Information Sciences 525 (2020) 82–92

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Time sensitivity-based popularity prediction for online promotion on Twitter

Chunjing Xiao[a,b], Chun Liu[b], Ying Ma[c,*], Zheng Li[b], Xucheng Luo[a]

[a] *School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China*
[b] *Henan Key Laboratory of Big Data Analysis and Processing, Henan University, Kaifeng 475004, China*
[c] *School of Computer and Information Engineering, Xiamen University of Technology, Xiamen, 361024, China*

ARTICLE INFO

Article history:
Received 28 April 2018
Revised 1 March 2020
Accepted 20 March 2020
Available online 24 March 2020

Keywords:
Social networks
Popularity prediction
Latent factor model
Social media strategy

ABSTRACT

Currently, companies and individuals tend to use social media to publish information for promoting products and options. Although an increasing body of research has focused on promotional skills on social media, the role of post-publication time is rarely studied. However, publication time of a post plays an important role in its popularity. To select suitable publication times, an effective approach is to predict popularity values of a post when it is published at a series of future time points. However, this task is not trivial because, (*i*) except for publication time, all the features are inefficient, as they are the same, and (*ii*) the new model needs to output multiple popularity values for each input post. To address these challenges, we introduce a latent factor model to build a time sensitivity-based predictive model that can predict posts' popularity values when they are published at various times. In this model, to alleviate data sparsity, we decompose posts into syntactic units that are derived from dependency parsing results. To take advantage of auxiliary information, we exploit the features of temporal information and neighborhood influence. Experiments with Twitter data demonstrate that the proposed model significantly outperforms state-of-the-art methods.

1. Introduction

Information-sharing activities on social media have become increasingly widespread, and companies and individuals have realized the importance of promoting information via social media networks. At the same time, these companies and individuals are confronted with the problem of how to obtain the full benefit that the platforms provide [1,2]. Hence, social media strategies for enhancing online promotion are becoming increasingly important. Correspondingly, a number of studies have attempted to present detailed skills to help users improve the promotional effectiveness of social media, such as how to use different social media sites for various tasks [3] and control the number of repeated posts [4]. However, most studies ignore the impact of posts' publication time on promotional effectiveness.

For social media on which posts have a short lifespan (such as Twitter), posts' publication times play an important part in their promotional effectiveness (i.e., popularity). First, the final popularity of posts depends mainly on their initial popularity during a short period after publication [5], whereas posts' initial popularity varies over time [6]. On Twitter, posts generally obtain the highest number of responses (i.e., retweets/replies) in the first hour after their publication, and then

[*] Corresponding author.
E-mail addresses: chunjingxiao@gmail.com (C. Xiao), liuchun@henu.edu.cn (C. Liu), maying@xmut.edu.cn (Y. Ma).

https://doi.org/10.1016/j.ins.2020.03.056
0020-0255/© 2020 Elsevier Inc. All rights reserved.

the number dramatically decreases in subsequent hours. In fact, the number of responses in the second hour accounts for only approximately 15% of that number in the first hour [5]. At the same time, depending on the time of day when posts are published, the average number of retweets that posts receive in the first hour after publication differs greatly on microblogging platforms [6]. Second, posts' popularity is highly correlated with users' activity levels when they are published [4], and the activity levels fluctuate widely over time [7]. Studies found that promotions generated when users are active can be very effective [4], and users' activity levels at various times are obviously different [7]. It should be noted that although Twitter is used around the world, activity levels of user followers can still exhibit broad fluctuations because the majority of followers always pay more attention to content from their countries. Existing studies showed that users tend to communicate with people in close geographic proximity to themselves [8], and users preferentially connect and exchange information with other users from their country [9]. Based on these analyses, we can conclude that publication time can substantially influence post popularity. Therefore, selecting suitable publication times for posts can effectively improve their promotional effectiveness.

To select publication times, an intuitive strategy is to publish posts as early as possible. However, the advantage of earlier publication on social media is not obvious. Tan *et al.* [10] showed that, for two posts with the same content, the earlier one might enjoy a first-mover advantage because it is the original; alternatively, the later one might be preferred because retweeters consider the first one to be stale. Hence, instead of publishing posts as early as possible, a better method for selecting suitable publication times for posts is needed.

An effective method for selecting appropriate times is to predict posts' popularity values when they are published at a series of future time points. Based on these values, users can effectively determine their publishing strategies. Currently, a variety of methods for predicting popularity have been proposed and have achieved remarkable success. However, they are not well-suited for selecting posts' publication times. First, except for publication time, all of the features are inefficient because they are the same in this task. For this task, multiple popularity values at various publication times must be predicted for a given post. The features, such as its author and content information, are the same, and only the publication time is different. In contrast, existing methods are typically designed to predict the popularity of different posts, for which the features from different items can be various and effective for their goals. However, if only one feature (publication time) is different, they may suffer from distinct performance degradation. Second, for an input sample, the number of output values is different. For an input sample, this task needs to output multiple popularity values at a time. However, existing methods for popularity prediction generally aim at outputting one value for one test item (e.g., a post). It is time consuming and inefficient to predict one popularity value for each time point and perform multiple predictive tasks for various time points.

To address these challenges, we propose a new method, a time sensitivity-based predictive model, to predict posts' popularity values when they are published at various times. More specifically, we introduce the Latent Factor Model (LFM) for our prediction problem. We build a matrix in which the rows and columns represent posts and publication times, respectively. The unknown items (popularity values) of new posts can be determined using matrix factorization techniques. This model fully considers the time factor because publication time is regarded as one of the two features. And it can also predict posts' popularity values when they are published at a series of future time points. However, the basic LFM suffers from the problems of data sparsity and lack of auxiliary information. To alleviate the data sparsity problem, we decompose posts into syntactic units (primarily noun and verb units) that are identified from word dependency relations. The motivation behind the syntactic units is based on the linguistic analysis characteristic that the main elements in textual sentences are nouns and verbs [11]. To take advantage of auxiliary information, we exploit two temporal features: post density and audience activity level. The former refers to the number of posts that have been published by users within the past hour, and the latter refers to the activity level of the audience (readers of text and viewers of videos embedded in posts) at the publication time of the posts. In addition, we consider for the model the influence of semantically similar posts, called the neighborhood influence, because audiences of posts with similar topics typically exhibit similar characteristics [12]. As a result, the three factors, syntactic units, temporal information, and neighborhood influence, are incorporated into the basic LFM to predict the popularity values at a series of future time points.

