Event Detection on Twitter by Mapping Unexpected Changes in Streaming Data into a Spatiotemporal Lattice

Zubair Shah and Adam G. Dunn

Abstract—Many applications seek to make sense of high volume streaming data from social media by identifying spatiotemporal patterns. Events, representing topics that emerge and decay over time, are detected by monitoring for changes in the language being used, but typical approaches do not consider the localisation of events in cities and countries, and within hours, days, and weeks. This work develops and evaluates a new approach to event localisation and ranking that can be applied to Twitter data streams. The proposed approach models the use of language in tweets per city per hour to produce a model that can be used to detect the magnitude of unexpected changes in the use of the language. The approach uses a spatiotemporal lattice structure and a method for traversing between hours, days, and weeks, as well as cities, regions, and countries to identify anomalies in the language used across millions of tweets. The output is a ranked list of events comprising a list of tweets posted within a location and period of time, and characterized by language features of interest. The approach was implemented and tested by comparing events detected across five example domains (suicide, shooting, elections, sports, and sentiment) using 11.7 million tweets from users located in 100 cities and posted within the 203-day study period. Experiments demonstrate that the approach can detect events across a range of application domains.

Index Terms—Hierarchical patterns, events detection, twitter stream

1 Introduction

Social media data are a rich source of information that can be used to help make sense of current events, attitudes and opinions of populations. Twitter is one of the most popular social media platforms, with around 500 million tweets posted per day.

The main challenge of event detection using Twitter is the need to detect small changes within large volumes of data. Methods designed to identify events in real-time cannot directly compare tweets across longer time periods and instead need to find ways to compare current observations against historical trends. However, the volume of data available on Twitter is also an advantage for event detection because it can be used to create a reasonable baseline model of the language expected by time and place [1].

Tools that support this research include methods for identifying where users are located [2], and methods for detecting events as they occur [3], [4]. On Twitter, events are represented by an unexpected short-term change in the number or content of tweets. Events can be further characterised by the locations where they occur or where users post from [5], or by the topics within or across events [6].

Twitter is a popular data source for applications of event detection because it is relatively open and accessible, making it amenable to answering questions about current news events [7], monitoring and responding to natural disasters [8], and as a data source to complement polling for predicting election voting [9]. In public health, Twitter data have been used for outbreak detection [10], and for measuring spatial and temporal differences in attitudes for use in modelling health outcomes [11], [12]. However, applications in public health surveillance have typically relied on aggregating large static datasets rather than implementing methods that can be used in real time surveillance, which has limited the ability to operationalize the methods in public health practice.

1) We developed an event detection framework to process tweets as they arrive in a non-stop, continuous, and streaming fashion. Information about spatial and temporal features extracted or estimated from the metadata of Twitter posts (tweets) are used directly in the construction of a spatiotemporal lattice, which then stores other features that are hierarchically aggregated to characterise events that occur over periods of time and are common across multiple cities or countries.
2) The four main modules represent a novel combination or extension of existing methods. The preprocessing module constructs a set of low-level features from incoming tweets by estimating the locations of

- *Z. Shah is with the Division of ICT, College of Science & Engineering, Hamad Bin Khalifa University, Qatar; and the Centre for Health Informatics, Australian Institute of Health Innovation, Macquarie University, Macquarie Park, NSW 2109, Australia. E-mail: zubair1.shah@yahoo.com.*
- *A. G. Dunn is with Centre for Health Informatics, Australian Institute of Health Innovation, Macquarie University, Macquarie Park, NSW 2109, Australia. E-mail: adam.dunn@mq.edu.au.*

Manuscript received 24 July 2018; revised 29 Apr. 2019; accepted 15 Oct. 2019. Date of publication 25 Oct. 2019; date of current version 14 Mar. 2022. (Corresponding author: Zubair Shah.)
Recommended for acceptance by P. Cui.
Digital Object Identifier no. 10.1109/TBDATA.2019.2948594

the users posting tweets where possible, and adjusting their timestamps to match a local hour of the day. The feature construction module aggregates tweets to construct a set of high-level features by computing aggregate statistics of signatures of selected terms or entire tweet. The estimation module uses a regression model to estimate the expected output value distribution with respect to spatiotemporal dimensions of tweets. The event detection module then maps the features into a lattice structure, applies statistical tests to compare expected and observed value distributions, and extracts a ranked list of events from the lattice based on their severity or importance.

3) The generality of the approach is due to the use of a detection function that can take a range of different forms but still work effectively within the framework to robustly detect events at multiple spatial and temporal granularities. As a consequence, the approach can be used for applications in which users may prespecify individual or multiple terms of interest, or embed a function that transforms all tweets into a score to detect unspecified events.
4) The approach was illustrated for three different types of event detection. Experiments demonstrate how the detection functions can generalise to detect events across a range of application domains.

2 Related Work

Event detection in Twitter may draw on approaches from various fields including data mining, machine learning, information retrieval and extraction, text mining and natural language processing [13], [14], [15]. Data mining and machine learning based approaches aim to group similar tweets using classification, clustering or topic modelling algorithms, where a group or a class is considered as an event [16], [17], [18], [19], [20], [21]. Researchers have also used various classifiers to increase the precision of event detection methods. For instance, a content classifier trained on manually labeled events was used to predict event clusters [16]. Similarly, a support vector machine classifier trained on social, temporal, topical and other tweet-centric features was used to separate non-event clusters from real-world event clusters [17]. Information retrieval and extraction are often used in conjunction with other approaches such as machine learning for feature generation, selection, and event ranking [22], [23], [24].

Typical approaches comprise three phases: term monitoring, term clustering, and event ranking. The term monitoring phase relies heavily on the assumption that tweets containing a set of pre-defined event-related terms are good indicators of an event and an unusual burst in the tweets containing those terms signals the occurrence of an event [25], [26], [27], [28], [29]. The tweets identified in the term monitoring phase may correspond to many events, which are separated by clustering algorithm [21], [26], [30], [31], [32]. Strategies used to rank events have included a newsworthiness score computed using Wikipedia [25], the number of frequent terms appearing in a cluster [27], the average number of documents containing all tags of a topic, and the average burstiness of a topic [26]. Term selection is often subjective, requiring expertise from investigators to determine in advance the set of terms that are expected to correspond to an event. An advantage of the term-based approach is that it is relatively easy to implement, has less computational overhead, and has been shown to work well in a range of scenarios, including for relatively infrequent events [25], [26], [27], [28], [33].

