Online Burst Events Detection Oriented Real-Time Microblog Message Stream

Guozhong Dong[1, 2, *], Jun Gao[3], Liang Huang[4] and Chunlei Shi[1]

Abstract: The rapid spread of microblog messages and sensitivity of unexpected events make microblog become the public opinion center of burst events. Online burst events detection oriented real-time microblog message stream has become an important research problem in the field of microblog public opinion. Because of the large amount of real-time microblog message stream and irregular language of microblog message, it is important to process real-time microblog message stream and detect burst events accurately. In this paper, an online burst events detection framework is proposed. In this framework, abnormal messages are detected based on sliding time window and two-level hash table. Combined with event features, an online incremental clustering algorithm is used to cluster abnormal messages and detect burst events. Experimental results in the real-time microblog message stream environment show that our framework can be used in online burst events detection and has higher accuracy compared with other approaches.

Keywords: Burst event, abnormal message, microblog, message stream.

1 Introduction

Different from traditional news media, microblog allow users to broadcast short textual messages and express opinions using web-based or mobile-based platforms. Microblog provide the rapid communications of public opinion because of its immediacy, autonomy and interactivity. When emergency situation occurs, microblog play an important part in guidance and impetus. People can post short messages about emergency and share with microblog users using mobile services. Due to large number of people participating in conversation and discussions, some malicious messages may become burst messages or hot messages. It is important to detect and complete effective management on network popular feelings of microblog after emergency situation occurred. Considering millions of messages produced every day and large number of users, some emergency situations which cause a surge of a large number of relevant microblog messages are called burst events in this paper. Some microblog messages related to burst events may have a

[1] Henan University of Urban Construction, Pingdingshan, 467036, China.

[2] Singapore Management University, 188065, Singapore.

[3] Henan Science and Technology Information Research Institute, Zhengzhou, 450008, China.

[4] National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing, 100190, China.

[*] Corresponding Author: Guozhong Dong. Email: 20171010@hncj. edu.cn.

CMC. doi:10.32604/cmc.2019.05601 www.techscience.com/cmc

significant increase or become popular during a certain time interval which are called abnormal messages. These abnormal messages raise a question of immense practical value: Can we leverage abnormal messages for automated real-time burst events detection in microblog?

Unfortunately, this real-time burst events detection approach and system have not been solved by the existing work on Chinese microblog, such as Sina microblog. First of all, microblog's own trending topics list does not help much as it reports mostly those all-time popular topics, instead of the burst events in our work. Secondly, most topic modeling based works study the topics or events in microblog in a retrospective off-line manner, this real-time task is prohibitively challenging for existing algorithms because of real-time message stream processing and the accuracy of burst events detection.

The key research challenge that makes this problem difficult is how to solve the following two problems in real-time. In this paper, we investigate the problem of detecting burst events based on abnormal messages in microblog. It is necessary to detect and analyze burst events from microblog message stream in real-time by monitoring messages. To solve the challenging problems, we propose an online burst events detection framework based on abnormal messages (OBED) and implement an online burst events detection system. In OBED, abnormal messages are detected based on sliding time window and two-level hash table. Combined with event features, an online incremental clustering algorithm is used to cluster abnormal messages and detect burst events more accurately. Once burst events are detected, the system can summarize burst events and relevant abnormal messages.

2 Related work

Event detection has been studied for decades, with evolving interests on network attacks [Cheng, Xu, Tang et al. (2018)] and social media. As there are numerous research works focusing on it, we introduce the ones most related to our work, i.e., burst topics detection in social media. Here we categorize burst topics detection approaches into two categories: document-pivot approaches and feature-pivot approaches.