Based on two types of data sets from Twitter, we compare our proposed model with an array of baselines in terms of predictive performance. The results demonstrate that our model significantly outperforms the baselines. In addition, we conduct extensive experiments to evaluate the effectiveness of incorporating syntactic units, temporal information, or neighborhood influence into the basic model.

We summarize the main contributions of this paper as follows.

- We propose the problem of selecting suitable publication times for online promotion, which differs from existing studies in two aspects: (1) except for publication time, features such as user and content information are inefficient because they are the same, and (2) this task requires multiple outputs about popularity values for one input sample.
- We develop a time sensitivity-based predictive model. In this model, we decompose posts into syntactic units to alleviate the data sparsity of the basic LFM and exploit the post density, user activity level, and neighborhood influence features.
- Based on a real Twitter dataset, experiments in two scenarios demonstrate that our model outperforms state-of-the-art methods and proves the effectiveness of the post decomposition and proposed features.

2. Related work

This study is related to three research areas: exploring promotional strategies via social media, analyzing the impact of publication time on popularity, and predicting content popularity on social media. Next, we present an overview of the most closely related works in each area and highlight the major differences between our study and these works.

Exploring promotional strategies. In recent years, substantial attention has been paid to promotional methods for increasing engagement (such as the number of retweets on Twitter) on social media. For example, Tan *et al.* [10] studied the effects of wording during information propagation. They extracted many pairs of tweets that contain the same URL, use different words and are from the same user, and investigated whether a different choice of words affects the message popularity under controlling for the user and topic. Adamopoulos *et al.* [13] analyzed the effects of promotional skills via social media, quantified the influence of various promotional characteristics, and determined the types of promotional approaches that are effective. Kim *et al.* [14] examined the correlation between engagement on social media and content-oriented variables and found that content creation strategies are not universal and, instead, should depend on organization type. Kim *et al.* [15] studied the effectiveness of promoting products on Twitter and presented a new method to gauge the effect of crowdsourced promotional methods. Xue *et al.* [16] proposed a multi-task learning model to help users select proper publication platforms for their posts. These studies provided remarkable skills from various aspects for online promotion via social media. Our work also aims to assist users in obtaining greater engagement; however, we study promotional methods in terms of the aspect of publication time.

Analyzing the impact of publication time. A few researchers have analyzed the impact of the publication times of messages on their popularity. Szabo *et al.* [17] found that, depending on the time of day at which a submission is made to Digg, stories differ substantially in the number of initial diggs received a few hours after publication. They predicted the popularity at later times using data from earlier times based on a high log-linear correlation between the data at later and earlier times. To ignore the dependence of publication time, they introduced the notion of digg time, which is not measured in seconds but rather in the number of diggs that users cast on promoted stories. Gao *et al.* [6] and Canneyt *et al.* [18] also found that the initial popularity of a message depends strongly on the time of day at which it is published on Weibo and Facebook, respectively. To accurately predict popularity, they proposed time transformation methods to eliminate the effect of publication time. Sabate *et al.* [19] analyzed factors that impact the popularity of branded content on Facebook fan pages and demonstrated that posts published during business hours tend to receive more comments. In contrast, liking activity is not influenced by posting time. Kuang *et al.* [4] analyzed the effectiveness of promotion strategies on microblogs and demonstrated that generated promotions are more effective when users are active. These analyses indicate the importance of publication time for post popularity, which provides an important basis for our study. However, instead of analyzing the correlation between publication time and popularity, we attempt to develop a model to predict the popularity of posts when they are published at various times and aim to assist users in selecting suitable publication times.

Popularity prediction. Studies on popularity prediction can be divided into two main categories: prediction before and after post-publication. When predicting popularity before publication, predicting accurate values of popularity is very unstable because the distribution of popularity is strongly skewed [20,21]. As a result, instead of predicting the precise popularity value, researchers typically predict a probability range. For example, Hong *et al.* [22] defined a few genres for representing posts' popularity levels. They investigated a wide range of features, such as content features, graph topological features, temporal features, and metadata features, and used logistic regression to classify post genres. Jenders *et al.* [23] predicted whether posts can be reshared more frequently than a specified threshold. They exploited both obvious features (such as the number of followers, tweet length, and number of hashtags) and latent features (such as the sentiment and emotional divergence in tweets) and adopted linear regression and Naive Bayes to make predictions. Vasconcelos *et al.* [24] exploited content, temporal, spatial, topical, and social aspects to investigate the predictability of comment popularity and predicted the popularity levels of comments using multivariate linear regression and support vector machine.

To obtain better predictive performance, many studies made predictions after post-publication. In this scenario, the early retweet or view number within a short period after post-publication is adopted to predict popularity. Some studies still predict a range of popularity. For example, Cheng *et al.* [25] used multiple models, such as logistic regression and SVM, to predict whether the cascade of a post can reach size $2t$ when it currently has size t. Vallet *et al.* [26] explored a suite of features based on early popularity and user information and adopted a gradient boosted decision tree to predict whether videos on YouTube will be popular. Li *et al.* [27] proposed a concept drift-based mechanism for classifying the retweet number and popularity score for social multimedia. Han *et al.* [28] exploited multiple features, such as deep image information, image meta, and initial propagation pattern, and predicted whether an image on Pinterest will be popular in terms of the cascade volume or viral in terms of the structural virality using the random forest method.

Other works utilized early information to predict the accurate values of popularity. For example, Zhao *et al.* [29] developed a self-exciting point process model to predict tweet popularity. They model the self-exciting mechanism of retweets at an earlier stage to predict the final number of retweets. Mishra *et al.* [30] presented a model that combines feature-driven and point process methods to accurately predict the number of retweets. Rizoiu *et al.* [31] built the Hawkes intensity process to predict content popularity. This model links exogenous inputs from Twitter and endogenous responses within YouTube and can significantly improve predictive performance.

