IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 8, 2016, accepted November 21, 2016, date of publication December 5, 2016, date of current version March 2, 2017.

Digital Object Identifier 10.1109/ACCESS.2016.2633721

Real-time Public Mood Tracking of Chinese Microblog Streams with Complex Event Processing

SI SHI[1], DAWEI JIN[1], AND GOH TIONG-THYE[2]
[1]School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan 430073 China
[2]School of Information Management, Victoria University of Wellington, Wellington 6012, New Zealand

Corresponding author: D. Jin (jdw@zuel.edu.cn)

This work was supported in part by the National Social Science Foundation of China under Grant 13CTJ003 and in part by the China Postdoctoral Science Foundation under Grant 2014M562025.

ABSTRACT There are not many real-time public mood tracking frameworks over social media streams at present. Real-time public mood tracking over microblogs becomes necessary for further studies with low-latency requirements. To address this issue, we propose a hierarchical framework for real-time public mood time series tracking over Chinese microblog streams using complex event processing. Complex event processing is able to handle high-speed and high-volume data streams. First, we transform microblogs into emotional microblog events through the text sentiment analysis. Then, we apply an online batch window technique to summarize the public mood in different periods. For the public mood time series, we use smoothing and trend following methods to find the rising or falling trends of the public mood. Finally, we apply the method to 6606 microblogs to verify its feasibility. The result demonstrates that the proposed model is not only feasible but also effective.

INDEX TERMS Public mood tracking, microblog stream, complex event processing, event stream processing.

I. INTRODUCTION

Recently mobile social applications have become immensely popular, and people have grown accustomed to expressing their feelings on social platforms. At the same time group activity analysis gains popularity through researchers [1]–[3]. To reap the maximum gain from the social data, high-speed real-time public mood analysis can enhance poll analysis [4], stock market prediction [5], public opinion analysis [6], personalized medical recommendation [7], and many other applications. However to the best of our knowledge there is no openly available framework to track public mood time series from social platforms in real time though there are real-time applications in the time-domain [8]–[10]. Current public mood tracking systems are unable to produce real-time analyses of the huge amount of social media data streams because the underlying data processing method using a "process-after-store" model and a passive way to process data are not adequate to process a variety of large continuous data streams [11]. To implement real-time public mood tracking, the event model, event processing method and statistical method should be improved in order to process stream data.

Our research addresses this problem using an approach based on complex event processing (CEP) technology. Complex event processing can provide an effective potential solution to processing the large amount of microblog data streams in real time. As a stream processing technology, complex event processing can perform event stream processing and pattern matching over different types of high-volume and high-speed data from different sources in real time and deliver the results to the requesting units as the basis of actions.

However, there are still several problems that stand in the way of using complex event processing in real-time public mood tracking over microblog streams. First, current complex event processing engines process events that are naturally "events" such as stock ticks [12], [13], RFID data [14], [15], blog article documents [16], and other data types. They are well organized and easy to express using objects or tuples. However, unstructured data with rich emotions, like text, image, speech, video, and other data types, cannot be further processed by most complex event processing engines directly. Second, data streams differ from the conventional stored relation model in the following aspects:

elements in data streams arrive online, so once an element from a data stream has been processed it cannot be retrieved easily [17]. Most actual tasks need to traverse data more than once. As for public mood tracking, all the tasks including daily or hourly public mood statistics, smoothing and pattern matching, need to traverse data. This kind of task is difficult to perform based on current complex even processing frameworks. Last but not least, state-of-the-art CEP-based applications focus more on the process of categorical data instead of ordinal data and numeric data. Analysis of time series over stream data is far from widespread but worth researching. Some techniques such as incremental computing in numerical calculation can greatly improve the performance of complex event processing, and studies of pattern matching over numerical values are rare but valuable.

In order to solve the problems mentioned above, we proposed a hierarchical framework combining the study of text sentiment analysis, public mood analysis and real-time processing to perform real-time public mood tracking. First we apply text sentiment analysis methods to each microblog in order to transform them into processable emotional microblog events. Then we summarize public mood events using a batch window technique. Last we use smoothing and trend following methods to find the trends and change points of public mood time series. The methods in each hierarchy can be changed according to the issue of interest.

We combine previously isolated techniques – text sentiment analysis, complex event processing and statistics – to perform public mood tracking over social media data. More specifically, this article makes the following contributions:

- We propose a real-time public mood tracking framework completely based on stream processing techniques, such as data stream batch and sliding windowing, stream data aggregate function, and stream data pattern matching.
- We make microblogs available by a complex event processing engine via applying a text sentiment analysis technique as an input adapter to transform microblogs into processable events;
- We achieve pseudo-traverse in stream data with event hierarchies in complex event processing;
- We perform online public mood time series analysis using existing data stream processing and complex event processing techniques.

This paper is organized as follows. In Section II, an overview of text sentiment analysis methods and complex event processing is given. In Section III we propose a hierarchical framework for real-time public mood time series tracking over microblog streams. In Section IV we apply a text sentiment analysis method to transform text in microblogs into processable events. Daily public mood information is summarized through a batch window technique in Section V. To find trends and change points smoothing and trend following methods are used in Section VI. The performance evaluation results and public mood tracking illustrations are shown in Section VII and in Section VIII. Section IX shows our conclusion.

II. RELATED WORK

A. TEXT SENTIMENT ANALYSIS

Sentiment analysis for text is mainly divided into the following steps:

The first step is the preprocessing, aiming at parsing the sentence or text, such as word segmentation, stop-word deletion, and part-of-speech tagging. As the first step towards further processing, currently there are many Chinese word segmentation tools such as NLPIR, Jieba, IKAnalyzer, HanLP.

The second step is feature extraction and feature selection, used to choose all kinds of appropriate signals to represent the attitude of the sentence or text. Keywords of the emotional labels (ELs) are the most intuitive ways to detect textual emotions [18]. Some studies mainly take emotional keywords into account [19], [20], ignoring semantic and syntactic information, yet some other researchers consider the linguistic information [21]–[23]. Also, different studies focus on different emotional states. For example, Wu *et al.* [21] use only three kinds of emotions (happy, unhappy and neutral) to perform their evaluation. Some are more complex, e.g. a sentiment analysis five major categories (anger, disgust, fear, joy, sadness). The studies by Danisman and Alpkocak [24], and Ruan *et al.* [25] use six emotional labels (happiness, sadness, disgust, anger, fear, surprise) to classify microblogs.

The last step is an emotion classifier, which means to use proper recognition algorithms to identify sentiments. These methods can be grouped into the following three categories: knowledge based, machine learning based and hybrid. The knowledge-based methods always use affect lexicons such as WordNet-affect [26] and a baseline algorithm to classify the text, for example, to tag the headline sentiment [27]. For machine learning methods, Yang *et al.* [20] combine the Support Vector Machine (SVM) with conditional random fields (CRF) to classify emotions at the document level. The Hidden Markov Model (HMM) is also a popular method when considering the mental state causing the emotion [28]. Similarly, Ruan *et al.* [25] use Markov logic networks (MLNs) to establish the model for Chinese microblogs. Besides these supervised learning algorithms, an unsupervised and greedy layer-wise algorithm – deep belief networks (DBN) – are used for sentiment analysis [29]. Hybrid approaches combine both or add different components to improve accuracy and refine the categories. Wu *et al.* [21] utilize a rule-based approach to extract semantics related to specific emotions for sentence level emotion mining. Yang *et al.* [30] propose a hybrid model that includes lexicon-keyword spotting, CRF based emotion cue identification and machine learning based methods including SVM, naive Bayesian and max entropy to classify emotions. Besides the above methods, dynamic structure-based neural networks [31] can also be used in text sentiment analysis.