Anomaly detection technologies are used to detect abnormal documents in document-pivot burst topics detection approaches. Kasiviswanathan et al. [Kasiviswanathan, Melville, Banerjee et al. (2011)] propose a framework to detect emerging topics through the use of dictionary learning. They determine novel documents in the stream and subsequently identify cluster structure among the novel documents. The approach must set the number of topics in advance and cannot apply to detect burst topics online. Takahashi et al. [Takahashi, Tomioka and Yamanishi (2013)] apply a recently proposed change-point detection technique based on Sequentially Discounting Normalized Maximum Likelihood (SDNML) coding to detect abnormal messages and detect the emergence of a new topic from the anomaly measured through the model. Agarwal et al. [Agarwal, Ramamritham and Bhide (2012)] model emerging events detection problem as discovering dense clusters in highly dynamic graphs and exploit short-cycle graph property to find dense clusters efficiently in microblog streams. Alvanaki et al. [Alvanaki, Sebastian, Ramamritham et al. (2011)] presented the "en Blogue" system for emergent topic detection. En Blogue keeps track of sudden changes in tag correlations and presents tag pairs as emergent topics. Mathioudakis et al. [Mathioudakis and Koudas (2010)] identifies burst keywords and groups burst keywords into trends based on their co-occurrences. Cataldi et al. [Cataldi, Caro and Schifanella (2010)] formalize the keyword life cycle leveraging a novel aging theory intended to mine burst keywords and detect burst topics through keyword-based topic graph. They utilize an iterative method to compute user authority, which has high complexity and is not used in online burst events detection. Nguyen [Nguyen (2013)] introduce a novel concept of sentiment burst and employ a stochastic model for detecting bursts in text streams based on the work of Kleinberg [Kleinberg (2002)]. Then an effective method for evaluating and ranking events extracted using a combination of topic modeling is proposed. Cui et al. [Cui, Min, Liu et al. (2012)] study some event-related properties of hashtags, including temporal trends, authorships and pattern of texts. Based on event-related properties of hashtags, they examine the popular hashtags to discover breaking events. Li et al. [Li, Sun and Datta (2012)] propose “Twevent” system to detect events in twitter stream which can distinguish the realistic events from the noisy ones. Pei et al. [Pei, Lakshmanan and Milios (2013)] apply density-based clustering on evolving post network to identify the events. Wang et al. [Wang, Liu, Lin et al. (2013)] propose a system called SEA to detect events and conduct panoramic analysis on Weibo events from various aspects. Related works have been done in our previous works. For Twitter stream, Xie et al. [Xie, Zhu, Ma et al. (2014); Xie, Zhu, Jiang et al. (2013)] present a real-time system to provide burst event detection, popularity prediction, event summarization. For Chinese microblog stream, Shen et al. [Shen, Yang, Wang et al. (2015)] propose real-time burst topics detection oriented Chinese microblog stream. The method detect burst entities and cluster them to burst topics without requiring Chinese segmentation, which can obtain related messages and users at the same time, but the method does not extend to distributed framework. Previous works do not design distributed message stream processing framework to detect burst events oriented Chinese microblog stream. When message stream contains massive noise data, some approaches have low efficiency and accuracy. Our work presents an efficient framework to detect burst events in Chinese microblog message stream, which can be used in online burst events detection and has higher accuracy compared with other approaches.

3 Overview of OBED

The framework of OBED, shown in Fig. 1, comprises three functional layers, namely Message Stream Distribution, Abnormal Messages Detection and Burst Events Detection.

The “Message Stream Distribution” is designed to handle massive real-time microblog messages. As real-time messages keep coming in, it enables OBED to the distributed environment and constructs child message stream to abnormal messages detection node for further processing. The “Abnormal Messages Detection” computes each message’s influence series in hash table and determines whether it is an abnormal message in a given time window. The “Burst Events Detection” utilizes burst events detection algorithm combined with event features to cluster abnormal messages in each time window. As the number of abnormal messages is much smaller than the number of messages in message stream, the algorithm can decrease the computational complexity and detect burst events as early as possible.