The problem of selecting suitable publication times can be solved through popularity prediction. Only methods that perform prediction before post-publication are appropriate for this task because the early number of retweets after publication

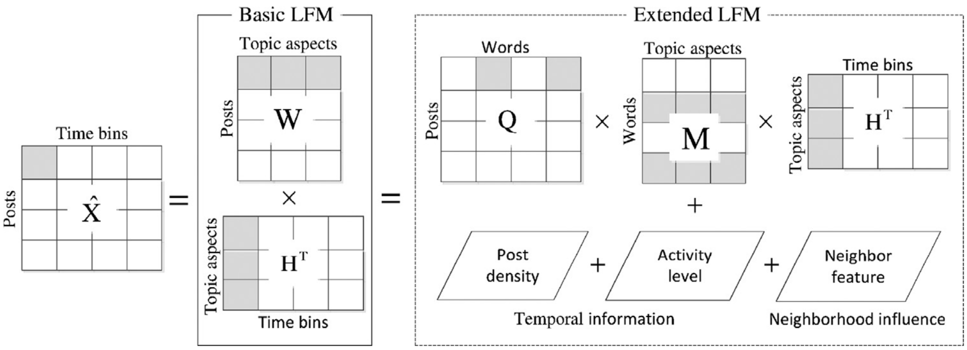


Fig. 1. Our extended Latent Factor Model.

is not available. Studies on prediction before post-publication generally extracted various hand-crafted features and utilized standard regression or classification methods for prediction. These studies achieved valuable results in predicting popularity but do not aim to assist users in selecting suitable publication times for posts and, correspondingly, do not stress the importance of publication time in predictive models. When applied to our task, such methods have poor predictive performance because, except for publication time, all of the features are the same and inefficient for our task. Moreover, these methods typically output one predictive value for each input sample; however, our task requires multiple output values for each input sample.

3. Predictive method

To provide insight into selecting suitable publication times for users, we propose a model for predicting the popularity of posts when they are published at various times. Specifically, we first present the basic LFM for our problem, which can capture the time sensitivity and predict a series of popularity values at a time. Next, to alleviate data sparsity, the basic LFM is extended by decomposing the posts into syntactic units. Then, we exploit the temporal and neighborhood features for the model to improve predictive performance. An overview of our model is presented in Fig. 1.

3.1. Basic approach

Let $P = \{p_1, p_2, \cdots, p_{|P|}\}$ be the set of posts and $T = \{t_1, t_2, \cdots, t_{|T|}\}$ be the set of time bins. Here, we discretize continuous time by dividing time into equally sized bins, such as every 30 minutes or every 60 minutes. For any $p \in P$ and its publication time bin t, we observe a ground-truth popularity x_{pt}. Hence, x_{pt} denotes the observed popularity of post p that was published in time bin t. Note that x_{pt} can be the exact number of retweets or binary value (indicating whether the number of retweets exceeds a specified threshold). Additionally, we only consider the final values of popularity because of the short lifespan of tweets, which is generally less than two days [5].

Then, each training instance can be represented by a tuple (x_{pt}, p, t), which is organized into a sparse matrix X of size $|P| \times |T|$ using (p, t) as the index and x_{pt} as the entry value. The objective is to predict popularity $\hat{x}_{pt}$ for the new posts and time bins. For the LFM, the primary idea is to map both posts and time bins into a joint latent factor space with dimension f, and the popularity values are modeled as inner products in that space. That is, the LFM attempts to approximate X by the product of two low-rank latent factor matrices that are denoted as W: $|P| \times f$ and H: $|T| \times f$, where f is a hyperparameter referring to the rank of the approximation and, in general, $f \ll \min\{|P|, |T|\}$. Therefore, each observed value $\hat{x}_{pt}$ in X can be approximated by the product of two components:

$$\hat{x}_{pt} = W_p \cdot H_t^T \tag{1}$$

where W_p is the post feature vector and H_t is the time feature vector. These vectors are regarded as topic factors of posts and time in f-dimensional space. That is, W_p and H_t indicate the affinities of post p and time t, respectively, toward the f topics. However, the posts may have a certain degree of biases: some posts are prone to achieving high popularity, whereas others are not. Similarly, time bins may also have some degree of biases. Generally, considering the biases in the LFM can yield a superior generalization [32]. Therefore, these biases are incorporated into the model, and the formula becomes:

$$\hat{x}_{pt} = \mu + b_p + b_t + W_p \cdot H_t^T \tag{2}$$

where μ represents the average popularity of all of the posts in the dataset, and b_p and b_t indicate the deviations of post p and time t, respectively, from the value of μ. Learning the unknown parameters, b_p, b_t, W_p, and H_t, is an optimization problem of minimizing the regularized squared error on the known training set.

$$\min_{b_p, b_t, W_p, H_t} \sum_{p,t} (x_{pt} - \hat{x}_{pt})^2 + \lambda_1(\|b_p\|^2 + \|b_t\|^2) + \lambda_2(\|W_p\|^2 + \|H_t\|^2) \tag{3}$$

where λ_1 and λ_2 are regularization parameters used to avoid overfitting.

This LFM fully considers publication time because it is one of the two features for the prediction. Hence, the LFM can capture the sensitivity of the published time. In addition, it can output a series of popularity values in various publication time bins for a post. However, the direct application of the LFM has two main drawbacks: (*i*) Each post is treated as a single item, and most rows (posts) have only one value because they are published only once. Thus, there are very few entry values and, accordingly, matrix X is typically very sparse, which may result in poor factorization efficiency. (*ii*) The LFM considers only posts and publication time information and ignores other features that can actually enhance the predictive performance.

3.2. Incorporating syntactic units

To overcome the sparsity issue, we present an extended model that decomposes content to a finer-grained level. Generally, the bag of words (BOW) of posts is adopted to enrich the features of predictive models. Thus, we decompose the post into a series of words. Supposing that the vocabulary K consists of all words, W can be decomposed as $W = Q \cdot M$. Here, Q with size $|P| \times |K|$ refers to the post-word matrix, whose p-th row refers to the existence of words in the p-th post, and M with size $|K| \times f$ refers to the estimated word-topic factor matrix that maps each word k to the feature vector M_k. Hence, the estimated item can be reformulated as:

$$\hat{x}_{pt} = \mu + b_p + b_t + (Q_p \cdot M) \cdot H_t^T \tag{4}$$

The matrix Q will become dense through this word-level decomposition; however, BOW can be rather coarse for the prediction. Two main problems should be considered: (*i*) words with different parts of speech probably have different weights in reflecting the topic factor. Typically, nouns and verbs are the main elements in textual sentences and can describe the topics concerned [11]. Thus, these words should play a more important role in the model. (*ii*) Some non-nouns and non-verbs may require special treatment. For example, intensifiers, such as really and suddenly, can strengthen the topic meaning and should be modeled together with the nouns and verbs that they modify. To this end, we identify the dependency relations of words in the posts and extract noun and verb units for the model.

