SentiDiff: Combining Textual Information and Sentiment Diffusion Patterns for Twitter Sentiment Analysis

Lei Wang, Jianwei Niu, *Senior Member, IEEE*, and Shui Yu, *Senior Member, IEEE*

Abstract—Twitter sentiment analysis has become a hot research topic in recent years. Most of existing solutions to Twitter sentiment analysis basically only consider textual information of Twitter messages, and struggle to perform well when facing short and ambiguous Twitter messages. Recent studies show that sentiment diffusion patterns on Twitter have close relationships with sentiment polarities of Twitter messages. Therefore, in this paper, we focus on how to fuse textual information of Twitter messages and sentiment diffusion patterns to obtain better performance of sentiment analysis on Twitter data. To this end, we first analyze sentiment diffusion by investigating a phenomenon called *sentiment reversal*, and find some interesting properties of sentiment reversals. Then, we consider the inter-relationships between textual information of Twitter messages and sentiment diffusion patterns, and propose an iterative algorithm called *SentiDiff* to predict sentiment polarities expressed in Twitter messages. To the best of our knowledge, this work is the first to utilize sentiment diffusion patterns to help improve Twitter sentiment analysis. Extensive experiments on real-world dataset demonstrate that compared with state-of-the-art textual information based sentiment analysis algorithms, our proposed algorithm yields PR-AUC improvements between 5.09 and 8.38 percent on Twitter sentiment classification tasks.

Index Terms—Sentiment analysis, sentiment diffusion, social networks, feature fusion, graph analysis

1 Introduction

Twitter, a popular micro-blogging service around the world, has been shaping and transforming the way people obtain information from people or organizations that they are interested in. On Twitter, users can publish status update messages, called *tweets*, to tell their followers what they are thinking, what they are doing, or what is happening around them. In addition, users can interact with another user by replying to or reposting his/her tweets. Since established in 2006, Twitter has become one of the largest online social networking platforms in the world [1]. Given the ever-growing amount of data available from Twitter, mining users' sentiment polarities expressed in Twitter messages has become a hot research topic due to its wide applications [2]. For example, by analyzing Twitter users' sentiment polarities on political parties and candidates, several tools have been developed to provide strategies for political elections [3], [4]. Business companies also use Twitter sentiment analysis as a fast and effective way to monitor people's feelings towards their products and brands [5].

The objective of sentiment analysis on Twitter data is to classify the sentiment polarity of a Twitter message as positive, neutral or negative. One way to perform Twitter sentiment analysis is to directly exploit traditional text sentiment analysis methods [6]. However, different from other text forms such as news reports and book articles, Twitter messages are often short and ambiguous. In addition, there are more slangs, acronyms, misspelled words and modal particles in Twitter messages due to their casual form [7], [8]. As a result, the performance of traditional text sentiment analysis algorithms drops drastically when applied to predict sentiment polarities of Twitter messages. To solve this problem, many novel sentiment analysis methods for Twitter messages have been developed. These methods can be roughly divided into two categories: fully supervised methods and distantly supervised methods [9].

The fully supervised methods aim to learn sentiment classifiers based on manually labeled data and sentiment lexicons [10], [11]. One major problem of fully supervised methods is that it is time-consuming and labor-intensive to manually build sentiment lexicons and label the data, and consequently the sentiment lexicons and labeled data used by most methods are often too small to guarantee good performance. In addition, fully supervised methods usually rely on hand-crafted features, and how to design effective features is still a challenging task. The distantly supervised methods learn sentiment classifiers from data with noisy labels such as emoticons and

- *L. Wang is with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, China. E-mail: lei@buaa.edu.cn.*
- *J. Niu is with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing 100191, China, and also with the Beijing Advanced Innovation Center for Big Data and Brain Computing (BDBC), and the Hangzhou Innovation Institute of Beihang University, Hangzhou, Zhejiang, China. E-mail: niujianwei@buaa.edu.cn.*
- *S. Yu is with the School of Software, University of Technology Sydney (UTS), Ultimo, NSW 2007, Australia. E-mail: shui.yu@uts.edu.au.*

Manuscript received 24 June 2018; revised 3 Jan. 2019; accepted 22 Apr. 2019. Date of publication 26 Apr. 2019; date of current version 10 Sept. 2020.
(Corresponding author: Jianwei Niu.)
Recommended for acceptance by T. Palpanas.
Digital Object Identifier no. 10.1109/TKDE.2019.2913641

hashtags. These methods use emoticons like ":)" and ":(" as noisy labels for sentiment analysis, and assume that a message containing ":(" is more likely to express a negative sentiment polarity and that containing ":)" is more likely to be a positive one [7], [12]. Although these distantly supervised methods can avoid labor-intensive manual annotation, their performance is not satisfactory because of the noise in the labels [9], [13]. The problem of noisy labels for sentiment analysis can be alleviated by applying pre-processing techniques [14]. However, recent work has verified that there are no effective pre-processing methods for all datasets and algorithms [15]. One pre-processing method effective for one specific algorithm and one specific dataset may even result in performance decrease of sentiment analysis when it is applied to another dataset or algorithm. In general, both fully supervised and distantly supervised solutions to Twitter sentiment analysis basically only focus on textual information of Twitter messages, and cannot achieve satisfactory performance due to unique characteristics of Twitter messages.

Sentiment diffusion, mainly about analyzing how information diffusion is affected by sentiments in social networks, has also attracted increasing attention from many research communities [16], [17], [18]. On Twitter, users can repost a tweet from another Twitter user and share it (i.e., *retweet*) with their own followers by clicking the retweet button within the tweet (or just typing "RT" or "via" at the beginning of a tweet to indicate that they are reposting someone else's content). When reposting a tweet, users can add a comment about the tweet and post it together with the original tweet (some tweets are reposted without any added comments, and these retweets are often ignored in sentiment diffusion studies as it is hard to know the sentiments expressed in these retweets). In this way, tweets and retweets can convey information about their authors' sentiment polarities on an issue. Therefore, we can investigate sentiment diffusion on Twitter by looking at how sentiment polarities differ from a tweet to its retweets [19].

Recently, fusing knowledge from multiple domains (but potentially connected) organically has offered new opportunities for doing research in many machine learning and data mining tasks [20], [21]. Recent studies on sentiment diffusion show that on Twitter, users' sentiment polarities are influenced by people they are following [22], as well as their positions within information propagation processes [23]. Although sentiment diffusion patterns have close relationships with sentiment polarities of Twitter messages, existing work on Twitter sentiment analysis basically only considers the textual information of Twitter messages, but ignores sentiment diffusion information.

Considering the shortcomings of existing solutions to Twitter sentiment analysis that only consider textual information and the close relationships between sentiment diffusion patterns and sentiment polarities of Twitter messages, we argue that the best strategy is to fuse textual information of Twitter messages and sentiment diffusion information in a supervised learning framework. However, how to organically integrate these two different kinds of information into the same learning framework is still a challenge. In this paper, we propose a novel algorithm called *SentiDiff* to handle this challenge. The main contributions of this paper are summarized as follows.

- We study sentiment diffusion on Twitter by investigating *sentiment reversal*, the phenomenon that a tweet and its retweet have different sentiment polarities. We analyze the properties of sentiment reversals, and propose a sentiment reversal prediction model.
- To predict the sentiment polarity of each Twitter message, we propose an iterative algorithm called *SentiDiff*, which takes the inter-relationships between textual information of Twitter messages and sentiment diffusion patterns into consideration. Given a tweet and its retweet, if their sentiment polarities predicted by textual information based sentiment classifier are consistent with the prediction result of sentiment reversal, the probability of messages to be classified correctly by textual information based sentiment classifier will increase. Otherwise, the probability will decrease. In this way, sentiment reversals can be combined with textual information of Twitter messages.
- We conduct a series of experiments to evaluate the performance of our proposed algorithm. The experimental results show that our proposed *SentiDiff* algorithm helps state-of-the-art textual information based sentiment analysis algorithms achieve PR-AUC improvements between 5.09 and 8.38 percent.

To the best of our knowledge, this work is the first to apply sentiment diffusion information to help improve Twitter sentiment analysis. Our *SentiDiff* algorithm is a general framework, and it can be easily extended to predict sentiment polarities of messages from other online social networks.

The rest of this paper is organized as follows. Section 2 describes the dataset used in this paper and some related definitions. We analyze the properties of sentiment reversals from two perspectives in Sections 3 and 4, and propose a sentiment reversal prediction model in Section 5. Then we introduce how to fuse textual and sentiment diffusion information in Section 6, and validate our proposed *SentiDiff* algorithm in Section 7. After introducing related work on sentiment analysis and sentiment diffusion in Section 8, we finally conclude this paper and point out future work in Section 9.

2 Data Description and Some Definitions

2.1 Dataset Description

We obtain Twitter tweet and retweet data through our collaborations on research with Beijing Intelligent Starshine Information Technology Corporation, a leading big data collection and mining service provider in China. In this paper, each tweet or retweet is assigned a sentiment label: $+1$ (positive), 0 (neutral) or -1 (negative). We employ 15 raters to manually assign a sentiment label for each tweet and retweet. Note that a lot of tweets are reposted without any added comments. Under such conditions, it is hard to know the sentiment polarities expressed in these retweets. Therefore, retweets without added comment information are ignored in sentiment labelling and sentiment analysis. We obtain a labeled dataset which contains over 100,000 tweets and retweets in total.

