeexplore.ieee.org/document/8509309/

[25] X. Wang, B. Fang, H. Zhang, and S. Su, "Predicting the popularity of online content based on the weak ties theory," in *Proc. IEEE 3rd Int. Conf. Data Sci. Cyberspace (DSC)*, Jun. 2018, pp. 386–391.

[26] A. K. Bhowmick, S. Pramanik, S. Pathak, and B. Mitra, "On the role of micro-categories to characterize event popularity in meetup," in *Proc. ICWSM*, 2021, pp. 71–82.

[27] X. Chen, X. Zhou, J. Chan, L. Chen, T. Sellis, and Y. Zhang, "Event popularity prediction using influential hashtags from social media," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 10, pp. 4797–4811, Oct. 2020.

[28] S. Madisetty and M. S. Desarkar, "Social media popularity prediction of planned events using deep learning," in *Advances in Information Retrieval* (Lecture Notes in Computer Science), vol. 12657, D. Hiemstra, M. F. Moens, J. Mothe, R. Perego, M. Potthast, and F. Sebastiani, Eds. Cham, Switzerland: Springer, 2021, doi: 10.1007/978-3-030-72240-1_31.

[29] S. L. Salzberg, "C4.5: Programs for machine learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993," *Mach. Learn.*, vol. 16, no. 3, pp. 235–240, 1994, doi: 10.1007/BF00993309.

[30] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, pp. 273–297, Apr. 1995.

[31] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[32] T. Trinh, D. Wu, and J. Z. Huang, "C3C: A new static content-based three-level web cache," *IEEE Access*, vol. 7, pp. 11796–11808, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8611139/

THANH TRINH received the Ph.D. degree in computer science from Shenzhen University, China, and the M.Sc. degree in information systems design from the University of Central Lancashire, U.K. He is currently a Lecturer with the Faculty of Computer Science, Phenikaa University. He has published many papers on his research topic. His research includes efficient query, database, social networks, classification, forecasting disasters, and climate change.

NHUNG VUONGTHI received the M.Sc. degree in information systems design from the University of Central Lancashire, U.K. She is currently a Lecturer with the Faculty of Digital Technologies and Cybersecurity, School of Business and Management, Vietnam National University. She has conducted several project consultation in her research topic. Her research interests include cybersecurity, data mining, and network optimization.