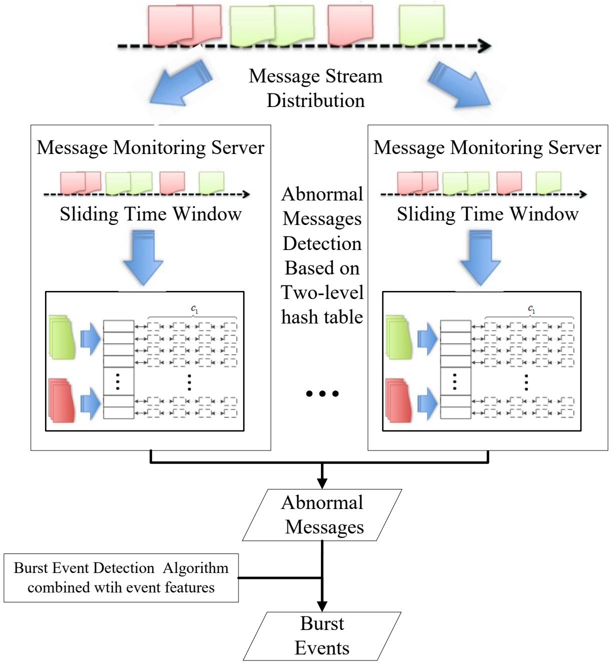
Message Stream
Distribution
Message Monitoring Server
Sliding Time Window
Message Monitoring Server
Sliding Time Window
Abnormal Messages Detection Based on Two-level hash table
Abnormal Messages
Burst Event Detection Algorithm combined wtih event features
Burst Events

Figure 1: The framework of OBED

4 Methods

4.1 Problem formulation

We first give some formal definitions and formulate our online burst events detection problem.

Definition 1. Microblog message. Each microblog message in microblog can be formalized as eight tuples:

$$m = (mid, root\_mid, uid, com\_num, ret\_num, post\_time, root\_time, content) \quad (1)$$

where mid is message id, $root\_id$ is original message id, uid is user id, com_num is the comment number of original message, ret_num is the retweet number of original message, $post\_time$ is the post time of message, $root\_time$ is the post time of original message, $content$ is the text content of message.

Definition 2. Microblog message stream. A microblog message stream according to post time of messages which can be define as

$$M = [m_1, m_2, \cdots, m_i, \cdots, m_N] \quad (2)$$

If $i < j$ and $i, j \in \{1, 2, \cdots, N\}$, the post time of m_i is smaller than m_j.

Definition 3. Sliding time window. The microblog message stream M can be divided into different time windows according to the post time of microblog message and time window size. Based on the concept of time window, the microblog message stream M can be formalized as

$$M = [W_1, \cdots, W_j, \cdots, W_L] \tag{3}$$

where W_j represents the message set of $j-th$ time window and $\sum_{j=1}^{L} |W_j| = M$. If W_L is current time window and K is the size of sliding window, sliding time window SW can be formalized as

$$SW = [W_{L-K+1}, \cdots, W_L] \tag{4}$$

Definition 4. Abnormal message. In sliding time window SW , the volume or velocity of abnormal message in current time window is large, but not before current time window.

Definition 5. Burst event. Burst event E can be formalized as

$$E = [M, U, F] \tag{5}$$

where M is the message set of burst event E . The messages in M are abnormal messages and semantically related. U is user set of burst event E . F is burst event feature set including event keywords, URL, etc.

Our task in this paper is, given a microblog message stream, to detect burst events from it as early as possible.

4.2 Message stream distribution

The number of Sina microblog message during one week is shown in Fig. 2. Through statistical analysis, Sina microblog produces about 50 million messages every day and the peak is about 2,000 messages per second. Because single message monitoring server can’t handle so large-scale real-time message stream, message stream distribution algorithm (Algorithm 1) is proposed to distribute message stream to different message monitoring servers. The algorithm can filter unlikely abnormal messages and effectively reduce data amount and complexity.