Before manual annotation, each rater is asked to assign sentiment labels for a dataset containing 500 Twitter messages with ground truth sentiment labels. Then 3 raters with the top 3 best labelling performance are selected as *senior raters*, and the rest of them are regarded as *normal raters*. For 30,000 tweets and retweets, each of them is assigned a sentiment label by one senior rater. For the remaining portion of

TABLE 1
Some Statistics for Twitter Dataset

# positive messages	# neutral messages	# negative messages
28,323	52,714	19,712
# total messages	# users	Cohen's Kappa value
100,749	8,516	0.91

tweets and retweets, the sentiment label of each one is evaluated by two normal raters. If two normal raters give conflicting sentiment labels for a message, the final sentiment label will be determined by a senior rater. For Twitter messages rated by two normal raters, we calculate the inter-rater agreement based on Cohen's Kappa measurement, and obtain a high agreement of 0.91, indicating that the sentiment labels of Twitter messages given by raters are quite reliable. On average, each Twitter message in our dataset is evaluated by 1.73 senior or normal raters. Each senior rater assigns sentiment labels for 11,179 messages on average, and each normal rater labels 11,790 messages on average. We display some statistics for Twitter dataset used in this paper in Table 1.

2.2 Some Definitions

Definition 1 (Repost Cascade Tree). *Repost cascade tree is a directed, acyclic labeled graph, which is used to capture the relationships between a tweet and its retweets. Formally, given a repost cascade tree $T(V, E, l)$ which contains a set of nodes V, a set of edges E and a function l, each node represents a tweet or retweet, and a directed edge from i to j is created if retweet j is a reposting of someone's tweet i. Let $\sum_V$ be the set of possible sentiment labels of Twitter messages (i.e., positive, neutral and negative). The function l attaches each tweet or retweet to its sentiment label, i.e., $l : V \rightarrow \sum_V$. The root node of a repost cascade tree is the node without parent node, and locates at the original tweet which is the earliest posted. In a repost cascade tree, every retweet has a unique parent, and it is impossible to find cycles in a repost cascade tree because a retweet is always posted later than its parent tweet. In Fig. 1a, message Ma is posted by user A at first. Then, messages Mb and Mc, two repostings of Ma, are posted by users B and C, respectively. At last, user D posts messages Md and Me, which are repostings of Mb and Mc respectively. According to the definition of repost cascade tree, a repost cascade tree is constructed and displayed in Fig. 1b.*

Definition 2 (Repost Diffusion Network). *Following [24], we utilize repost diffusion networks to describe how users interact with each other on Twitter. Repost diffusion network $N(V, E)$ is a directed graph with self loops and parallel edges, and contains a set of nodes V and a set of edges E. In a repost diffusion network, each node $v \in V$ represents a user, and a directed edge $e \in E$ from A to B is built if user B reposts a tweet posted by user A. If user B reposts user A's tweets many times, multiple directed edges from A to B will be created. In Fig. 1c, a repost diffusion network is generated, in corresponding to the example interactions among Twitter users in Fig. 1a. In this paper, we define a repost diffusion network as a weakly connected component, and a set of diffusion networks is constructed according to the interactions among Twitter users.*

Definition 3 (Sentiment Reversal). *Sentiment reversal is defined as the phenomenon that a tweet (parent tweet) and its retweet (child tweet) have different sentiment polarities. Formally, given a repost cascade tree $T(V, E, l)$, if a node i (parent tweet) and its child node j (child tweet) are attached different sentiment labels by function l (i.e., $l(i) \neq l(j)$), a sentiment reversal occurs between i and j. In Fig. 1b, sentiment reversals occur between Ma and Mb, and between Mc and Me. In our dataset, sentiment reversals occur in less than 20 percent of the total interactions between users.*

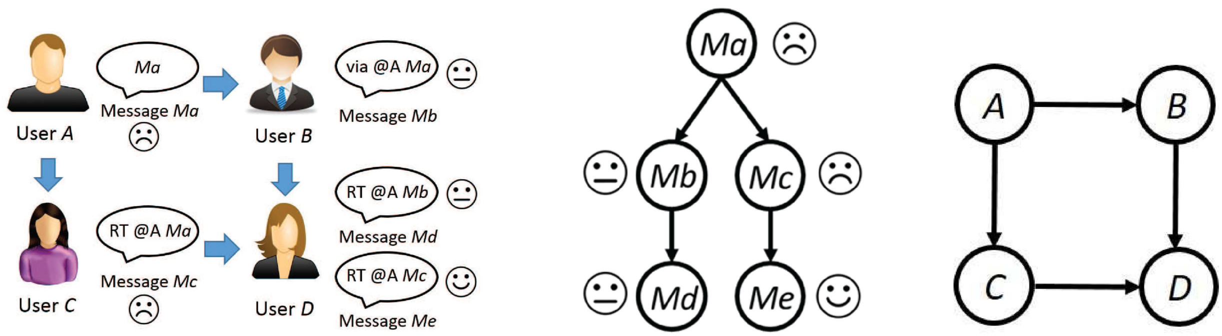
Note that to analyze sentiment diffusion on Twitter, we have to guarantee that the constructed repost cascade trees and repost diffusion networks are complete. Most of existing sentiment data on Twitter (e.g., SemEval [25]) is a small sample and most of the real cascades will be split in many short retweet chains due to some missing tweets. Therefore, we obtain Twitter tweet and retweet data through our collaborations on research with a commercial company to guarantee that the constructed repost cascade trees and repost diffusion networks are complete.

3 Properties of Sentiment Reversals from Repost Cascade Tree Perspective

In this section, we probe into the characteristics of sentiment reversals, and investigate how sentiment reversals are influenced by the properties of repost cascade trees.

3.1 Sentiment Reversal versus Cascade Tree Depth

It is a natural way to measure the properties of sentiment reversals by examining the distribution over cascade tree depths at which sentiment reversals happen. We measure the cascade tree depth for each sentiment reversal occurring



(a) *Example interactions among Twitter users* (b) *Repost cascade tree* (c) *Repost diffusion network*

Fig. 1. Example of constructing repost cascade tree and repost diffusion network.

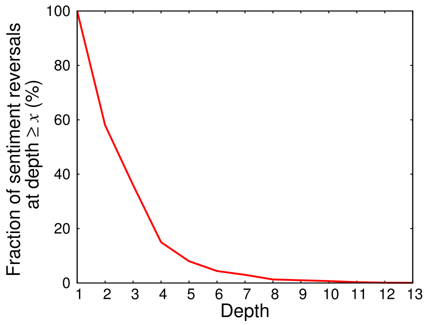


Fig. 2. Distribution over sentiment reversal depth.

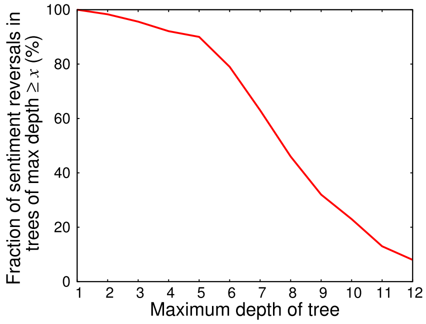


Fig. 3. Fraction of sentiment reversals in trees of specific maximum depth.

during our observation period, which is defined as the number of steps from the root node in the repost cascade tree. The distribution of sentiment reversals over cascade tree depth is displayed in Fig. 2, where we can observe that a large proportion of sentiment reversals occur near the root nodes of cascade trees. For example, 85 percent of sentiment reversals happen at depth 3 or lower (i.e., 15 percent of sentiment reversals occur at depth 4 or greater in Fig. 2).

3.2 Sentiment Reversal versus Cascade Tree Maximum Depth

Next, we investigate the relationship between sentiment reversals and cascade tree maximum depth. Here we divide all repost cascade trees into two categories by their maximum depths: deep cascade trees which have maximum depth 6 or greater, and shallow cascade trees which have maximum depth 5 or lower. From Fig. 3, we can see that around 80 percent of sentiment reversals are part of deep cascade trees, whereas only about 20 percent of sentiment reversals reside in shallow cascade trees. Therefore, we can draw the conclusion that sentiment reversals are more likely to happen in deep cascade trees than shallow cascade trees.

In repost cascade trees, how does sentiment flow from a parent tweet to its child tweets? The result is displayed in Fig. 4. In Fig. 4, the sentiment change between parent and child tweets at cascade depth L is calculated by computing the average absolute difference of sentiment labels between parent and child tweets with nodes of a specific cascade tree at a given cascade depth L, and then averaging overall cascade trees. As shown in Fig. 4, sentiment change between parent and child tweets for all the cascade trees exhibits three phases: (1) a slight increase at near of cascade tree root, but then (2) a precipitous drop up to $x = 6$, and finally (3) return slowly to a relatively stable state from $x > 6$.

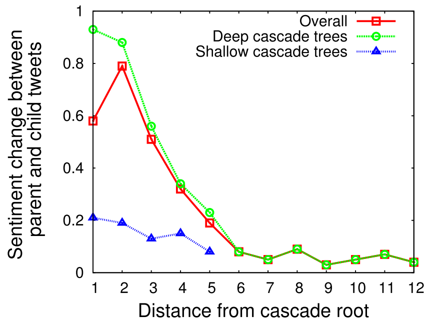
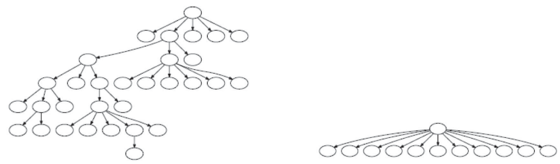


Fig. 4. Sentiment change between parent and child tweets as a function of distance from the cascade tree root.



(a) *Cascade tree for narrowly deep branching diffusion* (b) *Cascade tree for shallow, broadcast-like diffusion*

Fig. 5. Example of cascade trees for narrowly deep branching diffusion and shallow, broadcast-like diffusion.