Our focus here is on research that considers the spatial, temporal or spatiotemporal (both spatial and temporal together) signatures of tweets to detect events. The SAX algorithm [30] provides a technique for temporal analysis of tweets, where words in tweets are clustered based on the similarity of the corresponding temporal series. A subset of "interesting" strings is defined, which represents patterns of collective attention. Sliding temporal windows are used to detect co-occurring clusters of tokens with the same or similar string. Another recently proposed approach uses the temporal signals of tweets, classifies them from their content, and filters out event clusters that are about old stories [34]. Similarly, a framework consisting of a data model for description of morphological features of the populations of geo-location of social media is developed that defines a set of relationships by using differential measurements in spatial, temporal, and semantic dimensions [35]. However, this framework lacks the ability to identify differences in distributions across space and over time, which may produce results with keywords indicating completely random patterns. This problem is addressed in [36] by focusing on specific point patterns such as spatiotemporal auto-correlation of keywords that were clustered together. These techniques are shown to offer advantages over other methods in terms of performance. There is scope to expand on these approaches to consider spatial signatures of tweets, not only to help with identifying the event location but also to separate events where location matters.

Some event detection approaches are built specifically to detect spatiotemporal events such as typhoons and earthquakes. Approaches in this space consider Twitter users posting event-related tweets as sensors [33]. Here a spatiotemporal model is build that views the temporal signature of tweets as an exponential distribution curve and develops a probability density function to compute the probability of occurrence of an event. The approach relies on the assumption that Twitter users are independent; for example, when a user posts a tweet about an earthquake, the approach assures that the user's followers are not more likely to post a tweet about the same event than non-followers [37], and this may affect the validity of models. Approaches in this area may also benefit from considering the spatiotemporal granularity of events—where some important events spread more slowly or over broader regions.

Events can also be identified through space-time scan statistics (STSS) using only time and space, without using tweet context [37]. STSS views tweets in a space-time cube, where a cylindrical window of varying radius (space) and height (time) is moved over all possible space-time locations. Each cylindrical window is considered as a potential event cluster; with the number of tweets within each cylindrical window being compared to the number of expected tweets for that window. This finds windows possessing a

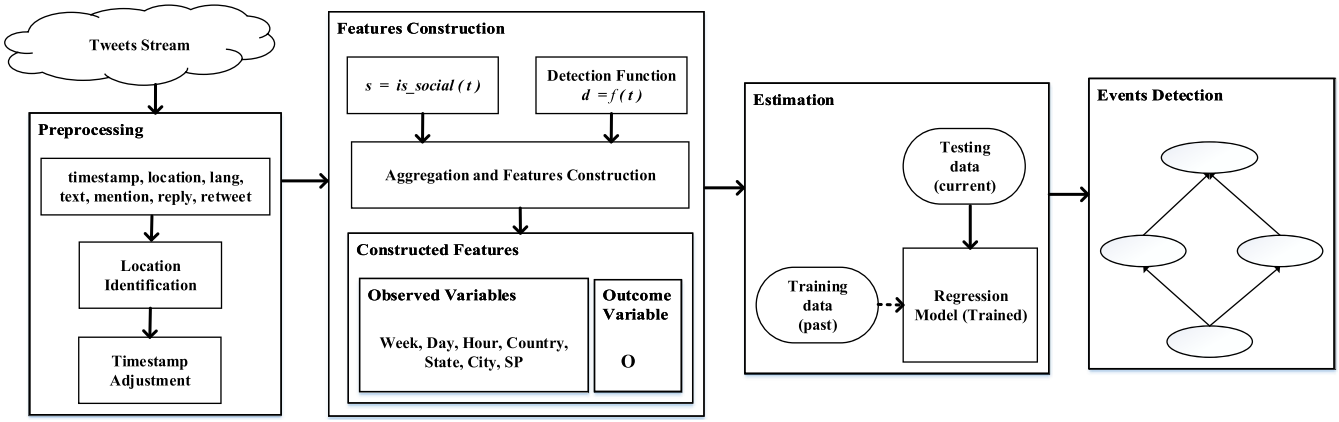

Fig. 1. An illustration of the proposed event detection method with four major modules. See Table 1 for definitions.

greater than expected number of tweets. The significance of each cluster is tested, describing the likelihood that it occurred by chance. Others used ST-DBSCAN algorithm to cluster tweets across time and space dimensions using timing and geo-tags [38]. Some researchers mapped term frequency inverse document frequency (TF-IDF) based feature words into spatiotemporal space and classified tweets into various events using machine learning algorithms [39]. A range of other approaches use geo-tagged tweets [40], [41], [42], [43], [44], [45]. Given that tweets with geo-tag information represent a biased sample of users and tweets [46], their use could detrimentally affect the ability to detect certain events. The use of high quality manual labelling of tweets is an alternative that might avoid some issues but requires resources that may limit the feasibility for events that are not known in advance [47], [48].

The approach we propose here complements existing techniques, combining text mining, machine learning and spatiotemporal analysis to design a flexible approach for detecting events in near real-time from Twitter data streams. Our approach uses a detection function design based on tweet content to assign a score to each tweet. We then build a model of what scores we expect to see in each city and each hour using a training period, test for unexpected changes in the following period, and map unexpected changes into a lattice structure to detect spatiotemporal events. The approach is implemented to work with the Twitter data stream available to the public, and we show that it is capable of automatically identifying and ranking the importance of events, even where those events are localised within a city or important across multiple countries, or where they peak and disappear within an hour or spread more slowly over a day or a week.

3 Proposed Event Detection Method

The proposed method includes four main modules (Fig. 1), preprocessing module, feature construction module, estimation module and event detection module. Tasks within these modules are performed as tweets arrive, in a non-stop, continuous and streaming fashion. Each time a new tweet arrives, preliminary features are extracted from it and are stored in a buffer or a database. At the end of each hour, features are aggregated to build the high-level features. Each hour the constructed high-level features are fed into a regression model to produce expected values based on the history or already seen examples. Next, these high-level features along with expected values are inserted into a lattice structure using a feature generalisation process. Finally, observed values are compared against the expected values at different lattice nodes to produce a ranked list of the events.

3.1 Preprocessing

The aim of preprocessing module is to extract a set of features (see Table 1) from tweets in a streaming fashion and feed them to the next module. It involves data collection, location identification or estimation, and timestamp adjustment (see Algorithm 1 in Appendix, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TBDATA.2019.2948594.).