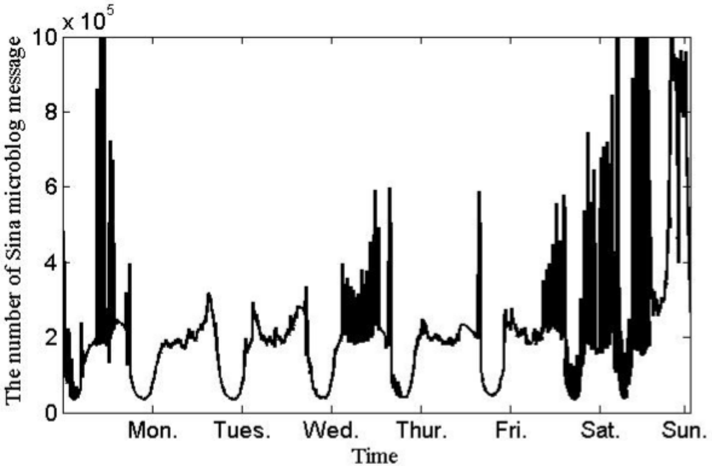

Figure 1: The number of Sina microblog message during one week

Algorithm 1. Message stream distribution algorithm

```
Input: $M$ is microblog message stream, $S$ is message monitoring server set , $PAT$ is filter condition to filter messages
Output: $M_s(1 \le s \le |S|)$ is the child message stream which is distributed to message monitoring server $S_s$
for each $m_i \in M$
    if $(m_i.com\_num + m_i.ret\_num) \ge PAT$
        $s = m_i.root\_mid \% |S|$
        distribute message $m_i$ to message monitoring server $S_s$
    else
        filter message $m_i$
end for
```

4.3 Burst events detection

In this section, we first detect abnormal messages based on our previous work [Dong, Wang, Yang et al. (2015)]. In our previous work, microblog message stream processing algorithm based on two-level hash table can generate the message influence series of each message node in two-level hash table. When current time window is full, a hash table copy signal is sent to abnormal messages detection thread and abnormal messages detection algorithm will detect abnormal messages in each time window.

In order to detect burst events, burst events detection model combined with event features is proposed. The event features are labeled by 40 volunteers through reading news section of Sina news[5]. The model processes abnormal messages detected by all abnormal messages monitoring server in each time window, which has two stages: abnormal messages pre-processing and abnormal messages clustering. In the stage of message pre-processing, user's nickname and illegal characters in text content are first removed. Text content is segmented into two blocks: hashtag text content and non-hashtag text content. Then we extract noun, verb and URL in each block as entity set. URL and extracted keywords in hashtag text content are added to feature entity set FE. If the entities in entity set match labeled event features, they are also added to FE. Other entities in entity set are added to common feature entity set NFE. So each abnormal message can be formalized as

$$am_i = (FE, NFE) \tag{6}$$

In the stage of abnormal messages clustering, abnormal messages clustering algorithm (Algorithm 2) is proposed.

[5] http://news.sina.com.cn/

Algorithm 2. Abnormal messages clustering algorithm

Input: AM is abnormal messages set in current time window, BE is burst events set, MT is similarity threshold

Output: updated burst events set BE

```
(1)  for $\forall am_i \in AM$
(2)    for $\forall E_j \in BE$
         /*compute the similarity between $am_i$ and $E_j$ */
(3)      $S_{i,j} = 2 * |am_i.FE \cap E_j.F| + |am_i.NFE \cap E_j.F|$
(4)        if $\exists S_{i,j} > MT$
(5)          add $am_i$ to $E_j$ and update $E_j$
(6)        else
(7)          create a new cluster and add to $BE$
(8)    end for
(9)    goto (1)
(10)  endfor
```

5 Experiments

We conduct extensive experiments to evaluate the performance of our framework for burst events detection and perform a validation by comparing it against state-of-the-art method. All of experiment are conducted on a Linux Server with twelve 2.3 GHz Intel Xeon E5-2630 processors, 32 GB RAM memory and running 64 bit Redhat 2.6.18. The programs are implemented with Java and C. Message stream process nodes can be divided into three types according to function: message stream distribution node, abnormal message detection node and burst events detection node. Message stream distribution node can construct message stream and distribute to abnormal message detection node for further processing. Abnormal messages detection node can detect and store abnormal messages in message stream. Burst events detection node can cluster abnormal messages and detect burst events.

5.1 Dataset

We selected Sina microblog as observation platform to detect burst events. Considering the characteristic of real-time and huge data, we developed distributed web crawler to collect Sina microblog data. The collected data set covered the period from January 24 to January 30 in 2015 which contains nearly over 410 million messages. Collected microblog messages are divided into two types: original messages and non-original messages. We make a statistical analysis on two message types. The ratio of non-original messages can be seen in Fig. 3. As shown in Fig. 3, the ratio of non-original messages is higher than 60%. If we only filter message stream based on mes-sage type, the scale of message are still quite large.