In particular, we resort to the Stanford Neural Network Dependency Parser[1] to parse the posts and obtain a series of pairwise units. One issue is that one word might appear in multiple pairwise units obtained from dependency parsing, and some units might not be useful. Hence, we use the parts of speech of words to extract noun and verb units and adopt three rules to extract word units: (*i*) units including nouns are kept as noun units, and other units, including verbs, are kept as verb units; (*ii*) when a noun or verb appears in multiple units, only the first unit is selected, and the others are removed; and (*iii*) the units without nouns or verbs are removed because they are usually not topical.

Based on the extracted syntactic units, we decompose the latent factors of a post into a weighted combination of words that exist in the two types of units. The prediction formula is as follows:

$$\hat{x}_{pt} = \mu + b_p + b_t + \left(\sum_{n \in N_p} \frac{\alpha}{|N_p|} \sum_{k \in n} M_k + \sum_{v \in V_p} \frac{\beta}{|V_p|} \sum_{k \in v} M_k \right) \cdot H_t^T \tag{5}$$

where N_p and V_p are the sets of noun and verb units, respectively, in post *p;* M_k refers to the vector of the word k; and α and β are the parameters used to individually adjust the weights of words in noun units and verb units. The corresponding objective function is presented in Equation 9 of Table 1.

3.3. Incorporating temporal information

For our prediction problem, all of the information except for the publication time is the same. Therefore, to improve predictive performance of the model, we exploit two temporal features based on publication time: the post density and audience activity level.

Typically, publishing too many posts within a short period might result in information overload [33,34]. This overload may lead to the lower average popularity of posts. For example, if users publish too many posts within a short period, the large amount of information will be beyond the audience's receptive ability, and they will skip some posts. Therefore, the number of tweets in past hours must be considered to improve predictive performance. Thus, we exploit the post density feature, which refers to the number of posts published by the user during the past hour. Suppose that D_p and $\bar{D}$ denote the post density of post p and the average post density of all posts published by the user, respectively. Hence, the post density bias can be computed as $(D_p - \bar{D})$.

Another factor to consider is the audience's activity level at publication time. Generally, the audience's activity level may vary with time, which affects posts' popularity [4,7]. Here, for a given user and time bin, the number of posts published by its followers within this bin is used to represent the audience's activity level, and the publication time needs to be converted to the local time. Because obtaining the activity level in real time is difficult and users' activity levels exhibit clear weekly patterns [35], we use the historical activity level to represent the real-time activity level. Specifically, we regard one week

[1] https://nlp.stanford.edu/software/nndep.shtml.

as a cycle and divide the time per cycle into equal-sized bins. For example, if a time bin consists of 60 minutes, there are 24 bins per day and 7*24 bins per cycle. Then, we regard the average number of posts in a given bin from all historical cycles as the activity level. Thus, for a given post p, if we learn its publication time bin, its activity level, A_p, can be computed using the historical data of all of the cycles in this bin. We can further compute the average activity level per post, $\bar{A}$. As a result, the activity level bias can be formulated as $(A_p - \bar{A})$.

By considering post density and audience activity level, we extend the basic predictor as follows:

$$\hat{x}_{pt} = \mu + b_p + b_t + w_d(D_p - \bar{D}) + w_a(A_p - \bar{A}) + W_p \cdot H_t^T \tag{6}$$

where w_d and w_a are two learnable parameters for adjusting the weights. The objective function is presented in Equation 10 of Table 1, where λ_3 is a newly introduced regularization parameter, similar to λ_1 and λ_2.

3.4. Incorporating neighborhood influence

On social media, a common phenomenon is that posts with similar topics have audiences with similar characteristics [12,36]. Therefore, we exploit the influence of the neighborhood for the predictive model. Here, the neighborhood refers to semantically similar posts from the same author and the same publication time bin.

First, we must identify posts that are similar to the given post. To measure the similarity between posts, we introduce the word mover's distance (WMD) proposed by Kusner *et al.* [37]. The WMD measures the dissimilarity between two text documents as the minimum distance that the embedded words of one document need to travel to reach the embedded words of another document. A shorter distance indicates greater similarity between the two documents. The WMD outperforms the other baselines on Twitter data [37].

For a post p, we use the WMD to select its top k similar posts with the same author and the same time bin. The neighborhood feature can be combined with the basic model additively:

$$\hat{x}_{pt} = \mu + b_p + b_t + |S(p)|^{-\frac{1}{2}} \sum_{j \in S(p)} w_{pj}(x_{jt} - \bar{x}_p) + W_p \cdot H_t^T \tag{7}$$

where $S(p)$ is the set of posts similar to post p, x_{jt} is the popularity of post j when published at time t, $\bar{x}_p$ refers to the average popularity of the posts, and w_{pj} is the parameter that is learned during matrix factorization [38]. The objective function is given in Equation 11 of Table 1.

3.5. Overall model

Now, we incorporate all of the factors, syntactic units, temporal information, and neighborhood influence, into the basic model. Then, the formula becomes:

$$\begin{aligned} \hat{x}_{pt} = \mu + b_p + b_t &+ \left(\sum_{n \in N_p} \frac{\alpha}{|N_p|} \sum_{k \in n} M_k + \sum_{v \in V_p} \frac{\beta}{|V_p|} \sum_{k \in v} M_k \right) \cdot H_t^T \\ &+ w_d(D_p - \bar{D}) + w_a(A_p - \bar{A}) + |S(p)|^{-\frac{1}{2}} \sum_{j \in S(p)} w_{pj}(x_{jt} - \bar{x}_p) \end{aligned} \tag{8}$$

The corresponding objective function is presented in Equation 12 of Table 1.

This objective function is convex, the minimum can be searched by stochastic gradient descent [39], and all of the variables in the parameter space $\{b_p, b_t, M_k, H_t, w_d, w_a, w_{pj}\}$ can be estimated automatically. λ_1, λ_2, λ_3, and λ_4 are regularization values to avoid overfitting and can be tuned using the training set. The other objective functions (e.g., Equations 9, 10 and 11) have similar forms and can be estimated in similar manners.