3.1.1 Twitter Data Collection

The preprocessing module accesses Twitter via its Streaming Application Programming Interface (API), receiving tweets as JSON objects. Each tweet contains information about the user posting the tweet including username, screen

TABLE 1
The Set of Low and High-Level Features Extracted From Tweets by the Feature Construction Module

Features	Description
Hour, Day, Week	Temporal features extracted from timestamp of tweets after adjusting to local time of tweets.
City, State, Country	Spatial features extracted from the location field of the user profile using a gazetteer.
Social Proportion (SP)	The proportion of social tweets in a given hour.
d	A score computed from tweet content using the detection function f.
O	An outcome variable derived by aggregating $d \in D$ using aggregation function g.
E	An estimate of the outcome variable by regression model.
N	The number of tweets.

TABLE 2
An Illustration of Various Approaches for Computing Detection Function Using Terms Present in the Text of Tweet

	Single-Term Functions	Multi-Term Functions					Tweet-based Functions		
Tweets	w_1	w_2	w_3	w_4	w_5	w_6	w_7	$\cdots$	w_n
t_1	1	0	1	0	1	1	0	$\cdots$	1
t_2	0	0	0	1	1	0	0	$\cdots$	0
t_3	0	1	0	0	1	0	1	$\cdots$	1
t_4	1	1	1	1	0	1	1	$\cdots$	0
t_5	0	0	0	0	0	0	0	$\cdots$	0
t_6	0	0	0	0	1	0	0	$\cdots$	1
t_7	0	0	1	0	1	0	0	$\cdots$	0
t_8	1	0	0	1	0	1	0	$\cdots$	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$

$t \in T$ are tweets and $w \in W$ are set of terms in a dictionary. An entry $(i, j) = 1$ represents the presence of term w_j in tweet t_i, and an entry $(i, j) = 0$ represents the absence of term w_j in tweet t_i.

name, profile information including a free text location field, the number of users he follows, and the number of other users who follow him. Tweet metadata also include a timestamp, information about whether the tweet is a reply to another tweet, a retweet, an estimated language, and entities in the message include other users mentioned or URLs.

3.1.2 Location Identification

Identifying or estimating the home locations of users on Twitter is a challenging task. A small proportion of users include precise location information meta-data (geo-tags) with their tweets. For the vast majority of users, home locations must be estimated by either parsing user-defined text using a gazetteer or using location inference methods [2]. To identify the home locations of users, the preprocessing module takes the user-defined text from the location field in Twitter user profiles and uses Nominatim to parse it. Nominatim is a gazetteer that returns structured geographical information—such as city, state and country—and a score associated with the confidence in the answer [49]. Not all Twitter accounts represent individuals. Organisations and celebrities that include identifiable city names in their location field may introduce biases in the data due to popularity. Following the celebrity removal approach used by Rahimi et al. [50], the preprocessing module removes tweets from accounts that had more than 300,000 followers.

3.1.3 Timestamp Adjustment

The timestamps of tweets are represented by a single world clock running on Coordinated Universal Time (UTC) zone. In event detection, local time (not UTC) is a more useful way to characterise tweets, so the preprocessing module was developed to additionally convert the timestamps of tweets from UTC to local time using the location assigned to the user posting the tweet. In what follows, timestamps refer to the localised time when a tweet was posted.

3.2 Feature Construction

Feature construction module takes the raw features from preprocessing module and constructs a set of high-level features that are useful for event detection. First, the feature construction module label each tweet as either broadcast (retweets and tweets that do not mention other users), or social (replies and mentions of other users in the tweet). The social vs broadcast feature is represented by s which has two possible values 0 and 1, where a 1 means social and a 0 means broadcast. The module also computes a score for each tweet, called detection score. Let d denote a score that is computed from tweet t by function f, as:

$$d = f(t), \tag{1}$$

f is a *detection function*, which produces a score d for a tweet t. A typical detection function might be a binary function that assigns 1 to a tweet that contains one or more terms, and 0 to a tweet that does not contain those terms. While there is no restriction on the nature of the function, it must produce a score that can be compared when the scores for multiple tweets are aggregated. Detection functions can roughly be divided into three categories, single-term functions, multi-term functions and tweet-based functions (see Table 2).

1) *Single-term Functions:* Single-term functions are defined using a term where a match produces a 1 and a 0 otherwise. Some examples of single terms related to different events are flood, earthquake, shooting, suicide.
2) *Multi-term Functions:* Multi-term functions use a set of related and common terms where a match of one or more terms produces a 1 and a 0 otherwise. Some examples of multi-terms related to an event are election, elected, vote and poll.
3) *Tweet-based Functions:* Rather than being defined by a set of terms *a priori*, tweet-based functions consider the entire set of words used in a tweet and assign a detection score. Sentiment analysis methods that do not use a constrained dictionary, change in the correlation of tag pairs, and discrete wavelet signals are examples of tweet-based functions.

Based on the event type, the events detected using single and multi-term functions can be categorised into *specified* [51], whereas the events detected using tweet-based functions can be categorised into *unspecified* events [17]. The specified event detection techniques rely on the prior available information on the event of interest, while the unspecified event detection techniques find events from the bursts or trends in Twitter streams.

Both d and s are the two low-level features that are extracted from each tweet by feature construction module. The module also extracts another six low-level features from each tweet, three of them are temporal features (Hour, Day, and Week) and other three of them are spatial features (City, State, and Country). The feature construction module then takes these low-level features and aggregates them to produce high-level features (see function AGG() in lines 30-46 in Algorithm 1 in Appendix, available in the online supplemental material). This process is aggregation of the tweets along the temporal and spatial dimensions to produce three scores, N, Social Proportion (SP) and O for a set of tweets in a given hour for a given city, where N is the number of tweets, SP is the proportion of social tweets and O is defined as:

$$O = g(D), \tag{2}$$

where g is a function that takes the set of detection scores $d \in D$ and aggregates them to produce a single outcome score O. Further detail on the choice of aggregation functions is provided in Section 4.

At the end of feature construction process, the module produces a total of nine high-level features, which are Hour, Day, Week, City, State, Country, SP, N, and O. The data with these high-level features are maintained in a tabular form, where the columns are the high-level features and rows are a set of tweets aggregated per hour per city. Each row is a data instance that represents a set of tweets from a city at a certain hour. We denote such data instance by $\mathcal{I}$.