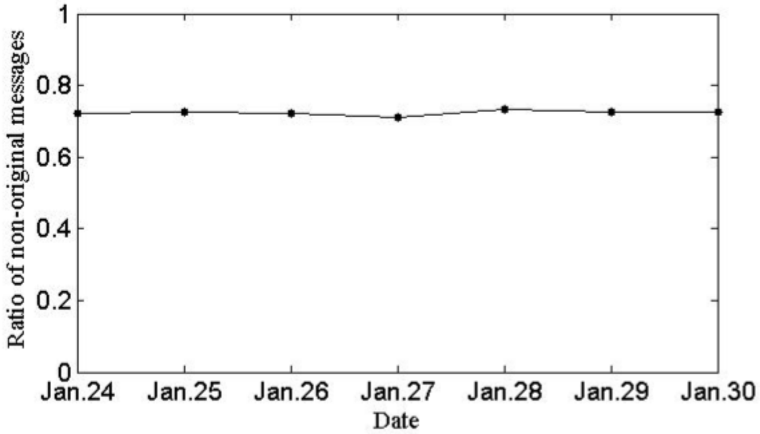

Figure 3: The proportion of non-original messages

Due to the lack of burst events detection dataset in Chinese microblog, 40 volunteers labeled burst events in collected dataset. The representative labeled burst events are shown in Tab. 1.

Table 1: The representative labeled burst events

The date of burst events	The keywords of burst events
January 24	Beijing, road, collapse
January 25	Hebei, girl, disappearance
January 26	Tianjin, bus, accident
January 26	Haikou, blackmail, post-deletion
January 26	New York, storm
January 26	Fuzhou, pervert
January 27	Haikou, godfather, rape, abortion
January 27	Guangdong, illegal, confiscate land
January 27	Luanchuan, violence, doctor
January 27	Zhangmo, drug, court

In order to analyze the influence distribution of messages and set filter condition to filter messages, the cumulative probability distribution of each message's original messages influence in labeled burst events is shown in Fig. 4.

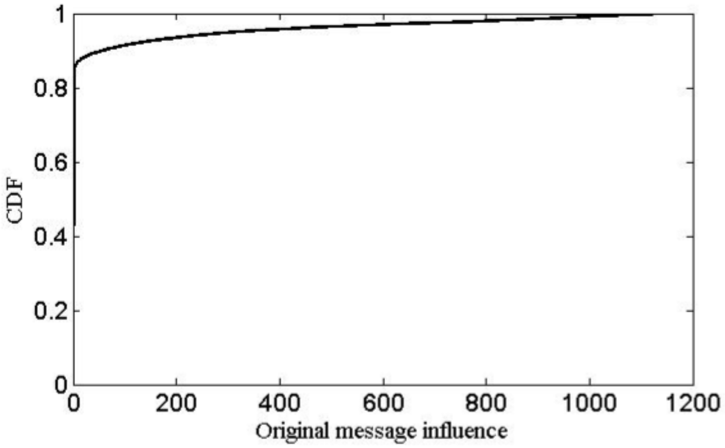

Figure 4: The distribution of each message's original messages influence

As shown in Fig. 4, the proportion of messages is about 90% when original messages influence is less than 100. So we set filter condition to filter messages to 100, which can filter unlikely abnormal messages and effectively reduce data amount and complexity.

5.2 Performance evaluation

In order to demonstrate the real time performance of our framework, time window ratio (TWR) is proposed to compute burst events detection efficiency in each time window, which can be define as follows:

$$TWR = \frac{\sum_{m \in M} t_m + \sum_{am \in AM} t_{am}}{T_{W_L}} \tag{7}$$

where t_m represents the cost time of message m storing in two-level hash table, t_{am} represents the cost time of abnormal message am clustering into burst events, T_{W_L} is the size of time window.

In this experiment, we set aging time $delay\_time = 86400s$, the threshold for popular messages $HT = 300$ in our previous work Dong et al. [Dong, Wang, Yang, Wang and Sun (2015)]. Besides, we set the size of sliding window $K = 6$, the size of time window $T_{W_L} = 300s$, similarity threshold $MT = 5$, which is decided by a larger number of experiments. The result of time window ratio is shown in Fig. 5. As shown in Fig. 5, time window ratio increases constantly with the time window increasing, but it maintains steady when the number of time window reach to a certain number. Besides, the time window ratio is smaller than 1, which shows that our framework can finish detecting burst events before next time window comes. So our framework can work online and detect burst events within one time window.