B. COMPLEX EVENT PROCESSING

The concept of complex event processing was proposed by Luckham [32], and the development of the complex event processing model is complete [33], [34]. The complete theory includes its functional model, processing model, deployment model, interaction model, data model, time model, rule model and language model. The state-of-the-art research on complex event processing focuses on the standardization of event processing language [35], distributed processing [36], [37], and performance improvement in certain scenarios [38]. There are many CEP prototypes like Cayuga [39], SASE [40], [41] and established commercial software like Esper by EsperTech, BusinessEvents and Stream Base by TIBCO and Sybase complex event processing systems.

In terms of application, CEP has been widely used in many domains such as medical treatment in smart hospitals [15], traffic congestion detection [42], power management for wireless data transmission [43], alarming event detection for coal mine safety [44], intrusion detection for network security [45], [46], vehicular context perception [47], etc. From the listed applications we can see that recent applications of complex event processing are limited to well-organized data types, as it is seldom used in unstructured data. Most of the listed studies focus on categorical data processing or numeric data processing using simple aggregate functions like SUM, AVERAGE, MAXIMUM, MINIMUM, and COUNT.

Research about statistical methods for complex event processing is rare. Useful statistical techniques for data streams include smoothing, generalized additive models, trend following, and classification methods [40]. Statistical techniques for data streams are used to detect technical chart patterns in stock time series data [13]. After feeding a stock data stream into a complex event processing engine, a smoothing technique is applied and extremes are identified. Then queries are applied to extremes to find out technical patterns. They provide a framework to handle stream data time series but not a direct method for real-time public mood tracking.

III. A HIERARCHICAL FRAMEWORK

Complex event processing abstracts lower level events into higher level events according to event abstraction rules. The set of events and event rules describing the extracted relationship between events constitutes event hierarchies. Event hierarchies indicate how data streams are extracted to support information flows. In CEP the event hierarchies organize event abstractions into levels [32]. In this paper we propose a hierarchical framework based on three-layer event hierarchies (Fig. 1) and different stream processing techniques are used in this hierarchical framework (Fig. 2). In this paper, we use Esper,[1] an open source solution, to perform event series analysis and complex event processing.

Before using complex event processing to track public mood, microblogs should be transformed into emotional microblog events because complex event processing is unable to handle unstructured data like text in microblogs. In preparation for event processing, we first apply text sentiment analysis method to identify the emotion contained in microblogs.

A single emotional microblog event is usually meaningless. Only when a considerable number of events is abstracted will it make sense. As for public mood tracking, a single emotional microblog cannot express the complete public mood. We use a batch window technique to summarize the public mood events from emotional microblog events.

The stream of summarized public mood events can be seen as a public mood time series. We first smooth the public mood time series. Then the smoothed public mood events will be used to detect the trends in public mood time series.

[1] http://www.espertech.com/

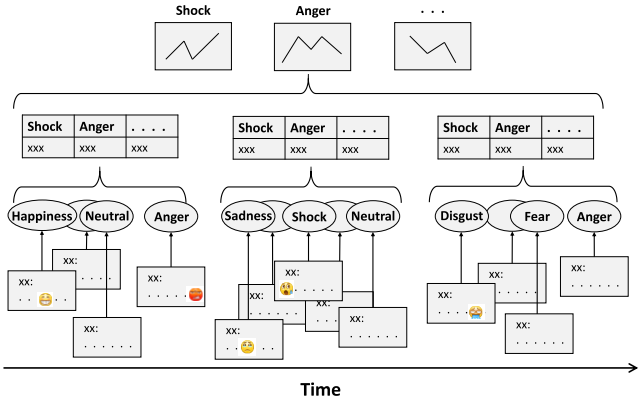


FIGURE 1. Online public emotion time series tracking hierarchical framework.

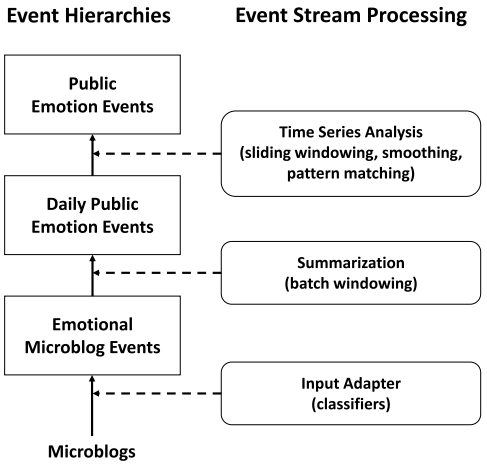


FIGURE 2. Event stream processing and event hierarchies in hierarchical framework.

IV. TEXT SENTIMENT ANALYSIS

The first step towards our framework is to use text sentiment analysis methods to transform original microblog data into emotional microblog events. In this paper, we identified six emotions proposed by Ekman and Friesen [48]: happiness, anger, sadness, fear, disgust, and surprise.

Because microblogs usually express a mixture of several emotions, we did not classify each microblog as a single-class emotion but multi-class emotions. For each emotion type, we applied classification methods to identify whether the text expresses this kind of emotion or not.

A. DATASET

Text sentiment analysis can be seen as a classification problem. In this paper, we consider it single-class classification. To illustrate our framework, we used the dataset from Shujutang about the topic "a female driver was beaten by a male driver" from 21:33 on May 3 (the first topic-related microblog) to 23:07 on June 9, 2015.[2] Sensitive information has been deleted from the dataset.

TABLE 1. Detail of labeled dataset.

Emotion type	Number	Proportion
Happiness	695	10.52%
Anger	864	13.08%
Sadness	80	1.21%
Fear	168	2.54%
Disgust	2962	44.84%
Surprise	110	1.67%
Total	6606	-

The dataset contains 7,258 Chinese microblogs. After deleting advertisements and other unrelated microblogs, we retained 6,606 original Chinese microblogs. The data was marked by two annotators separately to ensure reliability. If the labels of the same microblog were different, we marked it a third time as the final label. The labeled dataset is shown in TABLE I. Note that one microblog may express no, single or multiple emotions.

B. PREPROCESSING AND FEATURE EXTRACTION

After performing preprocessing such as deleting pre-defined structures like hashtags (#topic #), highlighted titles (【title】), and usernames (@usernames), we used NLPIR[3] from the Chinese Academy of Sciences as a word segmentation tool. Then we extracted emotional features according to emotion lexicons and emoticons (like [怒], [anger]) as text emotion features. The emotion lexicons include the sentiment lexicons of Tsinghua University and HowNet,[4] a Chinese affective lexicon ontology from Dalian University of Technology,[5] NTUSD from National Taiwan University,[6] and topic specific words.