3.3 Estimation

The estimation module takes data instances from feature construction module and produces an estimated value E for each data instance $\mathcal{I}$ using a pre-trained regression model build using Elastic Net regression [52]. The dataset comprising 203 days of tweets is divided into training and testing periods. The regression model is trained on the first 80 days worth of data and is used in a streaming fashion to produce estimates for the remaining 123 days. The aim of regression model was to use it as a baseline model of expected language for detecting unexpected changes in the use of language.

The pseudocode of estimation module is given in Algorithm 2 (Appendix, available in the online supplemental material). The regression model uses only Day, Hour, City and SP as input variables and O as an output variable. The estimation produced by regression model is inserted into the corresponding data instance and is represented by E in data instance $\mathcal{I}$. Other features such as Week, State and Country are not used in the training or estimation of the model, but are used later. The SP feature is removed from the data instances after the estimation phase as it is not used in subsequent modules. The objective of using Day, Hour, City and SP in regression model was to determine whether baseline differences in spatiotemporal and social factors would introduce biases in the detection of extreme deviations in outcome variable due to any major localised event, and if accounting for them in a baseline model could address these biases.

3.4 Event Detection

The event detection module takes the data instances from estimation module and stores them in a lattice-like hierarchical structure. This module has two major tasks, building a lattice and extracting the potential events.

3.4.1 Building a Lattice

The event detection module builds a lattice structure from the high-level spatial and temporal features (Fig. 2a) of the data instances. Extending from earlier work [53], [54], the lattice is build by taking the Cartesian product of temporal (Hour, Day, Week) and spatial (City, State, Country) features and then arranging the resulting ordered pairs ($\langle Hour, City\rangle, \langle Hour, State\rangle, \langle Hour, Country\rangle \cdots \langle Week, Country\rangle$) into ancestor-descendant relationships in a hierarchical structure as shown in Fig. 2a.

Each incoming data instance $\mathcal{I}$ from estimation module is recursively inserted into the lattice nodes. The lattice has twelve nodes (Fig. 2a). First, an incoming data instance $\mathcal{I}$ is inserted into the lowest lattice node using function $\textsc{Insert}(\mathcal{I})$. Then the function recursively inserts the ancestors of the data instance $\mathcal{I}$ using $\textsc{Insert}(par(\mathcal{I}))$ until $\mathcal{I} = *$, where $par(\mathcal{I})$ is a function that computes the parents (ancestors) of the data instance $\mathcal{I}$. During the insertion of a data instance $\mathcal{I}$, if another data instance $\mathcal{I}^{'}$ in the lattice matches this data instance $\mathcal{I}$, then $\mathcal{I}$ is added to existing data instance $\mathcal{I}^{'}$ using lines 8-10 in Algorithm 3 (Appendix, available in the online supplemental material). The match between $\mathcal{I}$ and $\mathcal{I}^{'}$ is performed based on spatial and temporal features only. The addition is performed for O, E and N. The pseudocode for these steps is given in lines 1-15 of Algorithm 3.

The addition of O, E and N during insertion ensures that at each lattice node, the data instances holds aggregate values of O, E and N of their descendants, which ensures a mathematical property of the lattice. The values of O, E and N of each data instance at various lattice nodes can be computed using following equations. Let $\mathcal{I}_a$ represent a generalized data instance $\mathcal{I}$ at an ancestor node a in the spatiotemporal lattice. For the sake of convenience, let the notation $\prec$ represents descendant–ancestor relationship between data instances, for instance, $\mathcal{I} \prec \mathcal{I}_a$ would represent that $\mathcal{I}_a$ is the ancestor of $\mathcal{I}$, then the values O, E and N for a data instance $\mathcal{I}_a$ at an ancestor node a can be computed from its descendants as:

$$\mathcal{I}_a.O = \frac{\sum_{\forall\, \mathcal{I} \prec \mathcal{I}_a} \mathcal{I}.O \times \mathcal{I}.N}{\sum_{\forall\, \mathcal{I} \prec \mathcal{I}_a} \mathcal{I}.N} \tag{3}$$

$$\mathcal{I}_a.E = \frac{\sum_{\forall\, \mathcal{I} \prec \mathcal{I}_a} \mathcal{I}.E \times \mathcal{I}.N}{\sum_{\forall\, \mathcal{I} \prec \mathcal{I}_a} \mathcal{I}.N} \tag{4}$$

$$\mathcal{I}_a.N = \sum_{\forall\, \mathcal{I} \prec \mathcal{I}_a} \mathcal{I}.N, \tag{5}$$