Do all the cascade trees have such a 3−phase sentiment change process? To answer this question, we study sentiment change between parent and child tweets for deep and shallow cascade trees separately, and show the result in Fig. 4. As illustrated in Fig. 4, for deep cascade trees, the sentiment change between parent and child tweets decreases gradually with the increasing distance from cascade tree root at first, and then it enters a stable state with a relatively small value. However, for shallow cascade trees, the sentiment change between parent and child tweets remains a relatively stable state with some mild oscillations all the time. Note that when analyzing overall cascade trees, there is a slight increase in sentiment change near the cascade tree root. However, we cannot observe similar phenomena when analyzing deep or shallow cascade trees separately. This is caused by the overwhelmingly high count of shallow cascade trees with relatively small sentiment changes.

3.3 Sentiment Reversal versus Structural Virality

In this section, we explore the relationship between sentiment reversals and structural virality of cascade trees. Structural virality is used to distinguish the narrowly deep branching diffusion structures (see Fig. 5a) from the shallow, broadcast-like diffusion structures (i.e., star-like cascade trees, see Fig. 5b).

The Wiener index is the most commonly used evaluation metric for structural virality [26], and it is computed by averaging path distance between any two nodes in the cascade tree. A narrowly deep branching diffusion results in a relatively high score since the path distance between two nodes is comparatively long, while a shallow, broadcast-like diffusion has a quite low score on this metric. To ensure the distribution of sentiment reversals over structural virality is not skewed by small-sample effects, we restrict our focus to cascade trees comprising more than 100 nodes in this section, as was done in [27].

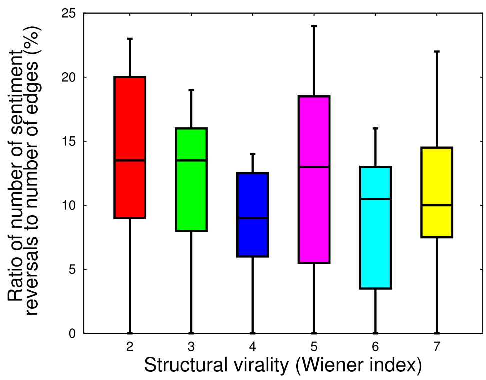


Fig. 6. Impact of structural virality of cascade trees on sentiment reversals.

Are sentiment reversals more likely to reside in cascade trees with lower Wiener index? Or conversely, is it easier for us to find sentiment reversals in cascade trees with higher Wiener index? The answers to these questions are shown in Fig. 6. In Fig. 6, for a given Wiener index W, we first obtain all the cascade trees with Wiener index W. Then for each cascade tree, we calculate the probability of finding sentiment reversals in it. From Fig. 6, we can observe that for all the cascade trees, the correlation between structural virality of cascade trees and the probability of finding sentiment reversals in them is surprisingly low, which implies that structural virality of cascade trees does not reveal much about the mechanisms of sentiment reversals.

Next, we investigate whether the low correlation between structural virality of cascade trees and the probability of finding sentiment reversals in cascade trees appears in all the cascade trees. We divide all the cascade trees into three types based on the sentiment labels of their roots $SentiLabel(root)$: negative cascade trees ($SentiLabel(root) = -1$), neutral cascade trees ($SentiLabel(root) = 0$) and positive cascade trees ($SentiLabel(root) = 1$). Then for each type of cascade trees, we compute the correlation between structural virality of cascade trees and the probability of finding sentiment reversals in them. The results are displayed in Fig. 7.

From Fig. 7, we can observe an interesting phenomenon: for neutral cascade trees, the probability that sentiment reversals occur in them decreases gradually with the increase of Wiener index. However, we cannot see similar patterns for negative or positive cascade trees. We can explain this phenomenon as follows. Existing studies on sentiment diffusion have found that compared with neutral tweets, positive and negative tweets are more likely to result in large-scale diffusions [28], [29]. For a tweet with neutral sentiment, it tends to use relatively tame words, and thus does not attract a lot of reposts [23]. A neutral cascade tree with more than 100 nodes and low Wiener index indicates that a tweet with neutral sentiment leads to a large-scale broadcast-like diffusion. The authors of [30] have verified that messages about controversial topics usually result in large-scale broadcast-like diffusions. Therefore, we guess that the reason of a large-scale broadcast-like diffusion caused by a neutral tweet is that this neutral tweet involves a controversial topic, which brings about a hot discussion on Twitter. Users give their own comments on this controversial topic, and thus a large number of sentiment reversals occur during the discussion on this controversial topic. To verify our conjecture, we randomly choose 60 neutral tweets which cause large-scale broadcast-like diffusions, and find that over 80 percent of them are about controversial political issues.

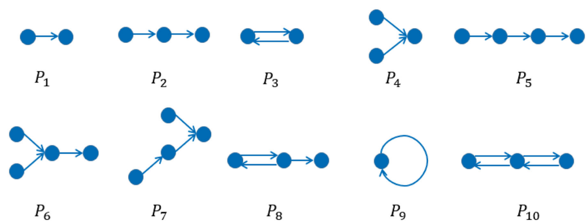


Fig. 8. Top-10 most frequent diffusion patterns in repost diffusion networks.

4 Properties of Sentiment Reversals from Repost Diffusion Network Perspective

In this section, we shift our focus to study the effect of repost diffusion networks on sentiment reversals.

4.1 Sentiment Reversal versus Diffusion Patterns

Are sentiment reversals more likely to occur in some specific diffusion patterns? Furthermore, does the relationship between sentiment reversals and specific diffusion patterns appear in all the repost diffusion networks? In this section, we will explore these questions. We restrict our analysis to diffusion networks with at least 20 edges in this section in order to avoid the small-scale sample effects.

To answer these questions, we need to find the most frequent diffusion patterns in repost diffusion networks at first [31]. We list all types of diffusion patterns appearing in all the repost diffusion networks, and rank them according to the frequencies they occur in diffusion networks. The top-10 most frequent diffusion patterns are displayed in Fig. 8.

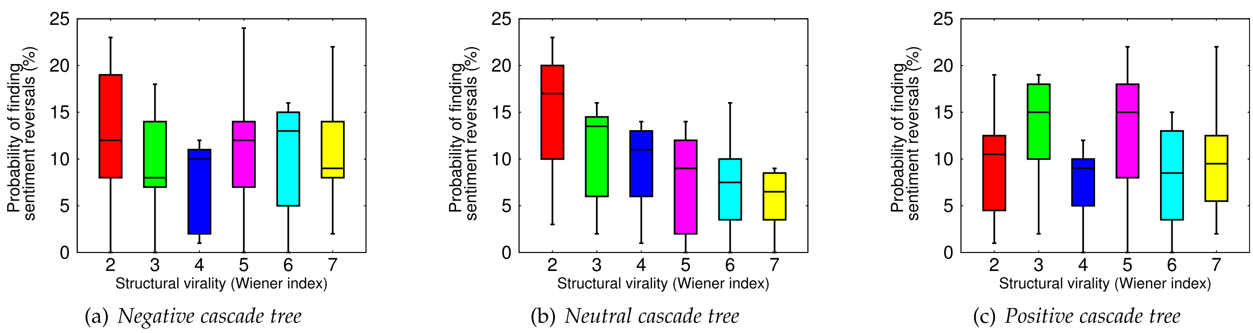


Fig. 7. Impact of structural virality of cascade trees on sentiment reversals for different cascade trees.

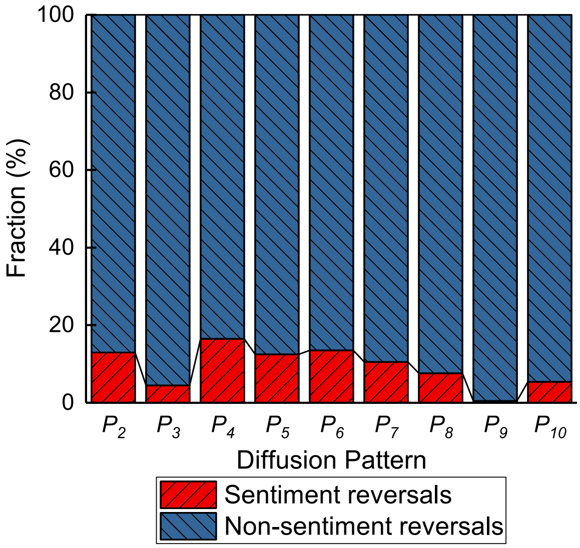


Fig. 9. Distribution of sentiment reversals over diffusion patterns.

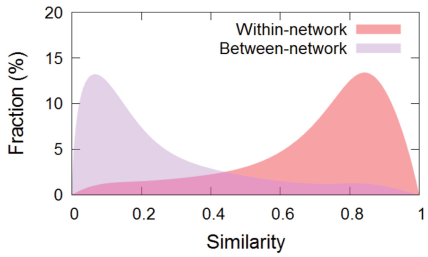


Fig. 10. Within-network and between-network similarity distributions over diffusion patterns.

Next, we investigate whether sentiment reversals are more likely to occur in some specific diffusion patterns. Note that diffusion pattern P_1 can represent all reposts between different users. Therefore, we ignore diffusion pattern P_1 when analyzing this issue. For each edge in diffusion networks, we first judge the diffusion patterns it belongs to. Then for each diffusion pattern, we calculate the number of sentiment reversals occurring in it. The distribution of sentiment reversals over diffusion patterns is shown in Fig. 9.

As illustrated in Fig. 9, the probability of finding sentiment reversals in diffusion patterns P_3, P_9 and P_{10} is much lower compared with other diffusion patterns. Diffusion pattern P_9 represents that a user reposts his/her own tweet, and the proportion of tweets in which sentiment reversals occur with diffusion pattern P_9 to the total tweets with diffusion pattern P_9 is surprisingly low (0.5 percent), implying that users tend to persist in their opinions on specific topics. Note that diffusion patterns P_3 and P_{10} are strongly connected graphs. Considering the reposting mechanism on Twitter, two users with diffusion pattern P_3 are likely to be friends. This result indicates that sentiment reversals are more likely to occur between users without friend relationship.