Given a new post p, the popularity when it is published at time t, $\hat{x}_{pt}$, can be predicted using Eq. 8. To predict the binary value that indicates whether the number of retweets exceeds a specified threshold, we use sigmoid $\delta(x) = 1/(1 + exp(-x))$ to convert the predicted value to a distribution between 0 and 1. If $\delta(\hat{x}_{pt})$ is less than 0.5, we label the result as negative; otherwise, it is positive.

Table 1
Objective functions for incorporating syntactic units, temporal information, neighborhood influence and all the factors.

$\min_{b_p,b_t,M_k,H_t} \sum_{p,t}(x_{pt} - \hat{x}_{pt})^2 + \lambda_1(\|b_p\|^2 + \|b_t\|^2) + \lambda_2(\sum_{k \in K}\|M_k\|^2 + \|H_t\|^2)$	(9)
$\min_{b_p,b_t,W_p,H_t,w_d,w_a} \sum_{p,t}(x_{pt} - \hat{x}_{pt})^2 + \lambda_1(\|b_p\|^2 + \|b_t\|^2) + \lambda_2(\|W_p\|^2 + \|H_t\|^2) + \lambda_3(\|w_d\|^2 + \|w_a\|^2)$	(10)
$\min_{b_p,b_t,W_p,H_t,w_{pj}} \sum_{p,t}(x_{pt} - \hat{x}_{pt})^2 + \lambda_1(\|b_p\|^2 + \|b_t\|^2) + \lambda_2(\|W_p\|^2 + \|H_t\|^2) + \lambda_4(\sum_{j \in S(p)}\|w_{pj}\|^2)$	(11)
$\min_{b_p,b_t,M_k,H_t,w_d,w_a,w_{pj}} \sum_{p,t}(x_{pt} - \hat{x}_{pt})^2 \quad + \lambda_1(\|b_p\|^2 + \|b_t\|^2) + \lambda_2(\sum_{k \in K}\|M_k\|^2 + \|H_t\|^2) + \lambda_3(\|w_d\|^2 + \|w_a\|^2) + \lambda_4(\sum_{j \in S(p)}\|w_{pj}\|^2)$	(12)

4. Experimental evaluation

We now conduct experiments on Twitter data to evaluate the proposed model and compare it with state-of-the-art baselines. The data and code are available online[2].

4.1. Data set

We evaluate our model by applying it to two scenarios. One scenario is predicting which is more popular for tweets that are repeatedly published by the same user, called *Repeated data*. The other scenario is predicting whether the retweet number of a tweet can exceed a given threshold, called *Random data*. We first collect a large set of basic data and then present how to extract corresponding data for the two scenarios.

To study the popularity of posts with different publication times, we collect a large number of tweets on Twitter in the following manner. First, from a large volume of users in our previous work [40], we filter out the users whose languages are not English, whose time zone fields are empty, or whose follower numbers are less than ten thousand. The main reasons for removing these users are that sentences must be written in English for decomposition, the publication time must be converted to the local time, and predicting the post popularity of users with fewer followers is of low value and difficult because their posts generally obtain very few retweets (e.g., 0 or 1). From the remaining users, we randomly select 1500 users for our experiments. Then, we collect the tweets of these users during the two months from October 1, to November 30, 2017. We remove the special characters and stop words and eliminate tweets with fewer than five words. Because tweet texts are typically informal and noisy, they are lexically normalized and corrected in preprocessing using the method in Han *et al.* [41]. From these basic data, we further extract two data sets for the two scenarios:

Repeated data: This data set consists of repeated tweets. Here, two tweets are regarded as the repeated one if they are published by the same user, and the only difference between them is spacing. Based on this definition, from the basic data, we extract approximately 23,000 pairs of repeated tweets for this experiment. The repeated tweets from October 1, to November 20, are used as the training set, and the data from the remaining 10 days are used as the test set. For a pair of repeated tweets in this data set, we predict which tweet is more popular. Because the only difference between the repeated tweets is the publication time, we can more clearly observe the effect of publication time through this experiment. Note that the popularity of repeated tweets might exhibit a temporal bias because of the chronological order of tweet presentation: the first published tweets might enjoy a first-mover advantage because it is the original; alternatively, subsequently published tweets might be preferred because retweeters consider the first one to be stale. However, no obvious bias is observed by Tan *et al.* [10]. Therefore, we ignore this bias in the experiment.

Random data: This data set is obtained by removing repeated tweets, retweets, and replies from the basic data set. As a result, we obtain approximately 323,000 tweets for this scenario. In this experiment, we predict whether the retweet number of a tweet exceeds a threshold. Here, the average number of retweets during a specified period is used as the threshold. We take the data from the first 21 days as the basic set, which is used to compute the threshold; those of the following 30 days as the training set; and those of the last 10 days as the test set. This prediction of whether the number of retweets will exceed the threshold can provide simple and intuitive results for users to facilitate the selection of a suitable publication time.

4.2. Experimental setting

Parameter setting: We set the super parameters empirically by performing a 5-fold cross-validation on the training set. We also optimize the number of factors, *f*, by performing a grid search on the values between 1 and 100. The accuracy is not sensitive to *f*, except at a few small numbers (e.g., 5 and 10). Thus, we set $f = 60$ because it can obtain a slightly better result. We tune λ using the training set and set all $\lambda = 0.004$ since we find that varying them influences the results only slightly. In the iteration, we set the learning rate to 0.002. The word weights of the noun and verb units in Eq. 5 and 8 are set to $\alpha = 2$ and $\beta = 1$. For the neighborhood number and time segmentation, by default, our proposed model uses the 6 nearest neighbors for each post and 60 minutes for each time bin.

Evaluation metric: Considering the imbalanced nature of our dataset, we adopt the geometric mean (GM) [42,43] as the evaluation metric, which is defined as $GM = \sqrt{Sensitivity \times Specificity}$. Here, Sensitivity (SEN) refers to the true positive rate, and Specificity (SPE) refers to the true negative rate.

4.3. Comparison of different methods

Because the early number of retweets, which is used for prediction after post-publication, is not available for our task, we compare our proposed model with three widely used methods for popularity prediction before post-publication [21] and other applications [44,45]: logistic regression (**Logistic**), J48 decision tree (**J48**) and support vector machine (**SVM**), and an advanced method, convolutional neural network (**CNN**). Here, SVM is implemented using libsvm with the radial basis

[2] https://github.com/ChunjingXiao/TwitterPopularity.