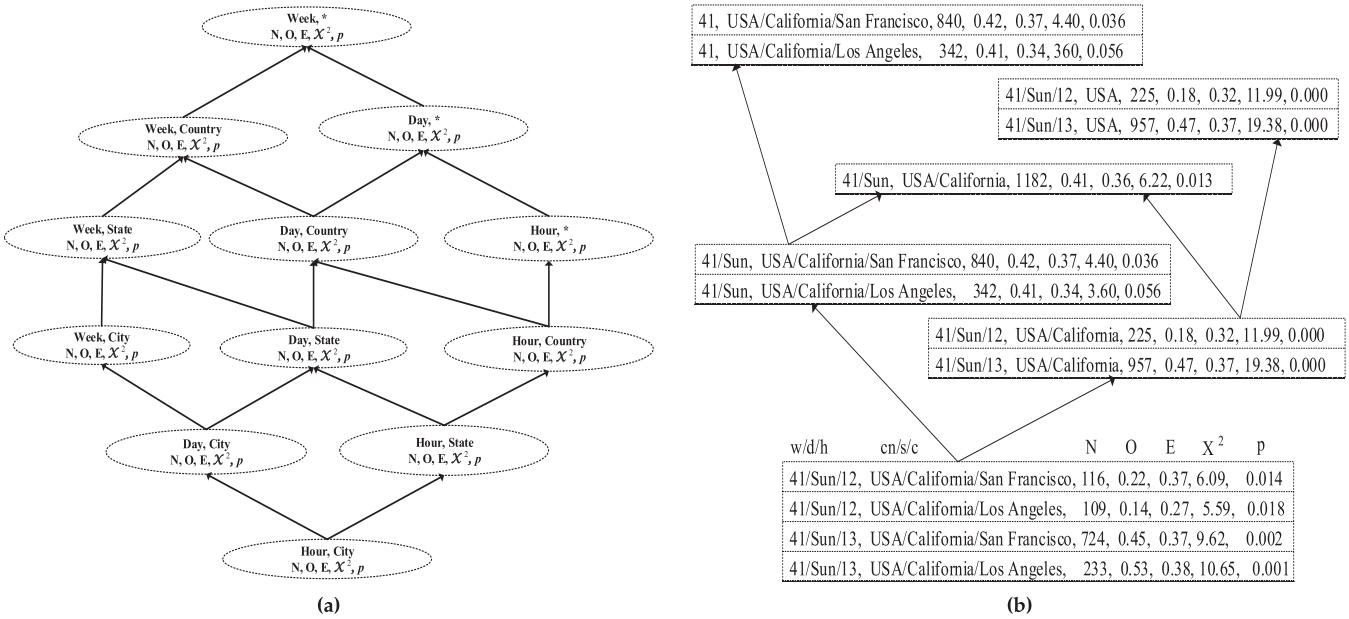

Fig. 2. **(a):** A representation of the lattice constructed using spatiotemporal features. The terms N, O, E, χ^2, and p represent the total number of tweets, the observed value, the estimated value of the outcome variable by the regression model, chi-square statistics (χ^2), and p-value (p) of chi-square test respectively. **(b)** An illustration of the data aggregation at various lattice nodes. The features are separated by a comma. At the lowest node N and O are computed from data, and E is produced by the regression model, but at all other lattice nodes N, O and E are computed using equations 3–5. The chi-square and p-value are calculated at each node using O and E.

For understanding these equations, an example of how the values O, E and N at various nodes are aggregated from their descendant nodes is provided in Fig. 2b.

The data instances at lattice nodes are gradually generalised while traversing up in the lattice. The values O, E and N are aggregated separately along either spatial, temporal or both dimensions (depending on the position of the node in the lattice). This aggregation is similar to a Group By query applied to a table in a database with grouping per hour per city or per day per country etc. The aim of such abstraction is to maintain aggregate statistics of tweets received at various spatiotemporal granularities (i.e., per hour per city or per day per country etc.), and to use them to quickly analyse a broad spatiotemporal range for detecting events in streaming fashion (e.g., in near real-time).

3.4.2 Extracting Events

The event detection queries can be issued at any time on the lattice maintained by the proposed method because the proposed method is designed to process tweets as they arrive, in the form of a continuous stream. At lattice nodes, each data instance has an observed value (from data) and an expected value (from model). The detection module compares the observed and expected values of the outcome variable using a chi-square test, and then use the resulting χ^2 value as an indicator of the magnitude of the deviation. Therefore, each time an event detection query is issued, chi-square statistics are computed to find the values of χ^2 and p for the data instances at the lattice nodes, see lines 16-44 of Algorithm 3 (Appendix, available in the online supplemental material). The event detection module takes n (the number of events) as a parameter and traverses the lattice from the bottom to the top. At each node, it scans the data instances in order, and finds the χ^2-value that is maximum in the entire lattice. We denote such a data instance by $\mathcal{I}_m$. Once $\mathcal{I}_m$ is found, the event detection algorithm removes this data instance from the lattice and considers it to be inserted into a new event lattice $\mathcal{F}$, separate from $\mathcal{L}$, which maintains the data. The detection algorithm inserts $\mathcal{I}_m$ into $\mathcal{F}$ only if there does not exist another data instance $\mathcal{I}_e$ in $\mathcal{F}$, such that $\mathcal{I}_e$ is generalizable to $\mathcal{I}_m$, i.e., $\mathcal{I}_e \prec \mathcal{I}_m$. If $\mathcal{I}_m$ satisfies the insertion criteria then after the insertion the detection algorithm also checks if any data instance $\mathcal{I}_p$ exist in $\mathcal{F}$, such that $\mathcal{I}_m$ is generalizable to $\mathcal{I}_p$, i.e., $\mathcal{I}_m \prec \mathcal{I}_p$. If any such data instance $\mathcal{I}_p$ exist in $\mathcal{F}$, then the detection algorithm removes it from $\mathcal{F}$. In short, the algorithm searches the data instances who has χ^2 maximum and whose descendant are not present in the list of existing events. Also, after the insertion, the detection module searches for data instances which are ancestors of the recently inserted data instance. If such data instances exist in the event lattice $\mathcal{F}$, the detection module removes them from the event lattice. This ensures that if an event is detected at the lower levels of the lattice, then the ancestors of the same event are not included in the output because this will be a redundant information to users.

At the end of the scanning of the entire lattice, the list of potential events is stored in $\mathcal{F}$. If the list of potential events is more than n, then the list in event lattice $\mathcal{F}$ is ranked based on χ^2-values and top n events are selected as final events.

4 Implementation and Evaluation

We implemented the proposed event detection method using Java (v 1.8). For the testing, we used an Intel Core i7 machine with a 2.10 GHz processor, a 64 GB RAM, and a 64 bit Windows 10 operating system. The data structures used in the algorithms are based on hashing techniques and require one hashing operation to lookup a particular data

TABLE 3
Event-Related Single and Multi-Terms Used in Experiment to Detect Events

Type	Event-Related Terms
S1	suicide
S2	shooting
M1	election, elected, electing, vote, poll
M2	match, game, team, player

S = single-term M = multi-term.

instance in the data structure. The search keys of data instances in the data structure are the spatial and temporal attributes of the data instance.

In our experiments, we used the Twitter Streaming API to retrieve the 1 percent sample of tweets, which is free available for research purposes. We retrieved an average of 3.6 million tweets a day for a period of 203 days from 13 July 2017 to 31 January 2018, for a total of 730.6 million tweets from 27.4 million unique users. In the dataset 30 percent (219.3 million) of tweets were tagged as English language, and 72.2 percent (158.4 million) of these tweets had location information in the user profiles. Using Nominatim as a gazetteer for resolving location text in user profiles to locations, we finally used 7.3 percent (11.7 million) of all English-language tweets. These were the tweets from users in one of the 100 cities that we selected based on the number of tweets (tweets that could be resolved to a country or other administrative area were not included). The top 100 cities comprised 52 cities in North America (45 from the United States, 6 from Canada, and 1 from Mexico), 11 cities in the United Kingdom, 6 cities in Europe, 16 cities in Asia and South-East Asia, 9 cities in Africa, 3 cities in Australasia, 2 cities in the Middle East and 1 city in South America.

We selected various event-related terms to test our proposed method. The detail of these terms is given in Table 3. We developed detection functions for these terms, which are discussed in the following subsections. We performed experiments on about six months worth of streaming data, used the detection functions in the proposed event detection method and present the results here. We ran the detection algorithm for various values of n and present the results for $n = 20$, i.e., to find the top 20 events for the selected domain.