For the relationship between sentiment reversals and diffusion patterns, we continue to investigate whether there is *network homophily* present for these diffusion networks. Network homophily is defined as the phenomenon that for each diffusion network, the sentiment reversals in it tend to follow the same diffusion pattern (we still ignore diffusion pattern P_1 here).

To answer this question, given a distribution of sentiment reversals over diffusion patterns for a diffusion network, we wish to find a metric to measure how similar or diverse it is. Moreover, given two distributions of sentiment reversals over diffusion patterns for two different diffusion networks, we need to find a metric to quantify how similar they are. We adopt the *cascade homophily* metric used in [27] to fulfill this purpose. We define:

- The *within-network similarity* $W_P(N)$ of a diffusion network N on diffusion pattern P is the probability that two randomly selected sentiment reversals in network N follow the same diffusion pattern.
- The *between-network similarity* $B_P(N_1, N_2)$ of two diffusion networks N_1 and N_2 on diffusion pattern P is the probability that a randomly selected sentiment reversal from network N_1 and a randomly selected sentiment reversal from network N_2 follow the same diffusion pattern.

Compared with other metrics, these two metrics are easily interpretable, and not affected by the size of the diffusion networks being considered.

For every "large" diffusion network N (with more than 20 edges), we calculate the within-network similarity $W_P(N)$ of the sentiment reversals in the network. However, a large amount of network *similarity* is not sufficient to draw the conclusion that network *homophily* is present because of the lack of the baseline to compare against. Therefore, we also take a random sample of pairs of diffusion networks, and compute the between-network similarity $B_P(N_1, N_2)$. The distribution over these between-network similarities can be regarded as the baseline. If there is no network homophily on diffusion patterns at all, the within-network and between-network similarity distributions should be exactly the same. Then the extent to which they differ is an effective metric of network homophily.

In Fig. 10, we display the distributions of W_P and B_P over all the large diffusion networks for diffusion patterns. The result in Fig. 10 demonstrates that most of diffusion networks have within-network similar values higher than 0.7, whereas the between-network similarity values for most of diffusion networks are below 0.2. This result strongly suggests that there indeed exists the network homophily phenomenon on diffusion patterns in diffusion networks, i.e., for each diffusion network, the sentiment reversals in it tend to follow the same diffusion pattern.

4.2 Sentiment Reversal versus User Status

As we all know, people have a tendency to obey instructions from higher-status members (such as job seniorities, leaders) in daily life. As in offline realms of daily life, user *status* is a very important part of one's identity on Twitter. On Twitter, the status of a user can be evaluated by his/her out-degree in the diffusion network. According to the definition of diffusion network, the value of each user's out-degree is equal to the total reposted times of his/her tweets. A user with higher out-degree indicates that he/she has the capacity of attracting more attention and reposts from other users. Thus, we think that a user with higher out-degree has higher status. In this paper, each user is assigned a user status ranging from 0

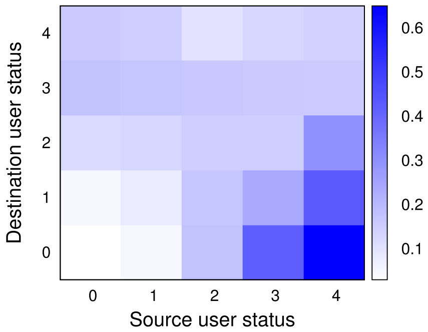


Fig. 11. Relationship between sentiment reversals and user status.

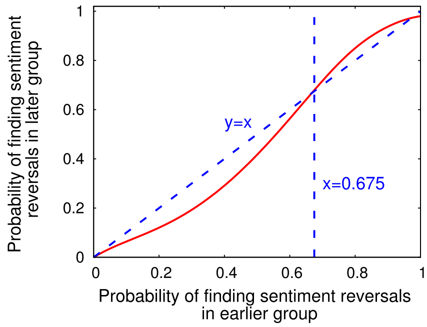


Fig. 12. Evolution of sentiment reversals.

(very low user status) to 4 (very high user status) according to his/her out-degree in the diffusion network. Then we analyze whether sentiment reversals follow a *status gradient*, meaning users tend to have the same or similar sentiment polarities with higher-status users on Twitter.

The answer to this question is shown in Fig. 11. In Fig. 11, the color of the cell (x, y) represents the likelihood that a sentiment reversal occurs if a user with status x reposts a tweet from a user with status y. In Fig. 11, there are several important points to observe. First, when a user reposts tweets from higher-status users, the probability of sentiment reversals occurring is not very low, indicating that users do not always have the same or similar sentiment polarities with higher-status users on Twitter. The underlying reason may be that sometimes some higher-status users post tweets about controversial topics and attract a lot of attention from lower-status users. Then, these lower-status users post a large number of retweets to express different sentiment polarities, leading to a mass of sentiment reversals. Second, if a user with very high status reposts a tweet from a user with very low status, there is a strikingly high probability that a sentiment reversal occurs between them. This phenomenon can be explained as follows. For users with very high status, it is unusual for them to be a follower of a user with very low status. Considering the mechanism of timeline on Twitter, they cannot see the tweets posted by these low-status users in their home timelines if they do not follow these low-status users. However, they will see how these low-status users are interacting with them from the notification timelines. From the notification timelines, high-status users can see which of their tweets have been reposted, plus the tweets directed to them (replies and mentions). If they find that some low-status users have posted some tweets opposite to their opinions, they may repost related tweets and add some comments to argue with these low-status users, consequently resulting in sentiment reversals. Third, the probability of observing sentiment reversals between two low-status users is surprisingly low, implying that low-status users are more likely to have the same or similar opinions with users having the same status.

4.3 Evolution of Sentiment Reversals

According to the definition of repost diffusion network, in a repost diffusion network, there are multiple edges between a specific pair of users A and B if one user interacts with another user by reposting another user's tweets (i.e., A reposts B's tweets, or B reposts A's tweets) more than one time. For each pair of users with multiple edges, how does the probability that sentiment reversals occur between them evolve over time? In this section, we move on to investigate this question. Our main object of analysis in this section is the probability of finding sentiment reversals within the same pair of users. To avoid the probability being affected by small-sample effects, we only focus on the pairs of users with at least 10 edges between them.

For each pair of users, we rank the edges between them according to the created time of these edges, and divide these edges into two groups. 50 percent of the total edges created earlier are contained in the *earlier* group, and the *later* group contains the rest of edges. Then for each pair of nodes, we can compute the probabilities of finding sentiment reversals between them in the earlier and later groups, respectively. To explore how the probability that sentiment reversals occur between a specific pair of users evolves over time, we calculate $P(x)$, the probability of finding sentiment reversals in the later group as a function of the probability x of finding sentiment reversals in the earlier group. The relationship between the probability of finding sentiment reversals in the earlier group and that in the later group is shown in Fig. 12.

As displayed in Fig. 12, $P(x) < x$ when $x < 0.675$, whereas $P(x) > x$ if $x > 0.675$. This result exhibits an interesting phenomenon. If the *affinity* between two users can be evaluated by the probability that sentiment reversals occur between them (lower probability means closer affinity), for two users who have already had close affinity, the affinity between them tends to be closer over time. However, for a pair of users who have had distant affinity, they are more likely to obtain more distant affinity over time.

5 Sentiment Reversal Prediction

In the previous sections, we have analyzed the properties of sentiment reversals from the perspective of repost cascade tree and repost diffusion network. In this section, we propose a sentiment reversal prediction model by integrating a comprehensive set of features used in the previous sections. More specifically, given the structural features of repost cascade trees and diffusion networks as well as the historical behaviors of users, can we predict whether a sentiment reversal occurs in a specific pair of parent and child tweets?

To build a sentiment reversal prediction model, we need to determine the features used in the prediction model first. Given a parent tweet $t_p(a)$ posted by user a and a child tweet $t_c(b)$ posted by user b ($t_p(a)$ and $t_c(b)$ reside in cascade tree T, and users a and b are in diffusion network N), we can extract three sets of features (cascade tree features, diffusion network

TABLE 2
List of Features Used in the Sentiment Reversal Prediction Model

Cascade Tree Features	Depth of $t_p(a)$ Maximum depth of T Wiener index of T Number of nodes in T Sentiment polarity of the root node of T based on textual information Out-degree of $t_p(a)$ in T
Diffusion Network Features	Diffusion pattern users a and b belong to User status of user a User status of user b Number of nodes in N Edge density of N
Historical Behavior Features	Number of sentiment reversals between users a and b Number of total interactions between users a and b Number of tweets which are reposted by user a from user b Number of tweets which are reposted by user b from user a

features and users' historical behavior features). The full list of features can be found in Table 2.

First, for each set of features, we train a non-liner SVM model with RBF kernel. Note that sentiment reversals occur in less than 20 percent of the total interactions between users, causing the number of positive examples and negative examples quite imbalanced. SVM classifiers will decline in classification performance if we use this imbalanced dataset to train models directly [32]. Recent studies [33] show that in an imbalanced situation, the majority class (the class with more samples) pushes the ideal decision boundary towards the minority class (non-majority class). SVM assumes that only Support Vectors (SVs) are informative for classification in maximum margin hyperplanes problems, and removing other samples does not substantially affect the classification performance. Therefore, we adopt the method similar to [34] to solve the imbalanced data classification problem. First, we assume that all minority class samples are informative due to their rarity. For majority class samples, only SVs are considered as informative. Due to highly skewed data distribution, it is hard to identify and extract all informative samples by using a single SVM. Despite this, a single SVM can identify a fraction of, although not all, informative samples. Then we remove these extracted informative samples from the original training dataset and form a smaller dataset, on which a new SVM is trained to identify another part of informative samples. This process is repeated several times. Finally, the majority class samples still remaining in the training dataset are discarded, and only these extracted informative samples are aggregated together with the minority class samples to generate the training set, on which the final SVM model is trained. In this way, we can obtain three sentiment reversal prediction models based on three sets of features.