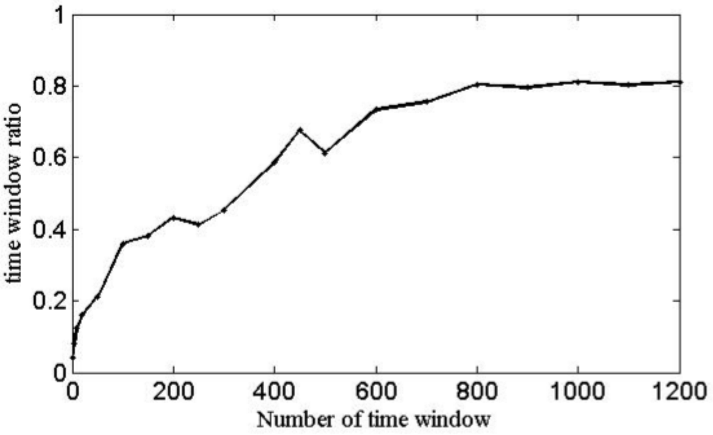

Figure 5: The result of time window ratio

5.3 Comparison to other methods

Furthermore, we pay more attention to the comparison with other representative previous related works: DBEH [Cui, Min, Liu et al. (2012)], BTDOM [Shen, Yang, Wang et al. (2015)] and TopicSketch [Xie, Zhu, Jiang et al. (2013)]. DBEH discovers breaking events based on event-related properties of hashtags. BTDOM detects burst entities in Chinese microblog and clusters them to burst topics based on high order co-clustering algorithm. TopicSketch utilizes a novel sketch-based topic model together with a set of techniques to detect burst topics in Twitter.

In the section, we contrast the performances of four approaches based on a com-mon metric, F-value. The comparison results of different approaches are shown in Figure 6. As the ratio of messages containing hashtags is low in Chinese microblog, DBEH can only detect burst events based on event-related hashtags. Besides, hashtags are usually post when events become hot events, which could cause DBEH not to detect burst events real-time. BTDOM detects burst topics based on burst Chinese character. However, it is sensitive to noise data and does not combine with event features. BTDOM may detect fake burst Chinese character and detect fake burst events in microblog message stream. TopicSketch can detect burst topics in microblog message stream in real time. As it does not combine with event features, it will detect non-event burst topics. Our framework can detect abnormal message based on efficient two-level hash table. Combined with event features, abnormal messages are aggregated as different clusters in increments, and each cluster represented a burst event, which can filter non-event abnormal messages and more accurately detect burst events. Furthermore, as the number of abnormal messages are much smaller than the number of message in message stream, our framework can decrease the computational complexity and detect burst events as early as possible.

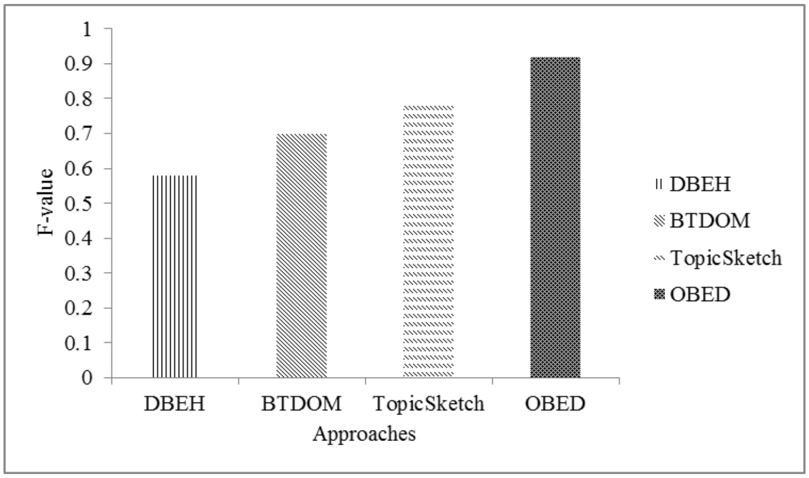

Figure 6: The comparison results of different approaches

6 Conclusions

In this paper, we propose OBED to detect burst events from large-scale microblog message stream. Because of the large amount of real-time microblog message stream and irregular language of microblog message, it is important to process real-time microblog message stream and detect burst events detection accurately. In our framework, abnormal messages are detected based on sliding time window and two-level hash table. Combined with event features, an online incremental clustering algorithm is used to cluster abnormal messages and detect burst events. Experimental results in the real-time microblog message stream environment show that our framework can be used in online burst events detection and has higher accuracy compared with other approaches.