Table 2
Comparison of different approaches.

DataSet	Metric	Logistic	J48	SVM	CNN	CNN-Excl	Ours
Repeated Data	SEN	0.6199	0.5469	0.5634	0.5482	0.5687	**0.6343**
	SPE	0.5330	0.5582	0.6171	0.6658	0.6143	**0.7373**
	GM	0.5748	0.5525	0.5896	0.6041	0.5911	**0.6839**
Random Data	SEN	0.6127	0.5424	0.6127	0.5953	0.5564	**0.7269**
	SPE	0.5514	0.5937	0.5714	0.6573	0.6674	**0.6890**
	GM	0.5812	0.5675	0.5917	0.6255	0.6094	**0.7077**

Table 3
Comparison of different model configurations.

DataSet	Metric	Basic	BOW	Syntactic	Temporal	Neighbor	Full
Repeated Data	SEN	0.5374	0.6250	**0.6654**	0.5402	0.5513	0.6343
	SPE	0.6397	0.6893	0.6843	0.6706	0.6850	**0.7373**
	GM	0.5863	0.6564	0.6748	0.6019	0.6145	**0.6839**
Random Data	SEN	0.5288	0.5805	0.6626	0.5660	0.6030	**0.7269**
	SPE	0.6110	**0.7672**	0.7337	0.6461	0.6702	0.6890
	GM	0.5684	0.6674	0.6972	0.6047	0.6357	**0.7077**

function (RBF) kernel. For CNN, we use the code provided in [46] and employ the same setting. To use these methods, we extract the available features from [40], including the publication time, user, and content information. We also add the BOW features and the two temporal features (post density and audience activity level).

The predictive accuracies on the test sets for the two scenarios are listed in Table 2. The best results for each metric are emphasized in boldface. Clearly, we observe that our method significantly outperforms the baselines for all of the metrics. More strikingly, in terms of GM, our method achieves a 23% improvement over the lowest baseline (*J48*) and 13% over the best one (*CNN*) on Repeated data. Our method also achieves a 24% improvement over *J48* and 13% over *CNN* on Random data. Hence, our model can effectively improve prediction performance by capturing the time sensitivity and taking advantage of the auxiliary information.

In addition, we perform the prediction for the four baselines when excluding our proposed temporal features to evaluate their effectiveness. All of the results are less accurate than the ones including the temporal features. Here, we report only the best one, CNN excluding the temporal features (**CNN-Excl**), in Table 2. This table shows that *CNN* with our proposed temporal features outperforms *CNN-Excl* for both Repeated data and Random data. These results indicate that our proposed temporal features are effective for different models at improving the predictive performance.

4.4. Comparison of different configurations

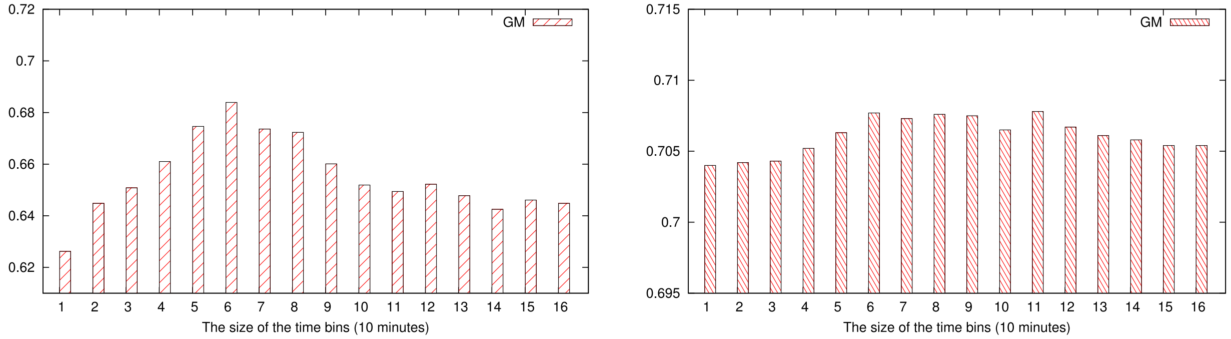
Here, we study the effect of considering different features: **Basic**: direct application of the basic LFM (Eq. 2); **BOW**: the LFM incorporating the bag-of-words (Eq. 4); **Syntactic**: the LFM incorporating the dependency-based syntactic units (Eq. 5); **Temporal**: the LFM incorporating the temporal information (Eq. 6); **Neighbor**: the LFM incorporating the neighborhood influence (Eq. 7); and **Full**: the model fully incorporating all of the features (Eq. 8).

The results of the comparison are presented in Table 3 with the best results highlighted in boldface. We make three observations from the results. First, *Full* performs the best on both data sets. Although the highest values of SEN for Repeated data and SPE for Random data do not appear in the results of *Full*, the overall metric of *Full*, GM, is always the best performing metric on both data sets. In particular, the GM of *Full* is approximately 9% and 13% greater than that of *Basic* for Repeated data and Random data, respectively. This result indicates that all of the factors need to be considered for the model, and our proposed features can effectively improve performance. Second, compared with *Basic, BOW* significantly improves the predictive performance. The GM of *BOW* outperforms *basic* by 7% and 9% on Repeated data and Random data, individually. Hence, *BOW* can obviously alleviate the data sparsity problem, whereas *Syntactic* can further enhance the predictive result relative to *BOW*, which suggests the effectiveness of identifying syntactic units. Last, both *Temporal* and *Neighbor* achieve better GM than *Basic*. In particular, for Random data, the GMs of *Temporal* and *Neighbor* are approximately 4% and 7% greater than that of *Basic*, respectively. These results show that both features can facilitate the prediction.

4.5. Impact of the time bin size

Before building the post-time matrix, we divide time into bins of equal size, such as 10 minutes, to discretize continuous time. In this set of experiments, we examine how the size of the time bins impacts the accuracy of the prediction. Our full model is selected for evaluation.