Next, we combine various features from three different feature sets in a supervised learning framework. For classifier (e.g., feature set) j and class label k, we can obtain learned weight, w_{jk}, and bias term, b_{jk}, of the jth classifier for the kth class for score calibration. To learn the weight and bias terms of the jth classifier for the kth class, we train a linear SVM model, where samples of kth class are positive samples, and samples of other classes are negative samples. Details about how to learn the weight and bias terms can refer to [35]. In this way, we can get the weight and bias terms of each classifier for each class. Next, for a new sample, we apply the method similar to [36] to determine its class label.

For a new sample i, its score vector for classifier j is represented as

$$S_{ij} = (s_{i1j}, \ldots, s_{ikj}, \ldots, s_{iCj}), \tag{1}$$

where C denotes the number of classes of an object (C equals to 2 in this paper, i.e., whether sentiment reversal occurs between a pair of parent and child tweets or not). s_{ikj} represents the score of sample i for the kth class obtained by the jth classifier. Here s_{ikj} equals to the accurate probability of sample i to be classified as the kth class correctly by the jth classifier, and can be computed by using LibSVM [37]. Then for sample i, its calibrated score vector of jth classifier for each class is calculated as

$$\begin{aligned}\lambda_{ij} &= w_j S_{ij} + b_j \\ &= (w_{j1} s_{i1j} + b_{j1}, \ldots, w_{jk} s_{ikj} + b_{jk}, \\ &\quad \ldots, w_{jC} s_{iCj} + b_{jC}).\end{aligned} \tag{2}$$

The final score vector for sample i is calculated by summing the calibrated score vectors of each classifier

$$\begin{aligned}\Lambda_i &= \sum_{j=1}^{F} \lambda_{ij} = \left(\sum_{j=1}^{F} (w_{j1} s_{i1j} + b_{j1}), \ldots, \right. \\ &\quad \left. \sum_{j=1}^{F} (w_{jk} s_{ikj} + b_{jk}), \ldots, \sum_{j=1}^{F} (w_{jC} s_{iCj} + b_{jC}) \right) \\ &= (\Lambda_{i1}, \ldots, \Lambda_{ik}, \ldots, \Lambda_{iC}),\end{aligned} \tag{3}$$

where F is the total number of classifiers (feature sets). Λ_i is the final score vector of sample i after weighted feature fusion, and Λ_{ik} is the score for the kth class.

Finally, for sample i, its decision function of the classification problem is

$$l_i = \arg \max_{k=1,2,\ldots,C} \Lambda_{ik}, \tag{4}$$

where l_i is the final classification result (class label) of sample i.

6 Combine Textual Information and Sentiment Diffusion Patterns

In this section, we propose an iterative algorithm, called *SentiDiff*, to combine textual and sentiment diffusion information in a supervised learning algorithm. Before going to the details, we first list the notations used in this section in Table 3.

First, we train textual information based sentiment classifier and sentiment reversal prediction model based on the labeled dataset. Then, given a new set of Twitter messages which reside in the same cascade tree, the sentiment polarity of each Twitter message is predicted as Algorithm 1. The basic idea of fusing textual and sentiment diffusion information in *SentiDiff* algorithm is that if the prediction results between textual information based sentiment classifier and sentiment reversal prediction model are conflicting, the probability of Twitter messages to be classified correctly by textual information based sentiment classifier will decrease. Otherwise, the probability will increase.

Algorithm 1. *SentiDiff* Algorithm

Input: Textual information based sentiment classifier, sentiment reversal prediction model, N Twitter messages which reside in the same cascade tree.
Output: Sentiment labels of N Twitter messages.

1: Initialize $t \leftarrow 1$.
2: **for all** $sl \in \{-1, 0, 1\}$ **do**
3:   **for all** $m_i \in \{m_1, \dots, m_N\}$ **do**
4:     $TSP(m_i, sl) \leftarrow TP(m_i, sl)$
5:   **end for**
6: **end for**
7: **repeat**
8:   $V_{TS}^{(t)} \leftarrow (TSP(m_1, TL(m_1)), \dots, TSP(m_N, TL(m_N)))$
9:   **for all** $m_i \in \{m_1, \dots, m_N\}$ **do**
10:     **for all** $m_j \in parent(m_i) \bigcup child(m_i)$ **do**
11:       **if** $(TL(m_i) \neq TL(m_j) \bigwedge SP(m_i, m_j) < 0.5) \bigvee (TL(m_i) = TL(m_j) \bigwedge SP(m_i, m_j) \geq 0.5)$ **then**
12:         $TSP(m_i, TL(m_i)) \leftarrow TSP(m_i, TL(m_i)) - TP(m_i, TL(m_i)) \cdot TP(m_j, TL(m_j)) \cdot SP(m_i, m_j)$
13:         **for all** $sl \in \{\{-1, 0, 1\} - TL(m_i)\}$ **do**
14:           $TSP(m_i, sl) \leftarrow TSP(m_i, sl) + 0.5 \cdot TP(m_i, TL(m_i)) \cdot TP(m_j, TL(m_j)) \cdot SP(m_i, m_j)$
15:         **end for**
16:       **else**
17:         $TSP(m_i, TL(m_i)) \leftarrow TSP(m_i, TL(m_i)) + TP(m_i, TL(m_i)) \cdot TP(m_j, TL(m_j)) \cdot SP(m_i, m_j)$
18:         **for all** $sl \in \{\{-1, 0, 1\} - TL(m_i)\}$ **do**
19:           $TSP(m_i, sl) \leftarrow TSP(m_i, sl) - 0.5 \cdot TP(m_i, TL(m_i)) \cdot TP(m_j, TL(m_j)) \cdot SP(m_i, m_j)$
20:         **end for**
21:       **end if**
22:     **end for**
23:   **end for**
24:   $V_{TS}^{(t+1)} \leftarrow (TSP(m_1, TL(m_1)), \dots, TSP(m_N, TL(m_N)))$
25:   $V_{TS}^{(t+1)} \leftarrow V_{TS}^{(t+1)} / \|V_{TS}^{(t+1)}\|_1$
26:   $t \leftarrow t + 1$
27:   **until** $\|V_{TS}^{(t)} - V_{TS}^{(t-1)}\|_\infty < \beta$
28:   **for all** $m_i \in \{m_1, \dots, m_N\}$ **do**
29:   $FL(m_i) \leftarrow \arg\ \max_{sl \in \{-1,0,1\}} TSP(m_i, sl)$
30:   **end for**

In our *SentiDiff* algorithm, for Twitter message m_i, first we initialize $TSP(m_i, sl)$, the probability that m_i is classified with sentiment label sl, as $TP(m_i, sl)$, the probability of m_i to be classified with sentiment label sl correctly by textual information based sentiment classifier (line 4). Next, we consider the textual information of m_i's parent and child tweets, as well as the probabilities that sentiment reversals occur among them, and fuse textual information and sentiment diffusion information through an iteration process (lines $8-26$). To be more specific, for Twitter message m_i and its parent or child tweet m_j, if textual information based sentiment classifier

TABLE 3
Notations Used in this Paper

Notation	Meaning
$child(m_i)$	Child tweets of Twitter message m_i
$parent(m_i)$	Parent tweet of Twitter message m_i
$TL(m_i)$	Sentiment label of Twitter message m_i predicted by textual information based sentiment classifier
$TP(m_i, sl)$	Probability of Twitter message m_i to be classified with sentiment label sl correctly by textual information based sentiment classifier
$SP(m_i, m_j)$	Probability that sentiment reversal occurs between Twitter messages m_i and m_j predicted by sentiment reversal prediction model
$TSP(m_i, sl)$	Probability of Twitter message m_i to be classified with sentiment label sl after combining textual and sentiment diffusion information
$FL(m_i)$	Sentiment label of Twitter message m_i after combining textual and sentiment diffusion information

predicts that m_i and m_j express the same sentiment polarity (i.e., $TL(m_i) = TL(m_j)$) while sentiment reversal prediction model predicts that sentiment reversal occurs between m_i and m_j (i.e., $SP(m_i, m_j) \geq 0.5$), we think the results of textual information based sentiment classifier are conflicting with the result of sentiment reversal prediction model. Likewise, we also think the results of textual information based sentiment classifier and sentiment reversal prediction model are conflicting when m_i and m_j are predicted to express different sentiment polarities (i.e., $TL(m_i) \neq TL(m_j)$) while sentiment reversal prediction model predicts that sentiment reversal does not occur between m_i and m_j (i.e., $SP(m_i, m_j) < 0.5$). Under such conditions, the probability of m_i to be classified with sentiment label $TL(m_i)$ correctly, $TSP(m_i, TL(m_i))$, will decrease as (lines $10-12$):

$$TSP(m_i, TL(m_i)) \leftarrow TSP(m_i, TL(m_i)) - TP(m_i, TL(m_i)) \cdot TP(m_j, TL(m_j)) \cdot SP(m_i, m_j), \tag{5}$$

where $TL(m_i)$ denotes m_i's sentiment label predicted by textual information based sentiment classifier, and $SP(m_i, m_j)$ is the probability that sentiment reversal occurs between m_i and m_j.