C. CLASSIFICATION ALGORITHMS

Text sentiment analysis can be considered as a classification problem. For each type of emotion, we compared several common classification algorithms to find an effective and efficient algorithm for an event adapter.

The experiment was performed using Weka data mining software[7] from the University of Waikato. For each emotion type classification, this paper randomly selected 75% of the dataset as the training dataset and 25% as the test dataset.

TABLE 2. Experimental results on datasets.

Emotion types	Algorithm	Precision	Recall	F-Measure
Happiness	Decision Trees	0.895	0.684	0.775
	Naive Bayes	**0.910**	0.695	0.788
	SVM	0.871	**0.736**	**0.798**
Anger	Decision Trees	**0.887**	0.435	0.584
	Naive Bayes	0.833	0.394	0.535
	SVM	0.831	**0.454**	**0.587**
Sadness	Decision Trees	**0.867**	**0.542**	**0.667**
	Naive Bayes	**0.867**	**0.542**	**0.667**
	SVM	**0.867**	**0.542**	**0.667**
Fear	Decision Trees	**0.600**	0.225	**0.327**
	Naive Bayes	0.450	0.225	0.300
	SVM	0.400	**0.250**	0.308
Disgust	Decision Trees	0.755	**0.668**	**0.709**
	Naive Bayes	0.721	0.612	0.662
	SVM	**0.797**	0.635	0.707
Surprise	Decision Trees	**0.786**	**0.786**	**0.786**
	Naive Bayes	0.706	0.429	0.533
	SVM	**0.786**	**0.786**	**0.786**

By observing and comparing the results in TABLE II, we made the following observations. First, different classification algorithms are suitable for detecting different emotions. For the happiness and anger SVM performs the best, for fear and disgust a decision tree is best, for surprise a decision tree and SVM perform well but naïve Bayes performs badly, and for sadness there is no significant performance difference between the three algorithms. Second, some emotions such as happiness, disgust and surprise are easy to detect according to emotion lexicons and emoticons (f-measure larger than 0.700) because there are usually distinct emotion lexicons and emoticons in such microblogs. Detection of anger and sadness is of medium difficulty. However Fear is difficult to identify because it has no distinct emotion lexicons and emoticons and classification can only be performed from semantic aspects.

For specific topic, people may focus on specific emotions. The performance of other emotions has little influence on analysis of the concerned topic. For common topic, the method used in this paper is evolving to satisfy future demand. The proposed hierarchical framework is loosely coupled, and the sentiment analysis approaches can be improved without affecting other layers.

V. BLIC MOOD WINDOWING

A single emotional microblog event is meaningless for public mood tracking. Only when many events are summarized can the public mood be recognized. For batch processing, it is easy to select stored data of a specific period in batches.

[2]http://www.datatang.com/data/47261
[3]http://ictclas.nlpir.org/
[4]http://www.keenage.com/
[5]http://ir.dlut.edu.cn/
[6]http://www.datatang.com/data/44317/
[7]http://www.cs.waikato.ac.nz/ml/weka/

However, in stream processing once the data have been processed it will be dropped. Summarization should be done whenever an event occurs.

A. RANDOM EVENT MODEL

In complex event processing an event is an object that represents or records an activity that happens, or is thought of as happening [32]. Simple events are events that cannot be divided, and are instantaneous. Complex events are abstracted from their sub-events and therefore represent their member events.

General public mood analysis systems used to use statistical methods and the proposed real-time public mood tracking system inherits that method. To suit public mood time series tracking better we used the random event definition described in [49] as our event model. From this point of view an event can be seen as a multivariate random variable.

A simple random event is an object or a tuple $E_{\text{simple}} = (A, X_1, \ldots, X_{P_A})$ where A is a random event type, and $X_A, \ldots, X_{P_A}$ are random event attributes and also component random variables.

A complex random event is an object or a tuple $E_{\text{simple}} = (A, X_1, \ldots, X_{P_A}, E_1 \ldots, E_{M_A})$ where A is a random event type, $X_1, \ldots, X_{P_A}$ are random event attributes and also component random variables, and $E_1, \ldots, E_{M_A}$ are member events (either complex or simple) constituting the complex random event.

An event extracted directly from microblog text sentiment analysis is a simple random event.

$$E_{ME} = (\text{'}MicroblogEmotionEvent\text{'}, \cdots, Happiness, Anger, Sadness, Fear, Disgust, Surprise)$$

Emotion attributes in each event represent whether the microblog expresses the corresponding emotion. If so the attribute values 1; otherwise it values 0. For more accurate analysis the emotion attributes can also be a decimal values between 0 and 1 according to specific issues.

B. AGGREGATION IN TIME BATCH WINDOW

The time batch window buffers emotional microblog events and releases them every specified time interval (Fig. 3).

In the time batch window we used aggregate functions COUNT and SUM to summarize public mood information. Information of different emotion types can be aggregated according to the value of each emotion type attribute. In CEP, public mood events can be obtained using the following aggregate functions:

Popularity:
COUNT(*)
Emotion summary:
SUM(Happiness| Anger| Sadness| Fear| Disgust| Surprise)
Proportion:
SUM(Happiness| Anger| Sadness| Fear| Disgust| Surprise) /COUNT(*)

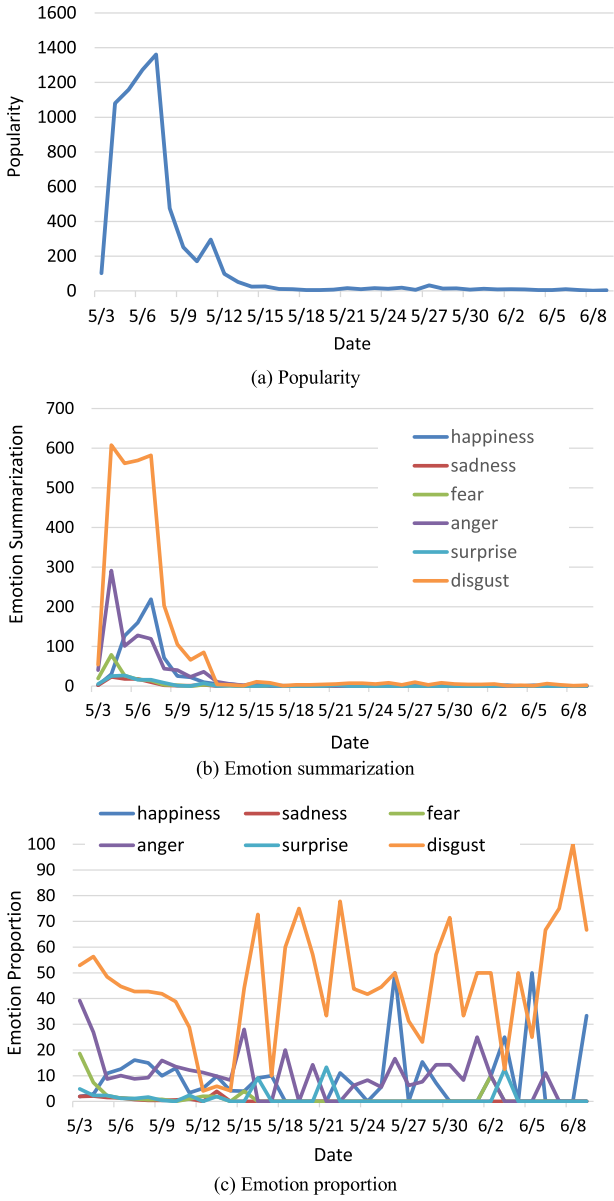


FIGURE 3. Daily public mood time series from May 3 to June 9. (a) Popularity. (b) Emotion summarization. (c) Emotion proportion.