Figs. 2(a) and (b) present the GMs of the two data sets for various time bin sizes, where the x-axis corresponds to the size of the time bin (e.g., 1 for 10 minutes and 2 for 20 minutes). The two figures show that the GM initially increases with the size of the time bin to its peak at a size of 60 minutes ($x = 6$) and subsequently decreases. When the size is smaller



(a) Repeated Data

(b) Random Data

Fig. 2. Predictive performance for different time bin sizes.

than 60 minutes, the time bins with the smallest size obtain the lowest accuracy. For example, the results for the bin with 10 minutes ($x = 1$) are the lowest for the two data sets because the smaller size leads to a larger number of bins, which may make the matrix sparser and further result in poorer performance. After the peak at a size of 60 minutes, the performance begins to decline as the size increases because a larger size might weaken the ability to capture the time sensitivity.

Moreover, the fluctuation of the GM for Repeated data is larger than that for Random data. For example, the difference between the GMs for 60 minutes and 10 minutes exceeds 5% for Repeated data but is less than 1% for Random data. The reason is that the number of tweets on Repeated data is significantly smaller, and the size of the time bins has an important impact on the results. However, when the training data set is large, such as Random data, the size of the time bins has little effect on the predictive performance.

5. Conclusions and discussion

In this paper, we investigated the problem of selecting suitable publication times for online promotion and devised a novel time sensitivity-based predictive model for this task. This model can capture the sensitivity of the publication time and output multiple popularity values for an input post. In this model, we incorporated the syntactic units by decomposing the posts into noun and verb units to alleviate data sparsity and further exploited the temporal information (post density and audience activity level) and the neighborhood influence feature to improve predictive performance. To evaluate the proposed method, we applied it in two scenarios based on data from Twitter. The experimental results demonstrate that the proposed model outperforms state-of-the-art methods, and the proposed features significantly improve accuracy.

This study is beneficial for three main applications. First, for social media optimization platforms, based on the proposed model, a system for optimizing the delivery of posts can be developed to help users select suitable publication times for their posts. Our proposed model can predict when a post should be published to obtain higher popularity. Hence, based on this model, the system can provide users with the appropriate publication times. Second, for content providers and content delivery networks, knowing when posts will be popular can facilitate planning for sufficient storage and computational resources. Providers and networks set large-scale caching infrastructures to distribute copies of content across multiple locations. It is vital for cache replacement policies to learn which content will be popular and when. Third, for content producers, understanding the trend of content popularity over time can help to adjust promotion strategies and create new content. By learning the impact of publication time on popularity, producers can create more accurately targeted content to improve the effectiveness of information propagation. Moreover, learning that certain content will be popular can help advertisers develop revenue models and search engines generate better rankings for user queries.

Several simplifying assumptions and limitations exist for our proposed model, which can become fruitful directions for further investigation. First, our analyses and experiments only show that the model can be applied to Twitter but might not be appropriate for platforms such as Facebook because of the long lifespan of posts. We would like to investigate whether this model can be applied to other platforms, such as Reddit and Pinterest, and whether particular features of these platforms, such as image information, can be incorporated into the model. Second, the popularity of a post is influenced by a variety of internal and external factors in the open world, and this model only takes into account factors of Twitter. An example of an external factor is the popularity of topics on the Internet (which can be measured via Google Trends). If posts are published when their topics are more popular on the Internet, they may be frequently searched on search engines and gain further responses. Hence, we are interested in exploiting external factors for the model.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Chunjing Xiao: Writing - original draft, Data curation, Methodology, Conceptualization. **Chun Liu:** Conceptualization, Methodology, Resources. **Ying Ma:** Resources, Visualization, Writing - review & editing. **Zheng Li:** Software, Validation, Investigation. **Xucheng Luo:** Writing - review & editing, Resources, Formal analysis.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61402151 and 61806074), Science and Technology Foundation of Henan Province of China (No.162102410010).

References

[1] M. Tavana, E. Momeni, N. Rezaeiniya, S.M. Mirhedayatian, H. Rezaeiniya, A novel hybrid social media platform selection model using fuzzy ANP and COPRAS-G, Expert Syst. Appl. 40 (14) (2013) 5694–5702.

[2] R. Effing, T.A. Spil, The social strategy cone: towards a framework for evaluating social media strategies, Int. J. Inf. Manage. 36 (1) (2016) 1–8.

[3] Y. Hou, C. Lampe, Social media effectiveness for public engagement: Example of small nonprofits, in: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, 2015, pp. 3107–3116.

[4] K. Kuang, M. Jiang, P. Cui, S. Yang, Steering social media promotions with effective strategies, in: 2016 IEEE 16th International Conference on Data Mining, 2016, pp. 985–990.

[5] D. Bhattacharya, S. Ram, Rt @news: an analysis of news agency ego networks in a microblogging environment, ACM Trans Manag Inf Syst 6 (3) (2015) 1–25.

[6] S. Gao, J. Ma, Z. Chen, Modeling and predicting retweeting dynamics on microblogging platforms, in: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, 2015, pp. 107–116.

[7] A.F. Costa, Y. Yamaguchi, A.J.M. Traina, J. Caetano Traina, C. Faloutsos, Rsc: Mining and modeling temporal activity in social media, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 269–278.

[8] S. Apreleva, A. Cantarero, Predicting the location of users on twitter from low density graphs, in: IEEE International Conference on Big Data, 2015, pp. 976–983.

[9] J. Kulshrestha, F. Kooti, A. Nikravesh, K.P. Gummadi, Geographic dissection of the twitter network, in: Proceedings AAAI International Conference on Weblogs and Social Media, 2012, pp. 202–209.

[10] C. Tan, L. Lee, B. Pang, The effect of wording on message propagation: Topic- and author-controlled natural experiments on twitter, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014, pp. 175–185.

[11] A.B. Rios-Alvarado, I. Lopez-Arevalo, V.J. Sosa-Sosa, Learning concept hierarchies from textual resources for ontologies construction, Expert Syst. Appl. 40 (15) (2013) 5907–5915.

[12] C. Xiao, F. Zhou, Y. Wu, Predicting audience gender in online content-sharing social networks, Journal of the American Society for Information Science and Technology 64 (6) (2013) 1284–1297.

[13] P. Adamopoulos, V. Todri, The effectiveness of marketing strategies in social media: Evidence from promotional events, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 1641–1650.

[14] S.M. Kim, K. Kageura, J. McHugh, S. Nepal, C. Paris, B. Robinson, R. Sparks, S. Wan, Twitter content eliciting user engagement: A case study on australian organisations, in: Proceedings of the 26th International Conference on World Wide Web Companion, 2017, pp. 807–808.