Meanwhile, the probability of m_i to be classified with another sentiment label $sl \in \{\{-1, 0, 1\} - TL(m_i)\}$ correctly, $TSP(m_i, sl)$, will increase. Note that we have to guarantee that the sum of probabilities of m_i to be classified with all possible sentiment labels (i.e., $\sum_{sl \in \{-1,0,1\}} TSP(m_i, sl)$) is a constant. Therefore, for each of the other two possible sentiment labels for m_i, $sl \in \{\{-1, 0, 1\} - TL(m_i)\}$, the probability of m_i to be classified with sentiment label sl will increase as (lines 13, 14)

$$TSP(m_i, sl) \leftarrow TSP(m_i, sl) + 0.5 \cdot TP(m_i, TL(m_i)) \cdot TP(m_j, TL(m_j)) \cdot SP(m_i, m_j). \tag{6}$$

Similarly, if the sentiment polarities of m_i and m_j predicted by textual information based sentiment classifier are consistent with the prediction result of sentiment reversal, the

TABLE 4
Performance of Sentiment Reversal Prediction and Feature Contribution Analysis (%)

Features used	PR-AUC	F1 score
All Features	**81.63**	**80.91**
-Cascade Tree Features	59.62	63.91
-Diffusion Network Features	75.75	72.65
-Historical Behavior Features	76.76	75.41
Random Guess	50	50
+Cascade Tree Features	72.27	71.57
+Diffusion Network Features	55.74	56.20
+Historical Behavior Features	53.21	58.51

The best performance is highlighed in bold.

probability of m_i to be classified with sentiment label $TL(m_i)$ correctly will increase (line 17), and the probability of m_i to be classified with other sentiment labels will decrease (lines $18-21$).

In line 27, we employ L_∞ (as it is a max) norm of the difference of V_{TS} over consecutive iterations to be less than $\beta = 0.001$ as our iteration terminating condition, as was done in [38]. Finally, for Twitter message m_i, its decision function of the sentiment classification problem is displayed in line 29.

We would like to note that our proposed *SentiDiff* algorithm is a general framework, and different textual information based sentiment classifiers can be combined into this framework.

7 Experimental Evaluation

7.1 Experimental Settings

We split our tweet and retweet data with sentiment labels into training, validation and test sets. The training set is used to train textual information based sentiment classifiers and sentiment reversal prediction model, the validation set is used for testing the generalization performance of sentiment classifiers and sentiment reversal prediction model, and the test set is for blind evaluation. The percentage of dataset used as the training set is indicated by a variable pc. For example, $pc = 0.7$ indicates that 70 percent of overall repost cascade trees are treated as training set. Then half of the remaining cascade trees are used for validation set, and the rest of the cascade trees are for test set.

Note that our labeled dataset is highly imbalanced. Therefore, we apply Area Under the Precision-Recall Curve (PR-AUC) as our evaluation metric in our experiments, since PR-AUC is a more appropriate measurement for imbalanced data [39].

7.2 Experimental Results

7.2.1 Performance of Sentiment Reversal Prediction

In this section, we evaluate the effectiveness of our proposed sentiment reversal prediction model. In this experiment, we set pc as 0.8. The prediction performance results are displayed in Table 4. We can find that our prediction model is effective in predicting sentiment reversals, with classification PR-AUC of 81.63 percent when applying all feature sets. Furthermore, we investigate how each set of features (i.e., cascade tree, diffusion network and historical behaviors of users) can affect the prediction performance by considering only one at a time. We can find that the set of cascade tree features by itself can obtain the best performance, which confirms that the cascade tree features are the most important factors in the task of sentiment reversal prediction. The prediction performance of sentiment reversals is not satisfactory when we only consider diffusion network features or historical behavior features. This phenomenon can be explained as follows. The feature sets of diffusion network and historical behavior in Table 3 require both users a and b to be present in the collection of Twitter messages to train the model. In our dataset, new Twitter users appear overtime, and thus some users may be not collected in the training set. When predicting sentiment reversals between two new Twitter users, we cannot extract their diffusion network and historical behavior features from the training set, causing low prediction performance of sentiment reversals.

7.2.2 Effect of Fusing Textual and Sentiment Diffusion Information

In this section, we conduct experiments to show the incremental improvement after combining textual and sentiment diffusion information as described in Section 6, measured on the test set. Based on the training set, we first adopt several state-of-the-art textual information based sentiment analysis algorithms designed for Twitter messages to train textual information based sentiment classifiers, and then apply these classifiers into our proposed *SentiDiff* algorithm, where textual information based classifiers and sentiment reversal prediction model are combined to predict the sentiment polarity of each Twitter message. In this experiment, 80 percent of our labeled dataset is used as the training set. First, we provide a concise description of these state-of-the-art textual information based sentiment analysis algorithms used in this section.

- Topic-Based Mixture Model (TBM model) [40]: This method first applies Latent Dirichlet Allocation (LDA) model [41] to identify the topics for Twitter messages, and splits the training data into multiple subsets based on topic distributions. For each subset, a separate topic-specific sentiment model is trained by utilizing various features (including word n-grams, manual lexicons, emoticons). Finally, a sentiment mixture model is proposed by combining multiple topic-specific sentiment models. By considering topic information of Twitter messages, we achieve improvement in sentiment classification accuracy.
- Coooolll [42]: Coooolll is a deep learning method for Twitter sentiment analysis. This method first learns sentiment-specific word embedding (SSWE) in order to encode the sentiment information of text into the continuous representation of word, and a tailored neural network is designed to learn SSWE features from Twitter messages with sentiment labels. Then we can concatenate the SSWE features with the handcrafted features and build the sentiment classifier.
- Deep CNN-Based Model [43]: This method is another deep learning method where deep convolutional neural network is applied for Twitter sentiment analysis. It turns out that providing a deep neural network with good initialisation parameters can have a significant influence on the accuracy of the trained model. To

TABLE 5
PR-AUC (%) Results of Textual Information based Algorithms and Our *SentiDiff* Algorithm

Method	PR-AUC	Method	PR-AUC
TBM Model	69.70	TBM Model +Sentiment Diffusion	76.17
Coooolll	68.14	Coooolll +Sentiment Diffusion	74.87
Deep CNN-Based Model	70.89	Deep CNN-Based Model +Sentiment Diffusion	79.27
CS Model	71.25	CS Model +Sentiment Diffusion	78.28
FastText	73.87	FastText +Sentiment Diffusion	78.96
DeepWalk	65.37	DeepWalk +Sentiment Diffusion	72.49

address this issue, word embeddings are initialized using a neural language model first, and then a convolutional neural network is used to further refine the embeddings. Finally, the word embeddings and other parameters of the network obtained are used to initialize the network with the same architecture.

- Context-Sensitive Model (CS model) [44]: Different from traditional sentiment analysis algorithms which only consider the content of Twitter messages itself, this method also takes contextual information of Twitter messages into consideration. This method studies features from conversation-based context, author-based context and topic-based context about Twitter messages, and builds a local-feature sub neural network considering only local information from Twitter messages, and a contextualized feature sub network. Then these two sub networks are combined through a non-linear combination.
- FastText [45]: In the FastText model, each Twitter message is first represented as a set of n-gram features, and these features are then embedded by utilizing a embedding layer. After that, the embeddings of these n-gram features are averaged to form the final representation of the message, and are projected onto the output layer.
- DeepWalk [46]: DeepWalk aims to learn distributed vector representation for each node in a network. To apply the DeepWalk model into Twitter sentiment analysis task, we need to build a network based on text data at first. To this end, we represent each node as a Twitter message, and an edge between two messages exists if a message is the reposting of another message. Then the method proposed in [47] is utilized to intergrate textual content into network structures.

In our experiments, the values of all hyperparameters in textual information based sentiment analysis algorithms are set according to related literatures mentioned above. The experimental results are shown in Table 5. In Table 5, each line of the left part displays the PR-AUC result of a baseline textual information based sentiment analysis method, and each corresponding line of the right part shows the performance of sentiment classification after combining the baseline method with sentiment diffusion information by using

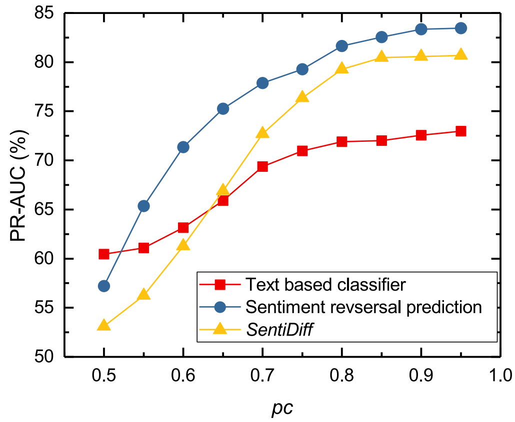


Fig. 13. Impact of amount of training data on our proposed algorithm.

SentiDiff algorithm. We can observe that for all 6 baseline sentiment analysis algorithms which only consider the textual information of Twitter messages, they can achieve significantly improved performance after combining textual and sentiment diffusion information in a supervised learning algorithm described in Section 6. After the combination, *SentiDiff* yields PR-AUC improvements between 5.09 and 8.38 percent on Twitter sentiment classification tasks, which verifies the effectiveness of *SentiDiff* by fusing textual and sentiment diffusion information for Twitter sentiment analysis. Among the six textual information based methods, the FastText model [45] can achieve the best performance when only considering the textual information of Twitter messages. However, the highest PR-AUC is achieved by the deep CNN-based model [43] after fusing textual and sentiment diffusion information.

7.2.3 Effect of Amount of Training Data

Here we conduct an experiment where we change the value of pc that determines the percentage of data used as training set from 0.5 to 0.95 with a step length 0.05, in order to test the effect of amount of training data. For each value of pc, we conduct experiments to compute PR-AUC of textual information based sentiment classifier, sentiment reversal prediction model and *SentiDiff* algorithm. In this section, we apply the deep CNN-based model [43] as textual information based sentiment classifier. The experimental results are displayed in Fig. 13.