C. DAILY AND HOURLY PUBLIC MOOD RESULTS

Daily public mood time series are shown in Fig. 3. From Fig. 3 (a) we can demonstrate the changing popularity of the topic. After the topic occurred it became popular rapidly and then the growth slowed down. From May 7 its popularity dropped sharply until May 10. On May 11 the popularity rose a little and then dropped to almost none. The emotion summarization shown in Fig. 3 (b) shows the emotions changing over time. At first on May 3 people felt disgusted, angry, scared and surprised at hearing the news that a female driver was fiercely beaten by a male driver. When people calmed down a little and felt less disgusted, angry or scared, they felt disgusted, angry and surprised again on May 5 at

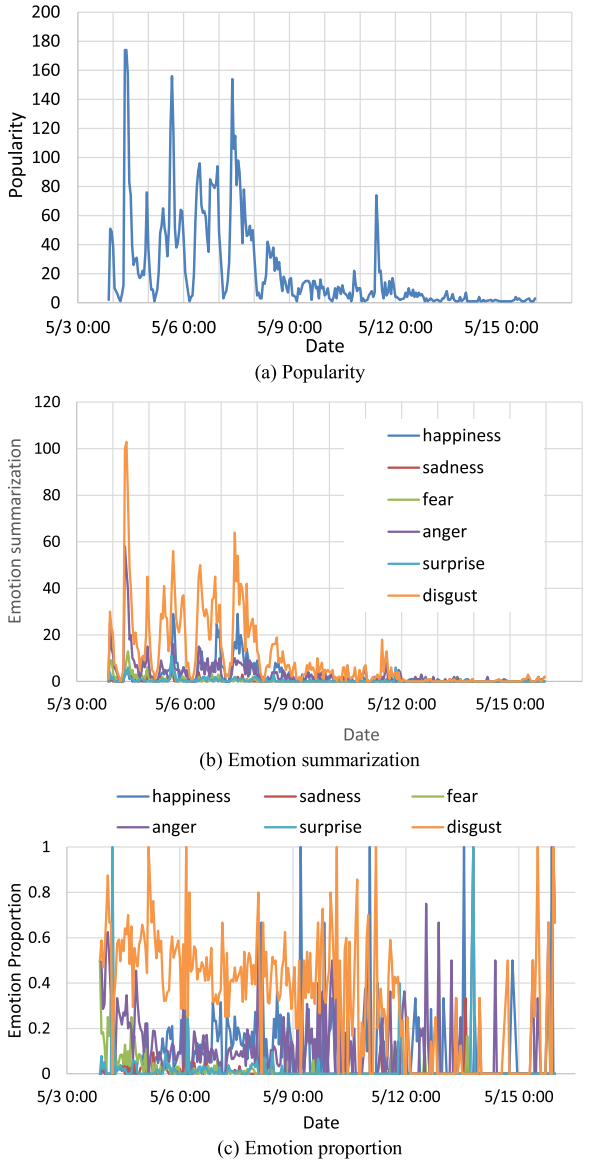


FIGURE 4. Hourly public mood time series from May 3 to May 15. (a) Popularity. (b) Emotion summarization. (c) Emotion proportion.

hearing that the female driver was beaten due to her dangerous driving and almost causing a car accident. Then people calmed down again until the female driver said that she was hurrying to do charity work on May 10. People thought she was lying and felt disgusted and angry again. At last people lost their interest in this topic. The emotion change is in accordance with news occurrences. Fig. 3 (c) shows that people mostly felt disgusted, angry and surprised about the whole affair.

In addition to daily public mood information, complex event processing can also extract hourly public mood information (shown in Fig. 4). As shown, hourly public mood is more volatile than daily public mood. Like market data, the changes of popularity and emotion summarization also have a significant intraday effect. Among the six emotions, people felt disgusted the most. People also felt angry and surprised.

VI. PUBLIC MOOD TIME SERIES TRACKING

A. SLIDING WINDOW

Processing data streams differs from the conventional stored relation model in the following aspects [17]:

- Elements in data streams arrive online, so for current element z_t it is impossible to use its future elements $z_u, u > t$;
- Data streams are potentially unlimited in size;
- Once an element from a data stream has been processed it cannot be retrieved easily.

To process stream data we used a sliding window to limit the number of events considered by a query. Unlike a batch window, sliding window is a moving window extending to the specified time interval in the past. Events arriving at the sliding window will be buffered until they leave the window. Once an event leaves the window it will be dropped and cannot be used any more.

As future elements cannot be used in stream processing, we used a sliding window to limit the past N elements as the basis of approximation. Thus stream processing lags $N/2$ terms compared with batch processing.

B. SMOOTHING

Smoothing is a common processing method to capture important patterns and filter out noise or fine scale structures/rapid phenomena. There are two goals of smoothing: (1) detecting actual change points with less lag time, and (2) ignoring noisy change points. To find important trends in the public mood time series, we implemented two common smoothing methods: simple moving average (SMA) and exponentially weighted moving average (EWMA).

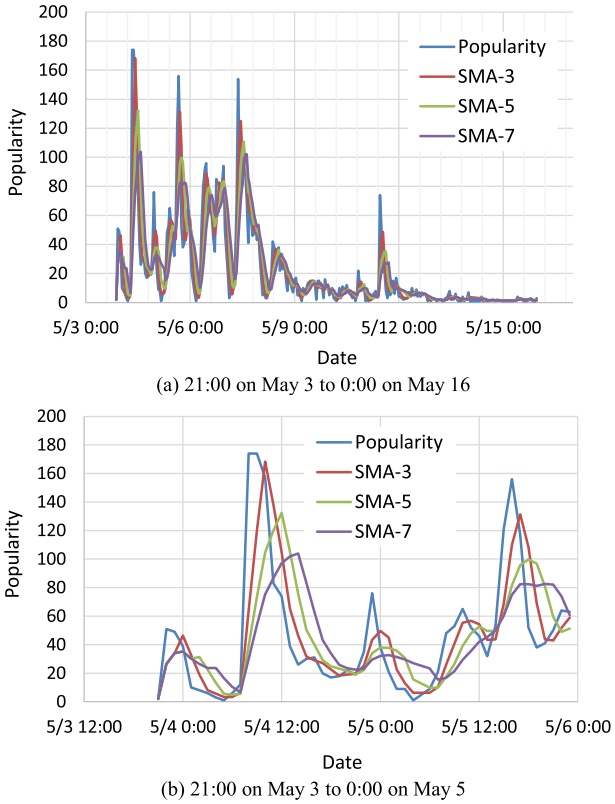
1) SIMPLE MOVING AVERAGE

Simple moving average (also called arithmetic moving average) is a common smoothing technique calculated by adding the value for a number of time periods and then dividing this total by the number of time periods. For real-time processing, a simple moving average makes it easy to perform incremental computing for lower latency. The formula to compute a simple moving average value is as follows.

$$\begin{aligned} S_t &= \frac{z_{t-N+1} + z_{t-N+2} + \cdots + z_{t-1} + z_t}{N} \\ &= S_{t-1} + \frac{z_t - z_{t-k}}{N} \end{aligned} \tag{1}$$

where S_t is the smoothed value at time period t, z_t is the value of emotion at time period t, and N is the size of time window.