4.1 Single and Multi-Term Detection Functions

We developed simple single and multi-term detection functions, which take a tweet t and return 1 if the selected terms are present in the tweet or return 0 otherwise. The aggregation function g is simply the computation of fraction of tweets ($O = \frac{\sum D}{N}$) that contains the selected term, aggregated along time and location dimensions.

The first single-term that we chose was suicide. Fig. 3 and Table 4 represents suicide related events detected by our method. The proposed method identified a mix of low and high-level spatiotemporal suicide related events. In some cases, the same suicide event is identified at lower as well as higher level of the spatiotemporal lattice but at different locations and for an extended period of time. For instance, the suicide of Korean pop singer Jonghyun was first identified between 6 PM to 7 PM on 18 Dec 2017 in Singapore but the same event is also detected in Seoul between 8 PM to 9 PM. The Jonghyun was pronounced dead at the hospital at around 6:32 PM. This event remains important in Seoul and Singapore for two days. Similarly, some events are identified only at the higher level of the spatiotemporal hierarchy, such as the anniversary of Canadian MP Dave Batters who died by suicide in 2009 as well as the suicide of 10 year old schoolgirl Ashawnty. This is because these types of events generate a steady tweet volume over an extended period of time (and possibly in different locations) so it becomes an event only at the higher level of the spatiotemporal hierarchy. Events like these are not significant at lower spatial or temporal granularity. These types of events may not be detected by existing event detection systems which consider a single level of spatial or temporal hierarchy. If an event detection system only considers hourly (or city) patterns, it may miss events that do not generate

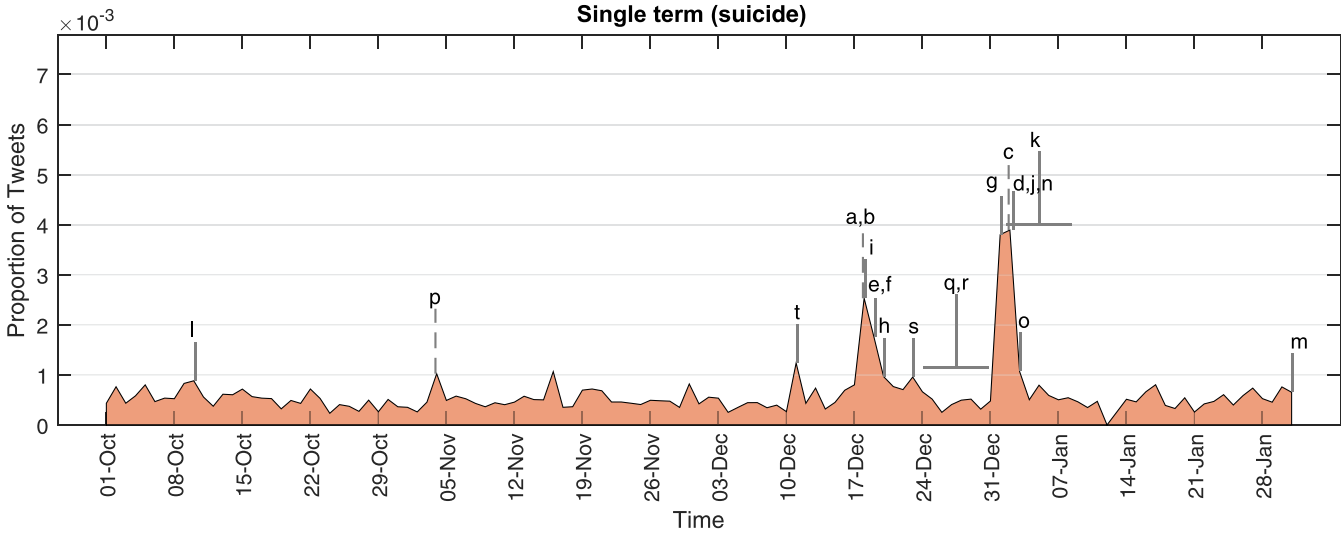

Fig. 3. Daily proportions of tweets containing term suicide (shaded area), with detected events labeled to align with Table 4. Hourly events are represented by vertical dashed lines, daily events by vertical solid lines, and weekly events by horizontal lines.

TABLE 4
List of Events Detected by the Proposed Method Using Term Suicide

Lattice Nodes	#Events	(Label) Time - Location	%O(%E)	χ^2	Frequently Mentioned Terms
[Hour, City]	3	(a) 9PM 18 Dec 17 - Seoul	5.96 (0.0534)	9.28	jonghyun, shinee, songs, addiction, depression, broken
		(b) 7PM 18 Dec 17 - Singapore	7.32 (0.0735)	6.23	jonghyun, sister, report, idols, police
		(c) 1AM 2 Jan 18 - New York City	2.68 (0.0664)	6.08	video, mocking, fans, logan, paul, narcissistic
[Day, City]	7	(d) 2 Jan 18 - London	0.60 (0.0632)	19.26	logan, paul, youtube, hanging, demonetizes, aokigahara
		(e) 19 Dec 17 - Seoul	1.69 (0.0643)	15.35	jonghyun, hospital, coal, briquettes, burning
		(f) 19 Dec 17 - Singapore	2.33 (0.0734)	14.4	jonghyun, shawol, celebrity, fansite, justice
		(g) 1 Jan 18 - Los Angeles	0.51 (0.0585)	9.97	paul, logan, mocking, hanging, apologizes, narcissistic
		(h) 20 Dec 17 - Seoul	1.17 (0.0607)	9.05	jonghyun, chester, shallow, bennington
		(i) 18 Dec 17 - Los Angeles	0.34 (0.0583)	6.24	jonghyun, health, mental, celebrity, artist, worldwide,
		(j) 2 Jan 18 - Singapore	0.86 (0.0715)	6.03	bruh, paul, logan, video, celebrity, youtuber, narcissistic
[Hour, State]	0	—	—	—	—
[Week, City]	1	(k) Week 1, Jan 18 - Toronto	0.19 (0.0578)	6.01	logan, forest, mocking, apologizes, canada, narcissistic
[Day, State]	2	(l) 10 Oct 17 - England	0.31 (0.0630)	11.59	#worldmentalhealthday, #wmhd, research, awareness
		(m) 31 Jan 18 - Ontario	0.38 (0.0643)	6.24	#bellletstalk, aboriginal, teen, #davebatters, survivors
[Hour, Country]	0	—	—	—	—
[Week, State]	0	—	—	—	—
[Day, Country]	2	(n) 2 Jan 18 - Australia	0.71 (0.0646)	7.39	paul, logan, video, forest, mocking, disrespectful
		(o) 3 Jan 18 - Australia	0.69 (0.0602)	7.39	forest, paul, logan, aokigahara, tree, hanging, japanese
[Hour, *]	1	(p) 9PM 4 Nov 17	0.32 (0.0476)	6.41	charlotte, jail, cheery, teenager, girl, aussie, lebanon
[Week, Country]	2	(q) Week 51, Dec 17 - Mexico	0.35 (0.0605)	8.02	jonghyun, pain, contemplating, anxiety, netizen
		(r) Week 51, Dec 17 - Philippines	0.47 (0.0642)	7.38	celebrity, jonghyun, #savealife, bennington, filipinos,
[Week, *]	0	—	—	—	—