From Fig. 13, we can observe that with the increase of pc, PR-AUC of textual information based sentiment classifier, sentiment reversal prediction model and *SentiDiff* algorithm increases gradually at first. When we further increase the value of pc, three PR-AUC curves enter a relatively stable state. We can also observe an interesting phenomenon from Fig. 13. When the performance of sentiment reversal prediction model is very poor ($pc < 0.65$), combining textual and sentiment diffusion information will have a negative influence on our Twitter sentiment analysis tasks. This is because if sentiment reversal prediction results are not reliable, sentiment diffusion information will decrease the probability of Twitter messages to be classified correctly by textual information based classifier described in Section 6.

8 Related Work

This paper combines ideas from sentiment diffusion and text sentiment analysis to analyze how sentiment diffusion information can help improve text sentiment analysis in social networks. Next, we introduce some related research work about sentiment diffusion in social networks, and text sentiment analysis in this section.

Most of previous work on sentiment diffusion is based on the structural features of networks, often modelling users as nodes in a graph and edges as relationships, invitations, interactions, etc [48]. Based upon the structural features of networks, a lot of problems have been solved, such as identifying influential users [49], discovering the most frequent diffusion patterns [31], and predicting public opinions about hot events [50]. When analyzing sentiment diffusion, one important issue is investigating how sentiment diffusion processes are influenced by various factors. In [16], the authors found that popular events are normally associated with increases in negative sentiment strength. The authors of [23] analyzed how sentiment flows through hyperlink networks, and observed that the sentiment polarity of a blog is influenced by its position within a cascade tree. In [19], the authors found that sentiment (positive or negative) of social media based content has correlation with information diffusion not only in terms of quantity but also propagation speed. The authors of [22] observed that users' opinions are influenced by people they are following, and proposed a learning algorithm to predict users' sentiment polarities in social networks. More recently, researchers have shifted their focus beyond network structures when analyzing sentiment diffusion. In [51], the authors considered user behaviors, user attributes as well as network structures to find influential users. In [52], the authors investigated how behaviors in physical social networks affect the sentiment propagation in online social networks, and proposed a novel theoretic framework to understand the characteristics of sentiment diffusion.

Text sentiment analysis, which identifies sentiment polarities expressed in text data, has become an important research field of opinion mining. Many machine learning based approaches have been applied to classify text sentiment polarity, such as the unsupervised learning based approaches [53], the supervised learning based approaches [54], the semi-supervised learning based approaches [55]. Recently, with the rapid growth of deep learning technologies, various deep neural networks have been developed and applied in text sentiment analysis tasks, such as deep convolutional neural networks [43], Phrase Recursive Neural Network (PhraseRNN) [56], Adaptive Recursive Neural Network (AdaRNN) [57] and attention-based long short term memory network [58]. However, these deep learning based methods are usually relatively slow both at train and test time, consequently limiting their use on large real-world corpus. To tackle this problem, the FastText model [45] has been proposed, where a sentence is represented as n-gram features, and these features are then embedded and averaged to form the hidden variable. Currently, the FastText model has been utilized for spatio temporal sentiment analysis of US election [4] and hate speech detection on Twitter [59], and proves to be very fast as well as achieving performance comparable to state-of-the-art methods. Now there are a lot of short texts on the Internet, such as tweets and product reviews. Compared with other text forms, they are much shorter, sparser, and noisier, which demands for a revisit of many fundamental technical problems for sentiment analysis [10], [60]. To solve this problem, the authors of [61] designed an efficient neural network in order to construct sentiment lexicons for short texts automatically. Then these sentiment lexicons can be utilized in both unsupervised and supervised sentiment analysis methods. In [12], the authors modeled the sentiment classification problem as a learning sentiment-specific word embedding issue, and designed three neural networks to effectively incorporate the supervision from text data with sentiment labels.

9 Conclusion and Future Work

Mining sentiment polarities expressed in Twitter messages is a meaningful while challenging task. Most of the existing solutions to Twitter sentiment analysis only consider textual information of Twitter messages, and cannot achieve satisfactory performance due to unique characteristics of Twitter messages. Although recent studies have shown that sentiment diffusion patterns have close relationships with sentiment polarities of Twitter messages, existing approaches basically only focus on textual information of Twitter messages, but ignore sentiment diffusion information. Inspired by recent work on fusion of knowledge from multiple domains, we take a first step towards combining textual and sentiment diffusion information to achieve better performance of Twitter sentiment analysis. To this end, we first analyze sentiment diffusion on Twitter by investigating a phenomenon called *sentiment reversal*, and find some interesting properties of sentiment reversals based on repost cascade trees and repost diffusion networks. We then build a sentiment reversal prediction model, and design a novel Twitter sentiment classification algorithm called *SentiDiff*. In *SentiDiff*, the interrelationships between textual information of Twitter messages and sentiment diffusion patterns are considered, and the textual information based sentiment classifier and the sentiment reversal prediction model are combined in a supervised learning framework. The experiments on real-world dataset demonstrate that our proposed *SentiDiff* algorithm can help state-of-the-art textual information based sentiment analysis algorithms achieve PR-AUC improvements between 5.09 and 8.38 percent.

In the future study, we plan to analyze how sentiment diffusion patterns differ in different topics, and consider the topic information of Twitter messages when fusing textual and sentiment diffusion information.

Acknowledgments

This work was supported by the National Key R&D Program of China (2017YFB1301100), the National Natural Science Foundation of China (61572060, 61772060, 61728201), and the CERNET Innovation Project (NGII20160316, NGII20170315).

References

[1] H. Li, L. Dombrowski, and E. Brady, "Working toward empowering a community: How immigrant-focused nonprofit organizations use twitter during political conflicts," in *Proc. ACM Conf. Supporting Groupwork*, 2018, pp. 335–346.

[2] D. Tang, B. Qin, F. Wei, L. Dong, T. Liu, and M. Zhou, "A joint segmentation and classification framework for sentence level sentiment classification," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 23, no. 11, pp. 1750–1761, Nov. 2015.

[3] H. Wang, D. Can, A. Kazemzadeh, F. Bar, and S. Narayanan, “A system for real-time twitter sentiment analysis of 2012 US presidential election cycle,” in *Proc. ACL Syst. Demonstrations*, 2012, pp. 115–120.

[4] D. Paul, F. Li, M. K. Teja, X. Yu, and R. Frost, “Compass: Spatio temporal sentiment analysis of US election what Twitter says!” in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2017, pp. 1585–1594.

[5] F. Bravo-Marquez, E. Frank, and B. Pfahringer, “Annotate-sample-average (ASA): A new distant supervision approach for Twitter sentiment analysis,” in *Proc. 22nd Eur. Conf. Artif. Intell.*, 2016, vol. 285, pp. 498–506.

[6] B. Pang, L. Lee, et al., “Opinion mining and sentiment analysis,” *Found. Trends® Inf. Retrieval*, vol. 2, no. 1/2, pp. 1–135, 2008.

[7] K.-L. Liu, W.-J. Li, and M. Guo, “Emoticon smoothed language models for twitter sentiment analysis,” in *Proc. 26th AAAI Conf. Artif. Intell.*, 2012, pp. 1678–1684.

[8] J. Zhao, L. Dong, J. Wu, and K. Xu, “MoodLens: An emoticon-based sentiment analysis system for chinese tweets,” in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 1528–1531.

[9] A. Go, R. Bhayani, and L. Huang, “Twitter sentiment classification using distant supervision,” *CS224N Project Report, Stanford*, vol. 1, no. 12, pp. 1–6, 2009.

[10] D.-T. Vo and Y. Zhang, “Target-dependent twitter sentiment classification with rich automatic features,” in *Proc. 24th Int. Conf. Artif. Intell.*, 2015, pp. 1347–1353.

[11] E. Cambria, “Affective computing and sentiment analysis,” *IEEE Intell. Syst.*, vol. 31, no. 2, pp. 102–107, Mar./Apr. 2016.

[12] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, “Learning sentiment-specific word embedding for Twitter sentiment classification,” in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, 2014, pp. 1555–1565.

[13] K. Schouten and F. Frasincar, “Survey on aspect-level sentiment analysis,” *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 3, pp. 813–830, Mar. 2016.

[14] J. Zhao and X. Gui, “Comparison research on text pre-processing methods on Twitter sentiment analysis,” *IEEE Access*, vol. 5, pp. 2870–2879, 2017.

[15] S. Symeonidis, D. Effrosynidis, and A. Arampatzis, “A comparative evaluation of pre-processing techniques and their interactions for Twitter sentiment analysis,” *Expert Syst. Appl.*, vol. 110, pp. 298–310, 2018.

[16] M. Thelwall, K. Buckley, and G. Paltoglou, “Sentiment in Twitter events,” *J. Assoc. Inf. Sci. Technol.*, vol. 62, no. 2, pp. 406–418, 2011.

[17] N. Du, Y. Liang, M. Balcan, and L. Song, “Influence function learning in information diffusion networks,” in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 2016–2024.

[18] M. Tsytsarau, T. Palpanas, and M. Castellanos, “Dynamics of news events and social media reaction,” in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 901–910.

[19] S. Stieglitz and L. Dang-Xuan, “Emotions and information diffusion in social media–sentiment of microblogs and sharing behavior,” *J. Manage. Inf. Syst.*, vol. 29, no. 4, pp. 217–248, 2013.

[20] Y. Fu, Y. Ge, Y. Zheng, Z. Yao, Y. Liu, H. Xiong, and J. Yuan, “Sparse real estate ranking with online user reviews and offline moving behaviors,” in *Proc. IEEE Int. Conf. Data Mining*, 2014, pp. 120–129.