The original popularity and the smoothed values of simple moving average when $N = 3$, $N = 5$, and $N = 7$ are shown in Fig. 5. There are significant lags in smoothed value using a simple moving average. With the increase of time window, extremes occur much later in smoothed values of simple moving averages. In other words, the simple moving average cannot find out change points in time.



(a) 21:00 on May 3 to 0:00 on May 16

(b) 21:00 on May 3 to 0:00 on May 5

FIGURE 5. Original values and smoothing results of simple moving average over popularity time series. (a) 21:00 on May 3 to 0:00 on May 16. (b) 21:00 on May 3 to 0:00 on May 5.

2) EXPONENTIALLY WEIGHTED MOVING AVERAGE

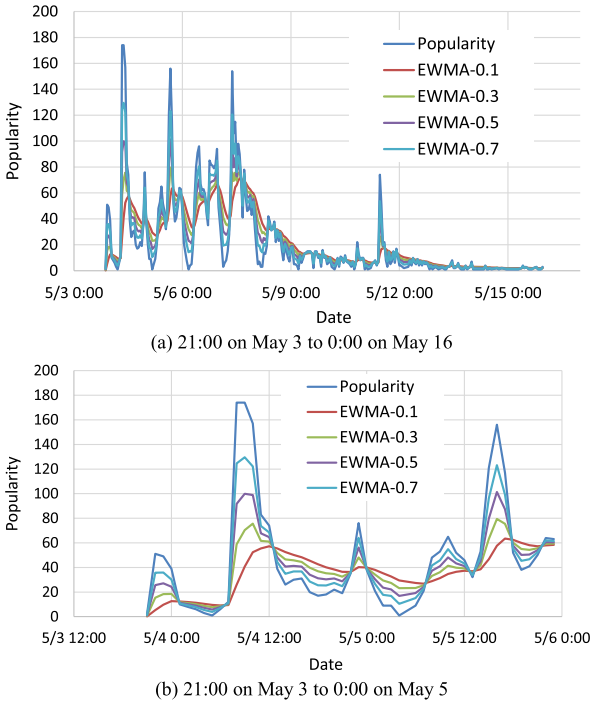
In simple moving average all elements earn the same weight. The last (very recent) element has no more influence on the variance than less recent elements. In exponentially weighted moving average more recent elements have greater weight on the variance. The formula to compute an exponentially weighted moving average value is as follows.

$$S_t = \alpha z_t + (1 - \alpha) S_{t-1} \tag{2}$$

where S_t is the smoothed value at time period t, z_t is the value of emotion at time period t, and α is a constant smoothing factor between 0 and 1 representing the degree of weighting decrease.

The original popularity and the smoothed values of exponentially weighted moving average when $\alpha = 0.1$, $\alpha = 0.3$, $\alpha = 0.5$, and $\alpha = 0.7$ are shown in Fig. 6. There are no significant lags in smoothed value using exponentially weighted moving average. With the decrease of α, slight fluctuations in original popularity trends are smoothed.

In summary, both simple and exponentially weighted moving average can detect important patterns and filter out fine-scale structures. The main difference is that there are serious lags using simple moving average while exponentially weighted moving average performs without significant lags. Therefore exponentially weighted moving average is more suitable for public mood time series smoothing.



(a) 21:00 on May 3 to 0:00 on May 16

(b) 21:00 on May 3 to 0:00 on May 5

FIGURE 6. Original values and smoothing results of exponentially weighted moving average over popularity time series.(a) 21:00 on May 3 to 0:00 on May 16. (b) 21:00 on May 3 to 0:00 on May 5.

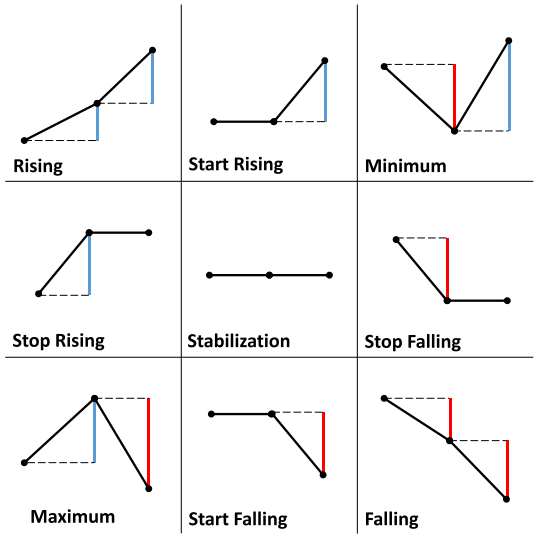


FIGURE 7. Local microstructure of trend patterns.

C. TREND FOLLOWING

After smoothing each data window using the kernel smoother, the toolkit checks whether there is any extreme inside the window. If any extreme exists at time t of smoothen window the algorithm checks the raw data window again to detect the real occurrence of the extreme and its real value.

Local microstructure of public mood trends fall into nine general categories: start rising, rising, stop rising, start falling, falling, stop falling, maximum, minimum, and stabilization (Fig. 7). The local microstructures can be specified using

TABLE 3. Conditions of local microstructures.

	a1 = a2	a1 = a2	a1 > a2
a2 < a3	Rising	Start rising	Minimum
a2 = a3	Stop rising	Stabilization	Stop falling
a2 > a3	Maximum	Start falling	Falling

a sequence pattern with constrains in complex event processing engine. Then complex event processing engine will perform pattern matching to follow public mood trends.

The constrains for trend following are as followed:

Sequence: a1 -> a2 -> a3

Event type: MicroblogEmotionEvent

Topic: FemaleDriverBeingBeaten

Time window: 3

Conditions: (TABLE III)

Once detected interest patterns, complex event processing engine will publish to all subscribers. Once a smoothed emotional event has received, there will be one and only one detected pattern notified to subscribers. Subscribers will receive the latest local microstructure of interest topic.

VII. PERFORMANCE EVALUATION

Real-time public mood tracking is difficult to perform because of high-speed data from numerous mobile devices and high-volume data from worldwide social media websites. Therefore, we simulated the performance of each scene to evaluate whether the proposed system can handle the large number of social media data streams produced by many users. To evaluate the proposed framework, we recorded the local microstructure patterns, replayed the dataset described in Section IV according to their timestamp, and recorded the outcome.

The server in the test bed is a x64-based PC with Intel(R) Core(TM) i5-3470 CPU and 8,080MB of RAM running Windows 7 Ultimate operating system. The java version is 1.8.0_60 and the Esper version is 5.4.0.