[15] H.-J. Kim, J. Lee, D.-K. Chae, S.-W. Kim, Crowdsourced promotions in doubt: analyzing effective crowdsourced promotions, Inf. Sci. 432 (2018) 185–198.

[16] Y. Xue, C. Xiao, X. Luo, W. Yang, Predicting platform preference of online contents across social media networks, IEEE Access 7 (2019) 136428–136438.

[17] G. Szabo, B.A. Huberman, Predicting the popularity of online content, Commun. ACM 53 (2010) 80–88.

[18] S. Van Canneyt, P. Leroux, B. Dhoedt, T. Demeester, Modeling and predicting the popularity of online news based on temporal and content-related features, Multimed. Tools Appl. 77 (1) (2018) 1409–1436.

[19] F. Sabate, J. Berbegal-Mirabent, A. Canabate, P.R. Lebherz, Factors influencing popularity of branded content in facebook fan pages, European Management Journal 32 (6) (2014) 1001–1011.

[20] E. Bakshy, J.M. Hofman, W.A. Mason, D.J. Watts, Everyone's an influencer: quantifying influence on twitter, in: Proceedings of the fourth ACM international conference on Web Search and Data Mining, 2011, pp. 65–74.

[21] I. Arapakis, B.B. Cambazoglu, M. Lalmas, On the feasibility of predicting popular news at cold start, J. Assoc. Inf. Sci. Technol. 68 (5) (2017) 1149–1164.

[22] L. Hong, O. Dan, B.D. Davison, Predicting popular messages in twitter, in: Proceedings of the 20th international conference companion on World wide web, 2011, pp. 57–58.

[23] M. Jenders, G. Kasneci, F. Naumann, Analyzing and predicting viral tweets, in: Proceedings of the 22nd International Conference on World Wide Web, 2013, pp. 657–664.

[24] M. Vasconcelos, J.M. Almeida, M.A. Goncalves, Predicting the popularity of micro-reviews: afoursquare case study, Inf Sci 325 (2015) 355–374.

[25] J. Cheng, L. Adamic, P.A. Dow, J.M. Kleinberg, J. Leskovec, Can cascades be predicted? in: Proceedings of the 23rd international conference on World wide web, 2014, pp. 925–936.

[26] D. Vallet, S. Berkovsky, S. Ardon, A. Mahanti, M.A. Kafaar, Characterizing and predicting viral-and-popular video content, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, 2015, pp. 1591–1600.

[27] C.-T. Li, M.-K. Shan, S.-H. Jheng, K.-C. Chou, Exploiting concept drift to predict popularity of social multimedia in microblogs, Inf Sci 339 (4) (2016) 310–331.

[28] J. Han, D. Choi, J. Joo, C.-N. Chuah, Predicting popular and viral image cascades in pinterest, in: The Eleventh International AAAI Conference on Web and Social Media, 2017, pp. 81–91.

[29] Q. Zhao, M.A. Erdogdu, H.Y. He, A. Rajaraman, J. Leskovec, Seismic: A self-exciting point process model for predicting tweet popularity, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 1513–1522.

[30] S. Mishra, M.-A. Rizoiu, L. Xie, Feature driven and point process approaches for popularity prediction, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, 2016, pp. 1069–1078.

[31] M.-A. Rizoiu, L. Xie, S. Sanner, M. Cebrian, H. Yu, P. Van Hentenryck, Expecting to be hip: Hawkes intensity processes for social media popularity, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 735–744.

[32] Y. Koren, Collaborative filtering with temporal dynamics, Commun ACM 53 (4) (2010) 89–97.

[33] G. Comarela, M. Crovella, V. Almeida, F. Benevenuto, Understanding factors that affect response rates in twitter, in: Proceedings of the 23rd ACM Conference on Hypertext and Social Media, 2012, pp. 123–132.

[34] K. Shen, J. Wu, Y. Zhang, Y. Han, X. Yang, L. Song, X. Gu, Reorder user's tweets, ACM Trans. Intell. Syst. Technol. 4 (1) (2013) 1–17.

[35] F. Benevenuto, T. Rodrigues, M. Cha, V. Almeida, Characterizing user behavior in online social networks, in: Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement, 2009, pp. 49–62.

[36] Z. Qin, Y. Wang, H. Cheng, Y. Zhou, Z. Sheng, V.C.M. Leung, Demographic information prediction: a portrait of smartphone application users, IEEE Trans. Emerg. Top. Comput. 6 (3) (2018) 432–444.

[37] M. Kusner, Y. Sun, N. Kolkin, K. Weinberger, From word embeddings to document distances, in: Proceedings of The 32nd International Conference on Machine Learning, 2015, pp. 957–966.

[38] R.M. Bell, Y. Koren, Scalable collaborative filtering with jointly derived neighborhood interpolation weights, in: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining, 2007, pp. 43–52.

[39] L. Bottou, Stochastic learning, Advanced Lectures on Machine Learning 3176 (2004) 146–168.

[40] C. Xiao, Z. Qin, X. Luo, A. Kuzmanovic, Understanding factors that affect web traffic via twitter, in: Proceedings of the 17th Web Information Systems Engineering, 2016, pp. 170–185.

[41] B. Han, P. Cook, T. Baldwin, Lexical normalization for social media text, ACM Trans. Intell. Syst. Technol. 4 (1) (2013) 1–27.

[42] M. Kubat, S. Matwin, Addressing the curse of imbalanced training sets: One-sided selection, in: Proceedings of the Fourteenth International Conference on Machine Learning, 1997, pp. 179–186.

[43] C. Zhang, J. Bi, S. Xu, E. Ramentol, G. Fan, B. Qiao, H. Fujita, Multi-imbalance: an open-source software for multi-class imbalance learning, Knowl. Based Syst. 174 (2019) 137–143.

[44] C. Zhang, C. Liu, X. Zhang, G. Almpanidis, An up-to-date comparison of state-of-the-art classification algorithms, Expert Syst Appl 82 (2017) 128–150.

[45] C. Xiao, D. Han, Y. Ma, Z. Qin, Csigan: robust channel state information-based activity recognition with gans, IEEE Internet Things J. 6 (6) (2019) 10191–10204.

[46] Y. Kim, Convolutional neural networks for sentence classification, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1746–1751.