%O and %E represent percentage of Observed and Expected tweets.

significant tweet volume at any specific hour (or city) but collectively become important at corresponding day level.

The second single-term used was shooting. Fig. 4 and Table 5 represents shooting related events identified by our method. Many important events-related to the Las Vegas and Texas church shooting were detected at various spatio-temporal granularities, i.e., across major locations (US cities, States and the US as whole) and at different time periods. The Las Vegas shooting event remained significant for the next two weeks, whereas the Texas church shooting disappeared within two days. The time and location along the frequent terms contain enough information to characterise the reported events. There are some less known events that are detected by our method, which we verified against various news sources using the time and location of the event and terms extracted by our method. The majority of the detected events corresponds to real-world events reported in local or national news media sources.

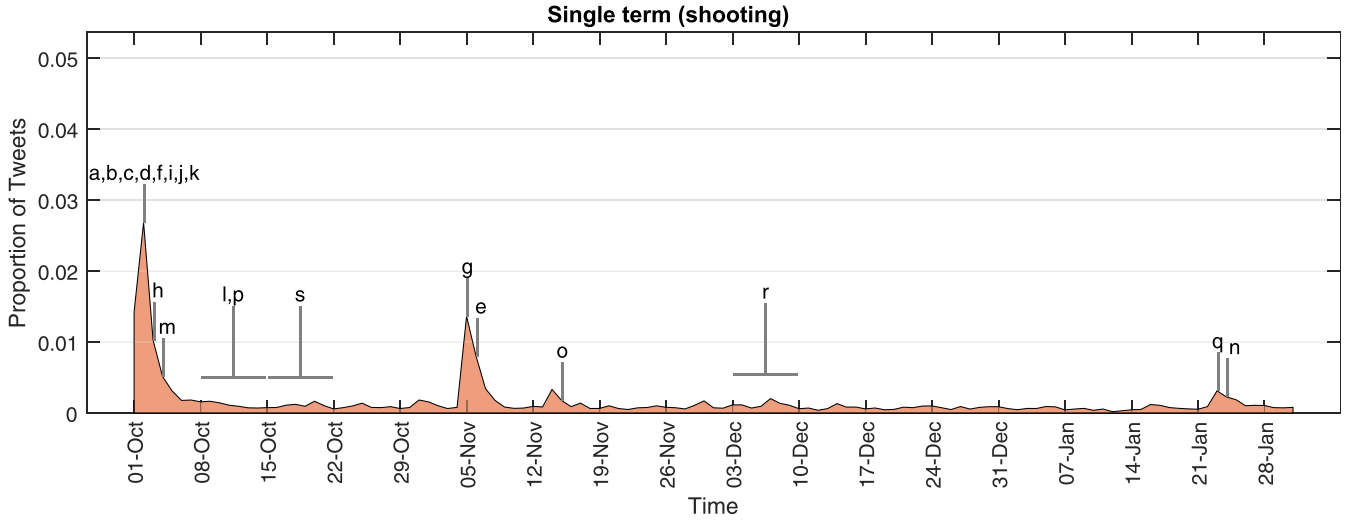

Fig. 4. Daily proportions of tweets containing term shooting (shaded area), with detected events labeled to align with Table 5. Hourly events are represented by vertical dashed lines, daily events by vertical solid lines, and weekly events by horizontal lines.

TABLE 5
List of Events Detected by the Proposed Method Using Term Shooting

Lattice Nodes	#Events	(Label) Time - Location	%O(%E)	χ^2	Frequently Mentioned Terms short
[Hour, City]	0	—	—	—	—
[Day, City]	10	(a) 2 Oct 17 - New York City	4.07 (0.0584)	266.74	mass, vegas, las, history, concert, hotel, largest,
		(b) 2 Oct 17 - Los Angeles	5.57 (0.0620)	236.83	vegas, mass, las, america, concert, #vegasshooting
		(c) 2 Oct 17 - Washington	6.13 (0.0796)	127.07	vegas, las, mass, history, police, sympathies
		(d) 2 Oct 17 - Las Vegas	8.13 (0.0832)	126.2	vegas, las, mass, strip, stephen, candles, massacre
		(e) 6 Nov 17 - New York City	1.72 (0.0572)	94.17	mass, shootings, texas, church, history, sandy, orlando
		(f) 2 Oct 17 - London	1.92 (0.0483)	88.03	mass, vegas, las, injured, deadliest, orlando
		(g) 5 Nov 17 - New York City	1.62 (0.0671)	81.35	mass, texas, church, prayers, politicians, australia
		(h) 3 Oct 17 - New York City	1.33 (0.0587)	72.24	vegas, mass, las, largest, bomber, women, kimmel
		(i) 2 Oct 17 - Atlanta	3.86 (0.0695)	59.23	vegas, mass, las,history, stephen, reports, atlanta
		(j) 2 Oct 17 - Toronto	3.50 (0.0874)	53.02	mass,vegas, las, banned, apartment, sandy, atlanta
[Hour, State]	0	—	—	—	—
[Week, City]	0	—	—	—	—
[Day, State]	1	(k) 2 Oct 17 - Texas	3.97 (0.0607)	141.96	vegas, mass, las, shootings, history, deadliest
[Hour, Country]	0	—	—	—	—
[Week, State]	1	(l) Week 40, Oct 17 - Pennsylvania	1.02 (0.0908)	69	vegas, las, mass, deadliest, paddock, australia, night
[Day, Country]	3	(m) 4 Oct 17 - USA	0.77 (0.0708)	184.67	mass, gun, #lasvegas, america, concert, police
		(n) 24 Jan 18 - USA	0.42 (0.0709)	76.01	school, kentucky, marshall, kids, children, weaponry
		(o) 15 Nov 17 - USA	0.39 (0.0708)	72.27	california, president, condolences, molester, rampage
[Hour, *]	0	—	—	—	—
[Week, Country]	1	(p) Week 40, Oct 17 - Australia	0.86 (0.0516)	74.41	mass, vegas, las, australia, police, orlando, newsaus
[Day, *]	1	(q) 23 Jan 18	0.26 (0.0626)	70.2	school, kentucky, suspect, students, fbi, helicoptered
[Week, *]	2	(r) Week 48 Dec 17	0.12 (0.0659)	58.52	mass, school, elites, shrug, prison, church, sutherland
		(s) Week 41 Oct 17	0.11 (0.0660)	49.41	vegas, las, mass, campus, lockdown, officer, tonight

%O and %E represent percentage of Observed and Expected tweets.