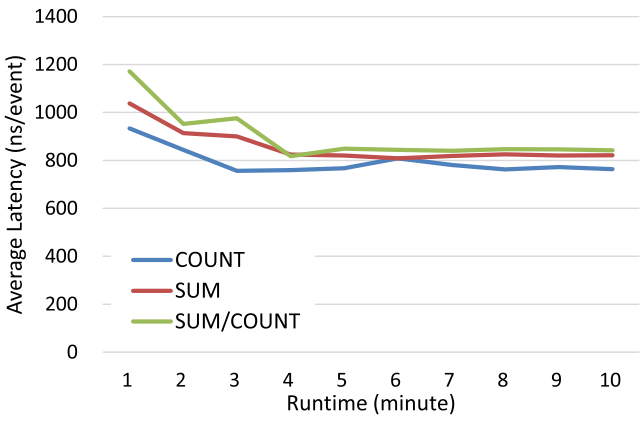


FIGURE 8. Average latency per event of different aggregate functions.

Fig. 8 shows the average latency for one event of three kinds of queries in batch windowing over time. We can see that the COUNT operator has the lowest latency, the SUM operator takes second place, and the SUM/COUNT operator has the highest latency. The reason is that the COUNT operator always adds one when receiving an event of interest, the SUM operator adds a number according to the received event, and the SUM/COUNT operator not only computes COUNT and SUM but also does the division. It shows that calculating popularity over microblog streams has the lowest latency, while emotion summarization is second, and proportion the highest. Though there are differences between operators, about 800 ns latency per event is negligible.

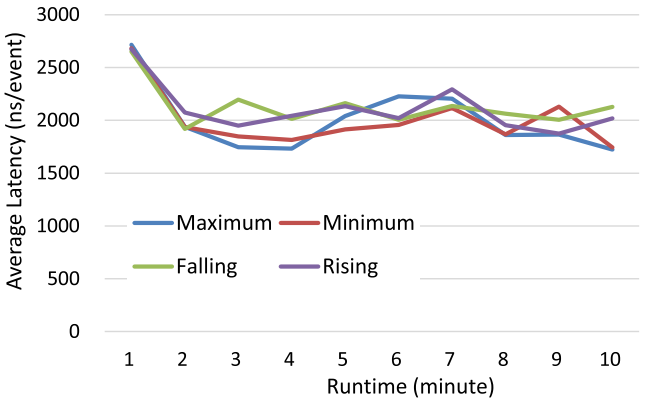


FIGURE 9. Average latency per event of different motion trend patterns.

Average latency per event of different motion trend patterns over time is shown in Fig. 9. Four common interest patterns (maximum, minimum, falling, and rising) were evaluated. The latency of the four patterns is nearly the same at 2000 ns when stable. Compared with aggregate operators in Fig. 8 (a), the latency of pattern matching is much larger but still unnoticeable.

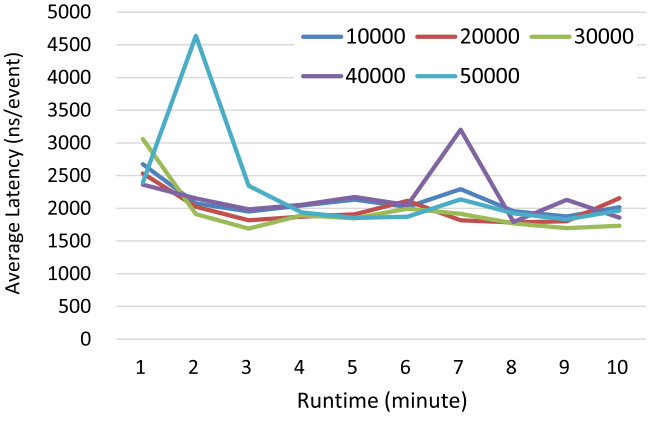


FIGURE 10. Average latency per event of different numbers of rules.

In addition to operators, the number of subscribed topics can also have an influence on events' average latency. The average latency per event of different numbers of rules (Fig. 10) shows that when there are more subscribed topics, the average latency per event becomes more unstable over time. Different numbers of rules have the same latency when stable.

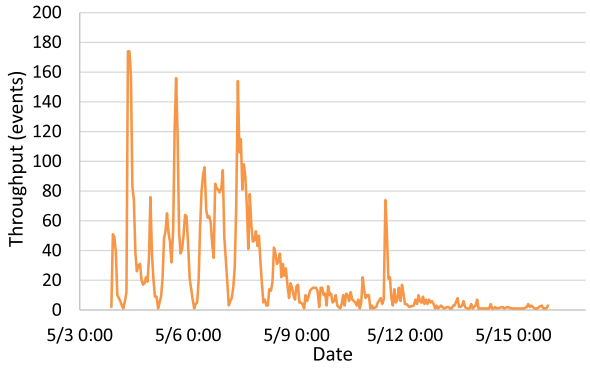


FIGURE 11. Throughput per hour about the topic "a female driver was beaten by a male driver" over time.

The occurrence of microblog events about the concerned topic is hard to predict. The throughput of the proposed system (Fig. 11) is the same as Popularity in Fig. 4. (a). We can conclude that when a large amount of microblogs arrive, the proposed framework can process them without missing microblogs even in rapidly changed situation.

VIII. REAL-TIME TRACKING OF PUBLIC MOOD

To illustrate our framework, we received a microblog feed on the topic "a migrant worker knelt to withdraw" from October 26 to 29 in 2016 via Sina Weibo API Weibo Open Platform.[8] In this illustration, we use EWMA-0.5 ($\alpha = 0.5$) as the smoother.

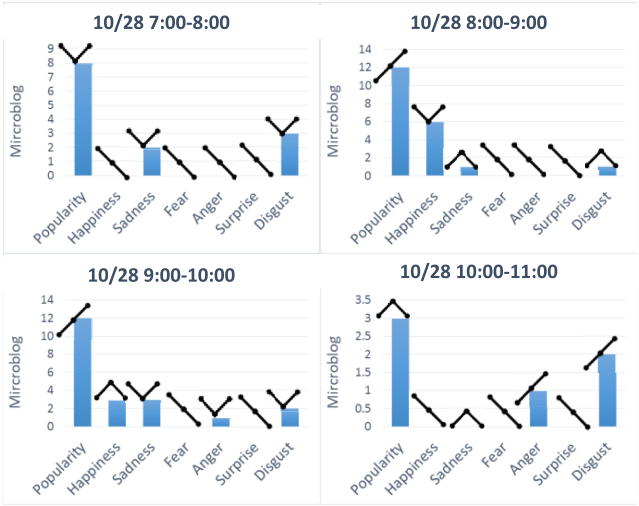


FIGURE 12. Part of real-time public mood results.

Parts of real-time public mood results are shown in Fig. 12. From the charts we observe that from 7:00 – 10:00 on 10/28, the popularity first rose then fell. According to microblogs on the topic, the public mood changed over time.

[8]http://open.weibo.com/

IX. CONCLUSION

With the popularity of social media, mobile social media applications and worldwide social media systems are producing large amounts of emotional text data. Many scenarios need real-time public mood tracking such as poll analysis, stock market prediction, and public opinion. However, analysis based on traditional batch processing techniques cannot handle high-speed and high-volume social data streams, let alone sentiment analysis and public mood tracking. To resolve the issue, this paper presented a novel approach using complex event processing to perform real-time public mood tracking over Chinese microblog streams. The approach is based on a hierarchical framework. It takes emotion-recognized microblog events as input. Then it applies a batch window to summarize the data followed by emotion summarization and emotion proportion. Lastly trend following is achieved using a sliding window, smoothing and pattern matching.