Acknowledgement: This work was partially supported by Doctoral Scientific Research Foundation for Henan University of Urban Construction (No. Q2017013), the Key Research Development and Promotion Projects in Henan Province (Science and Technology) (Nos. 172102210174, 172102210105, 182402210025, 192102210258), the Ministry of Education Industry-University Collaborative Education Project.

References

Agarwal, M. K.; Ramamritham, K.; Bhide, M. (2012): Real time discovery of dense clusters in highly dynamic graphs: identifying real world events in highly dynamic environments. VLDB Endowment. *Proceedings of the VLDB Endowment*, vol. 5, no. 10, pp. 980-991.

Alvanaki, F.; Sebastian, M.; Ramamritham, K.; Weikum, G. (2011): EnBlogue: emergent topic detection in web 2.0 streams. *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pp. 1271-1274.

Cataldi, M.; Caro, L. D.; Schifanella, C. (2010): Emerging topic detection on Twitter based on temporal and social terms evaluation. *Proceedings of the Tenth International Workshop on Multimedia Data Mining*, pp. 1-10.

Cui, A.; Min, Z.; Liu, Y.; Ma, S.; Zhang, K. (2012): Discover breaking events with popular hashtags in twitter. *Proceedings of the 21st ACM international conference on Information and knowledge management*, pp. 1794-1798.

Cheng, J.; Xu, R.; Tang, X.; Sheng, V.; Cai, C. (2018): An abnormal network flow feature sequence prediction approach for DDoS attacks detection in big data environment. *Computers, Materials & Continua*, vol. 55, no. 1, pp. 95-119.

Dong, G.; Wang, B.; Yang, W.; Wang, W.; Sun, R. (2015): SAMD: A system for abnormal messages detection oriented microblog message stream. *Communications in Computer and Information Science*, vol. 562, pp. 113-124.

Kasiviswanathan, S. P.; Melville, P.; Banerjee, A.; Sindhwani, V. (2011): Emerging topic detection using dictionary learning. *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pp. 745-754.

Kleinberg, J. (2002): Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, vol. 7, no. 4, pp. 373-397.

Li, C.; Sun, A.; Datta, A. (2012): Twevent: segment-based event detection from tweets. *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pp. 155-164.

Mathioudakis, M.; Koudas, N. (2010): TwitterMonitor: trend detection over the Twitter stream. *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pp. 1155-1158.

Nguyen, T. (2013): Event extraction using behaviors of sentiment signals and burst structure in social media. *Knowledge & Information Systems*, vol. 37, no. 2, pp. 279-304.

Pei, L.; Lakshmanan, L. V. S.; Milios, E. (2013): KeySee: supporting keyword search on evolving events in social streams. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1478-1481.

Shen, G.; Yang, W.; Wang, W.; Yu, M. (2015): Burst topic detection oriented large-scale microblogs streams. *Journal of Computer Research and Development*, vol. 52, no. 2, pp. 512-521.

Takahashi, T.; Tomioka, R.; Yamanishi, K. (2013): Discovering emerging topics in social streams via link anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 120-130.

Wang, Y.; Liu, H.; Lin, H.; Wu, J.; Wu, Z.; Cao, J. (2013): SEA: A system for event analysis on chinese tweets. *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1498-1501.

Xie, R.; Zhu, F.; Ma, H.; Xie, W.; Lin, C. (2014): CLEar: A realtime online observatory for bursty and viral events. *Proceedings of the Vldb Endowment*, vol. 7, no. 13, pp. 1637-1640.

Xie, W.; Zhu, F.; Jiang, J.; Lim, E. P.; Wang, K. (2013): TopicSketch: Real-time bursty topic detection from twitter. *Proceedings of IEEE 13th International Conference on Data Mining*, pp. 837-846.