[21] Y. Zheng, “Methodologies for cross-domain data fusion: An overview,” *IEEE Trans. Big Data*, vol. 1, no. 1, pp. 16–34, Mar. 2015.

[22] J. Tang and A. Fong, “Sentiment diffusion in large scale social networks,” in *Proc. IEEE Int. Conf. Consumer Electron.*, 2013, pp. 244–245.

[23] M. Miller, C. Sathi, D. Wiesenthal, J. Leskovec, and C. Potts, “Sentiment flow through hyperlink networks,” in *Proc. 5th Int. Conf. Weblogs Social Media*, 2011, pp. 550–553.

[24] X. Zhang, D.-D. Han, R. Yang, and Z. Zhang, “Users participation and social influence during information spreading on Twitter,” *PloS One*, vol. 12, no. 9, 2017, Art. no. e0183290.

[25] P. Nakov, A. Ritter, S. Rosenthal, F. Sebastiani, and V. Stoyanov, “Semeval-2016 task 4: Sentiment analysis in Twitter,” in *Proc. 10th Int. Workshop Semantic Eval.*, 2016, pp. 1–18.

[26] A. A. Dobrynin, R. Entringer, and I. Gutman, “Wiener index of trees: Theory and applications,” *Acta Applicandae Mathematicae*, vol. 66, no. 3, pp. 211–249, 2001.

[27] A. Anderson, D. Huttenlocher, J. Kleinberg, J. Leskovec, and M. Tiwari, “Global diffusion via cascading invitations: Structure, growth, and homophily,” in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 66–76.

[28] S. Tsugawa and H. Ohsaki, “Negative messages spread rapidly and widely on social media,” in *Proc. ACM Conf. Online Social Netw.*, 2015, pp. 151–160.

[29] E. Ferrara and Z. Yang, “Quantifying the effect of sentiment on information diffusion in social media,” *PeerJ Comput. Sci.*, vol. 1, 2015, Art. no. e26.

[30] D. M. Romero, B. Meeder, and J. Kleinberg, “Differences in the mechanics of information diffusion across topics: Idioms, political hashtags, and complex contagion on Twitter,” in *Proc. 20th Int. Conf. World Wide Web*, 2011, pp. 695–704.

[31] J. Niu, D. Wang, and M. Stojmenovic, “How does information diffuse in large recommendation social networks?” *IEEE Netw.*, vol. 30, no. 4, pp. 28–33, Jul./Aug. 2016.

[32] F. Wu, X.-Y. Jing, S. Shan, W. Zuo, and J.-Y. Yang, “Multiset feature learning for highly imbalanced data classification,” in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1583–1589.

[33] S. Wang, L. L. Minku, and X. Yao, “Resampling-based ensemble methods for online class imbalance learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 5, pp. 1356–1368, May 2015.

[34] Y. Tang, Y.-Q. Zhang, N. V. Chawla, and S. Krasser, “SVMs modeling for highly imbalanced classification,” *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)*, vol. 39, no. 1, pp. 281–288, Feb. 2009.

[35] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, “BING: Binarized normed gradients for objectness estimation at 300fps,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3286–3293.

[36] H. Kuang, L. L. Chan, C. Liu, and H. Yan, “Fruit classification based on weighted score-level feature fusion,” *J. Electron. Imaging*, vol. 25, no. 1, 2016, Art. no. 013009.

[37] C.-C. Chang and C.-J. Lin, “LIBSVM: A library for support vector machines,” *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, 2011, Art. no. 27.

[38] A. Mukherjee, B. Liu, and N. Glance, “Spotting fake reviewer groups in consumer reviews,” in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 191–200.

[39] J. Davis and M. Goadrich, “The relationship between precision-recall and ROC curves,” in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 233–240.

[40] B. Xiang and L. Zhou, “Improving Twitter sentiment analysis with topic-based mixture modeling and semi-supervised training,” in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, 2014, vol. 2, pp. 434–439.

[41] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *J. Mach. Learn. Res.*, vol. 3, no. Jan, pp. 993–1022, 2003.

[42] D. Tang, F. Wei, B. Qin, T. Liu, and M. Zhou, “Coooolll: A deep learning system for Twitter sentiment classification,” in *Proc. 8th Int. Workshop Semantic Eval.*, 2014, pp. 208–212.

[43] A. Severyn and A. Moschitti, “Twitter sentiment analysis with deep convolutional neural networks,” in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2015, pp. 959–962.

[44] Y. Ren, Y. Zhang, M. Zhang, and D. Ji, “Context-sensitive Twitter sentiment classification using neural network,” in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 215–221.

[45] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, “Bag of tricks for efficient text classification,” in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics: Volume 2, Short Papers*, 2017, vol. 2, pp. 427–431.

[46] I. Chaturvedi, S. Cavallari, E. Cambria, and V. Zheng, “Learning word vectors in deep walk using convolution,” in *Proc. 30th Int. Florida Artif. Intell. Res. Soc. Conf.*, 2017, pp. 323–328.

[47] C. Yang, Z. Liu, D. Zhao, M. Sun, and E. Y. Chang, “Network representation learning with rich text information,” in *Proc. 24th Int. Conf. Artif. Intell.*, 2015, pp. 2111–2117.

[48] A. Guille, H. Hacid, C. Favre, and D. A. Zighed, “Information diffusion in online social networks: A survey,” *ACM SIGMOD Record*, vol. 42, no. 2, pp. 17–28, 2013.

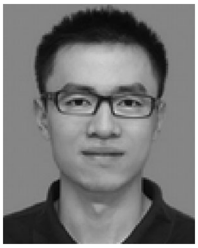
[49] W. Xu, W. Liang, X. Lin, and J. X. Yu, “Finding top-k influential users in social networks under the structural diversity model,” *Inf. Sci.*, vol. 355, pp. 110–126, 2016.

[50] X. Hao, H. An, L. Zhang, H. Li, and G. Wei, “Sentiment diffusion of public opinions about hot events: Based on complex network,” *PloS One*, vol. 10, no. 10, 2015, Art. no. e0140027.

[51] S. Aral and D. Walker, “Identifying influential and susceptible members of social networks,” *Sci.*, vol. 337, 2012, Art. no. 1215842.

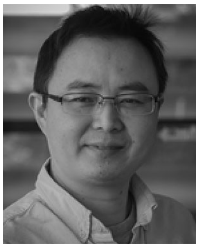
[52] O. Yagan, D. Qian, J. Zhang, and D. Cochran, “Conjoining speeds up information diffusion in overlaying social-physical networks,” *IEEE J. Sel. Areas Commun.*, vol. 31, no. 6, pp. 1038–1048, Jun. 2013.

[53] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Comput. Linguistics*, vol. 37, no. 2, pp. 267–307, 2011.
[54] F. Li, S. Wang, S. Liu, and M. Zhang, "SUIT: A supervised user-item based topic model for sentiment analysis," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, vol. 14, pp. 1636–1642.
[55] K. Kim and J. Lee, "Sentiment visualization and classification via semi-supervised nonlinear dimensionality reduction," *Pattern Recognit.*, vol. 47, no. 2, pp. 758–768, 2014.
[56] T. H. Nguyen and K. Shirai, "PhraseRNN: Phrase recursive neural network for aspect-based sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 2509–2514.
[57] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, and K. Xu, "Adaptive recursive neural network for target-dependent Twitter sentiment classification," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics (Volume 2: Short Papers)*, 2014, vol. 2, pp. 49–54.
[58] X. Zhou, X. Wan, and J. Xiao, "Attention-based LSTM network for cross-lingual sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 247–256.
[59] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep learning for hate speech detection in tweets," in *Proc. 26th Int. Conf. World Wide Web Companion*, 2017, pp. 759–760.
[60] C. dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *Proc. 25th Int. Conf. Comput. Linguistics: Technical Papers*, 2014, pp. 69–78.
[61] D. T. Vo and Y. Zhang, "Don't count, predict! An automatic approach to learning sentiment lexicons for short text," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics (Volume 2: Short Papers)*, 2016, vol. 2, pp. 219–224.



Lei Wang received the BE degree from the School of Computer Science and Engineering, Beihang University, in 2013. He is currently working toward the PhD degree in the School of Computer Science and Engineering, Beihang University. His current research interests include social network analysis, text mining, and information diffusion.

Jianwei Niu received the PhD degree in computer science from Beihang University, in 2002. He is a full professor with the School of Computer Science and Engineering, Beihang University. He has published more than 80 referred papers in conferences and journals such as IEEE INFOCOM, ACM Multimedia, ACM CHI, the *IEEE Transactions on Parallel and Distributed Systems*, the *IEEE Transactions on Mobile Computing*, the *IEEE/ACM Transactions on Networking*, the *IEEE Transactions on Industrial Informatics*, the *Journal of Parallel and Distributed Computing*, etc., and filed more than 30 patents in mobile and pervasive computing. He has served as editor of the *Journal of Internet Technology* and the *Journal of Network and Computer Applications*. His current research interests include mobile and pervasive computing, big data analysis. He is a senior member of the IEEE.



Shui Yu received the PhD degree from Deakin University, Victoria, Australia, in 2004. He is currently a full professor in the School of Software, University of Technology Sydney (UTS), NSW, Australia. Before joining UTS, he was an associate professor with the School of Information Technology, Deakin University, Victoria, Australia. He has published nearly 100 peer review papers in top journals and top conferences, such as the *IEEE Transactions on Parallel and Distributed Systems*, the *IEEE Transactions on Information Forensics and Security*, the *IEEE Transactions on Fuzzy Systems*, the *IEEE Transactions on Mobile Computing*, and IEEE INFOCOM. His research interests include security and privacy, networking, big data, and mathematical modelling. He is a senior member of the IEEE.