Although the proposed approach can process large amounts of emotional stream data in real time, it has several limitations. First, as input of the framework, the accuracy and efficiency of text sentiment analysis have a great effect on the performance of the approach. The method used in this paper is evolving to satisfy future demand. The proposed hierarchical framework is loosely coupled, and the sentiment analysis approaches can be improved without affecting other layers. Sentiment analysis can assess different aspects like syntax and semantics using advanced techniques like artificial neural networks and deep learning. Second, the time series analysis approaches used in this paper are rather simple. Advanced data analysis techniques can be used but they must accommodate the need for low latency and the characteristics of stream data. Last but not least, analysis of population data, emotion summarization, and emotion proportion can be improved further. Public mood analysis can be performed through various groupings such as regional analysis, emotion leader analysis, and contagious analysis.

The proposed framework is the first step towards real-time public mood tracking in social media. It combines the study of text sentiment analysis, public mood analysis and real-time processing. It could lead to a new area of research development. For example, text sentiment analysis, natural language processing for random internet language, sentiment analysis for emerging topics, and improvements in accuracy and efficiency of sentiment analysis are worth studying. Public mood analysis over real-time social media big data is still in its infancy. Real-time processing over unstructured data and time series data is still difficult to perform. The future work of such areas will bring benefits in real-time big data processing and analysis.

REFERENCES

[1] G. Fortino, W. Russo, C. Mastroianni, C. E. Palau, and M. Esteve, "CDN-supported collaborative media streaming control," *IEEE Multimedia Mag.*, vol. 14, no. 2, pp. 60–71, Apr. 2007.

[2] Y. Zhang, M. Chen, S. Mao, L. Hu, and V. Leung, "CAP: Community activity prediction based on big data analysis," *IEEE Netw.*, vol. 28, no. 4, pp. 52–57, Jul./Aug. 2014.

[3] Y. Zhang, "GroRec: A group-centric intelligent recommender system integrating social, mobile and big data technologies," *IEEE Trans. Serv. Comput.*, vol. 9, no. 5, pp. 786–795, Sep. 2016.
[4] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith, "From tweets to polls: Linking text sentiment to public opinion time series," in *Proc. 4th Int. AAAI Conf. Weblogs Social Media From (ICWSM)*, 2010, pp. 122–129.
[5] J. Arafat, M. A. Habib, and R. Hossain, "Analyzing public emotion and predicting stock market using social media," *Amer. J. Eng. Res.*, vol. 2, no. 9, pp. 265–275, 2013.
[6] T. Lansdall-Welfare, V. Lampos, and N. Cristianini, "Effects of the recession on public mood in the UK," in *Proc. 21st Int. Conf. Companion World Wide Web*, 2012, pp. 1221–1226
[7] Y. Zhang, M. Chen, D. Huang, D. Wu, and Y. Li, "iDoctor: Personalized and professionalized medical recommendations based on hybrid matrix factorization," *Future Generat. Comput. Syst.*, vol. 66, pp. 30–35, Jan. 2016.
[8] A. Andreoli, R. Gravina, R. Giannantonio, P. Pierleoni, and G. Fortino, "SPINE-HRV: A BSN-based toolkit for heart rate variability analysis in the time-domain," in *Wearable and Autonomous Biomedical Devices and Systems for Smart Environment*. Berlin, Germany: Springer, 2010, pp. 369–389.
[9] R. Gravina and G. Fortino, "Automatic methods for the detection of accelerative cardiac defense response," *IEEE Trans. Affect. Comput.*, vol. 7, no. 3, pp. 286–298, Jul./Sep. 2016.
[10] Y. Zhang, M. Qiu, C.-W. Tsai, M. M. Hassan, and A. Alamri, "Health-CPS: Healthcare cyber-physical system assisted by cloud and big data," *IEEE Syst. J.*, to be published.
[11] M. Stonebraker, U. Çetintemel, and S. Zdonik, "The 8 requirements of real-time stream processing," *ACM SIGMOD Rec.*, vol. 34, no. 4, pp. 42–47, 2005.
[12] A. Demers, J. Gehrke, M. Hong, M. Riedewald, and W. White, "Towards expressive publish/subscribe systems," in *Proc. Int. Conf. Extending Database Technol.*, vol. 3896. 2006, pp. 627–644.
[13] M. N. Bandara, R. M. Ranasinghe, R. W. M. Arachchi, C. G. Somathilaka, S. Perera, and D. C. Wimalasuriya, "A complex event processing toolkit for detecting technical chart patterns," in *Proc. IEEE 29th Int. Parallel Distrib. Process. Symp. Workshops (IPDPSW)*, May 2015, pp. 547–556.
[14] C. Bettini *et al.*, "SASE: Complex event processing over streams," in *Proc. 3rd Biennial Conf. Innov. Data Syst. Res. (CIDR)*, 2007.
[15] W. Yao, C. H. Chu, and Z. Li, "Leveraging complex event processing for smart hospitals using RFID," *J. Netw. Comput. Appl.*, vol. 34, no. 3, pp. 799–810, 2011.
[16] M. Hong, A. Demers, J. Gehrke, C. Koch, M. Riedewald, and W. White, "Massively multi-query join processing in publish/subscribe systems," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2007, pp. 761–772.
[17] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and issues in data stream systems," in *Proc. 21st ACM SIGMOD-SIGACT-SIGART Symp. Principles Database Syst. (PODS)*, 2002, pp. 1–16.
[18] E. C.-C. Kao, C.-C. Liu, T.-H. Yang, C.-T. Hsieh, and V.-W. Soo, "Towards text-based emotion detection a survey and possible improvements," in *Proc. Int. Conf. Inf. Manage. Eng. (ICIME)*, 2009, pp. 70–74.
[19] P. Subasic and A. Huettner, "Affect analysis of text using fuzzy semantic typing," *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 4, pp. 483–496, Aug. 2001.
[20] C. Yang, K. H.-Y. Lin, and H.-H. Chen, "Emotion classification using Web blog corpora," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell.*, Nov. 2007, pp. 275–278.
[21] C.-H. Wu, Z.-J. Chuang, and Y.-C. Lin, "Emotion recognition from text using semantic labels and separable mixture models," *ACM Trans. Asian Lang. Inf. Process.*, vol. 5, no. 2, pp. 165–183, 2006.
[22] A. Agrawal and A. An, "Unsupervised emotion detection from text using semantic and syntactic relations," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol. (WI-IAT)*, vol. 1. Dec. 2012, pp. 346–353.
[23] S. Shaheen, W. El-Hajj, H. Hajj, and S. Elbassuoni, "Emotion recognition from text based on automatically generated rules," in *Proc. IEEE Int. Conf. Data Mining Workshop*, Dec. 2014, pp. 383–392.
[24] T. Danisman and A. Alpkocak, "Feeler: Emotion classification of text using vector space model," in *Proc. AISB Conv. Commun. Interact. Social Intell.*, vol. 1. 2008, pp. 1–91.
[25] D. Ruan, X. Ping, and K. Gao, "Markov logic networks based emotion classification for Chinese microblogs," *Int. J. Intell. Inf. Database Syst.*, vol. 9, no. 2, pp. 197–211, 2016.
[26] C. Strapparava and A. Valitutti, "WordNet Affect: An affective extension of WordNet," in *Proc. LREC*, vol. 4. 2004, pp. 1083–1086.
[27] F.-R. Chaumartin, "UPAR7: A knowledge-based system for headline sentiment tagging," in *Proc. 4th Int. Workshop Semantic Eval.*, 2007, pp. 422–425.
[28] D. T. Ho and T. H. Cao, "A high-order hidden Markov model for emotion detection from textual data," in *Proc. 12th Pacific Rim Knowl. Acquisition Workshop*, 2012, pp. 94–105.
[29] K. Tang, L. Fei-Fei, and D. Koller, "Learning latent temporal structure for complex event detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1250–1257.
[30] H. Yang, A. Willis, A. De Roeck, and B. Nuseibeh, "A hybrid model for automatic emotion recognition in suicide notes," *Biomed. Inform. Insights*, vol. 5, no. 1, pp. 17–30, 2012.
[31] D.-W. Jin and J. Lu, "An improved determination approach to the structure and parameters of dynamic structure-based neural networks," *Appl. Math. Comput.*, vol. 215, no. 7, pp. 2787–2797, 2009.
[32] D. Luckham, *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Reading, MA, USA: Addison-Wesley, 2002.
[33] L. J. Fülöp, G. Tóth, R. Rácz, J. Pánczél, T. Gergely, and Á. Beszédes, "Survey on complex event processing and predictive analytics," in *Proc. 5th Balkan Conf. Informat.*, 2010, pp. 26–31.
[34] G. Cugola and A. Margara, "Processing flows of information: From data stream to complex event processing," *ACM Comput. Surv.*, vol. 44, no. 3, pp. 1–62, 2012.
[35] M. Ma, P. Wang, J. Yang, and C. Li, "OntoEvent: An ontology-based event description language for semantic complex event processing," in *Web-Age Information Management*, vol. 9098. New York, NY, USA: Springer, 2015, pp. 448–451.
[36] M. Kumarasinghe and F. Arbab, "VISIRI—Distributed complex event processing system for handling large number of queries," in *Proc. Int. Conf. Coordination Models Lang.*, 2015, pp. 230–245.
[37] S. Jayasekara, S. Kannangara, T. Dahanayakage, I. Ranawaka, S. Perera, and V. Nanayakkara, "Wihidum: Distributed complex event processing," *J. Parallel Distrib. Comput.*, vols. 79–80, pp. 42–51, May 2013.
[38] M. Cui, C. Zhang, Y. Su, and Y. Ji, "Feedback-based reduplicate complex event processing in IoT," in *Proc. Int. Conf. Big Data Comput. Commun.*, 2015, pp. 243–256.
[39] A. Demers, J. Gehrke, B. Panda, M. Riedewald, V. Sharma, and W. White, "Cayuga: A general purpose event monitoring system," in *Proc. 3rd Biennial Conf. Innov. Data Syst. Res.*, 2007, pp. 412–422.
[40] J. Agrawal, Y. Diao, D. Gyllstrom, and N. Immerman, "Efficient pattern matching over event streams," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2008, pp. 147–160.
[41] H. Zhang, Y. Diao, and N. Immerman, "On complexity and optimization of expensive queries in complex event processing," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, 2014, pp. 217–228.
[42] F. Terroso-Sáenz, M. Valdés-Vela, C. Sotomayor-Martinez, R. Toledo-Moreo, and A. F. Gómez-Skarmeta, "A cooperative approach to traffic congestion detection with complex event processing and VANET," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 914–929, Jun. 2012.
[43] Y. Xiao, W. Li, M. Siekkinen, P. Savolainen, A. Ylä-Jääski, and P. Hui, "Power management for wireless data transmission using complex event processing," *IEEE Trans. Comput.*, vol. 61, no. 12, pp. 1765–1777, Dec. 2012.
[44] C. Bo, Z. Peng, Z. Da, and C. Junliang, "The complex alarming event detecting and disposal processing approach for coal mine safety using wireless sensor network," *Int. J. Distrib. Sens. Netw.*, vol. 2012, pp. 1–12, Oct. 2012.
[45] K. Jayan and A. K. Rajan, "Preprocessor for complex event processing system in network security," in *Proc. 4th Int. Conf. Adv. Comput. Commun.*, 2014, pp. 187–189.
[46] C. Jun and C. Chi, "Design of complex event-processing IDS in Internet of Things," in *Proc. 6th Int. Conf. Meas. Technol. Mechatronics Autom.*, 2014, pp. 226–229.
[47] F. Terroso-Sáenz, M. Valdés-Vela, F. Campuzano, J. A. Botia, and A. F. Skarmeta-Gómez, "A complex event processing approach to perceive the vehicular context," *Inf. Fusion*, vol. 21, no. 1, pp. 187–209, 2015.
[48] P. Ekman and W. V. Friesen, "Constants across cultures in the face and emotion.," *J. Pers. Soc. Psychol.*, vol. 17, no. 2, pp. 124–129, 1971.
[49] P. H. Tendick, L. Denby, and W.-H. Ju, "Statistical methods for complex event processing and real time decision making," *Wiley Interdiscipl. Rev. Comput. Statist.*, vol. 8, no. 1, pp. 5–26, 2015.

SI SHI received the M.Sc. degree major in information management and information systems, and minor in accountancy in 2014. She is currently pursuing the master's degree in management science and engineering with the Zhongnan University of Economics and Law, Wuhan, China.

She has participated in multiple scientific research projects at national and ministerial level. Her research interests include real-time computing, complex event processing, data stream processing, financial time series, high-frequency trading, and algorithmic trading.

Ms. Shi was a recipient of National Scholarship, the Outstanding Graduate Award, and First-class Academic Scholarship.

DAWEI JIN received the bachelor's and master's degrees in management science from the Zhongnan University of Economics and Law, Wuhan, China, and the Ph.D. degree in computer science from Wuhan University, Wuhan.

He visited the School of Computer Science, Deakin University, Australia, funded by the China Scholarship Council from 2011 to 2012. He is currently a Professor and a Ph.D. Supervisor with the Zhongnan University of Economics and Law. His current research interests include financial information engineering, financial high-frequency data analysis, high-frequency trading, and algorithmic trading.

GOH TIONG-THYE received the B.Sc. and M.Sc. degrees in electrical engineering from Ohio State University, the GDipFM degree from the Singapore Institute of Management, the M.B.A. degree (Hons.) from the University of Manchester and Wales, and the Ph.D. degree in information systems from Massey University. He is currently a Senior Lecturer with the School of Information Management, Victoria University of Wellington, New Zealand. He is also the Wenlan Professor with the Zhongnan University of Economics and Law. His research focuses on the relationship between technologies, people, and society. In particular, his research involves the understanding of social media, cognitive and emotion computing, e-commerce, analytics, learning science, and user behavior.