We also tested our method using two multi-term domains related to election and sports (see Figs. 5 and 6 and Tables 6 and 7 for election and sports related events). Our method detected various events at different lattice nodes. Most of the events are well separated from each other, but some events are merged at the higher level of the lattice, specifically for the case of common events such as sports. In this case, when multiple events are aggregated at the higher levels of the lattice (e.g., a NFL game and a hockey game on the same day), the most frequent common terms across the two or three individual events that were aggregated end up being less useful—they are common to multiple sports, like game, match, today, tonight, gone, win, first, one, score, etc.

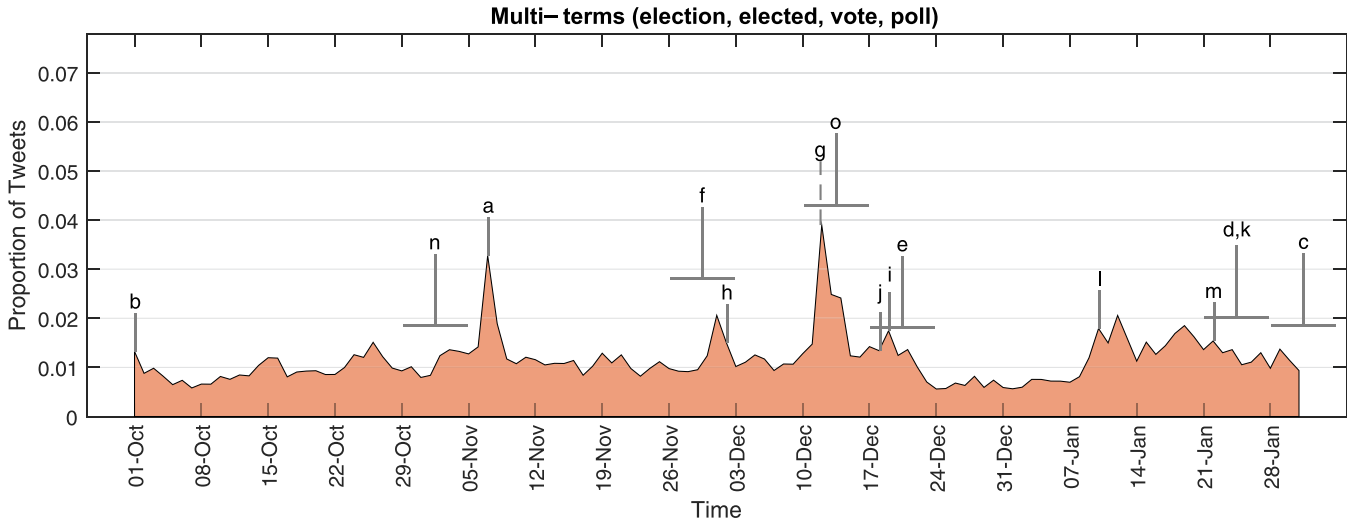

Fig. 5. Daily proportions of tweets containing election related terms (shaded area), with detected events labeled to align with Table 6. Hourly events are represented by vertical dashed lines, daily events by vertical solid lines, and weekly events by horizontal lines.

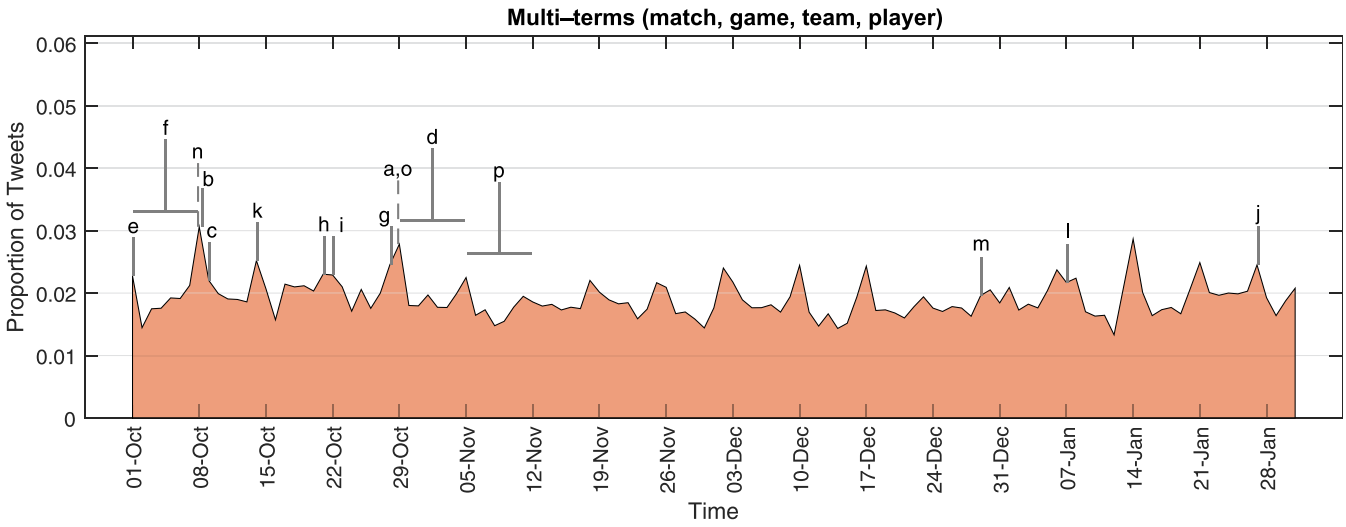

Fig. 6. Daily proportions of tweets containing sports related terms (shaded area), with detected events labeled to align with Table 7. Hourly events are represented by vertical dashed lines, daily events by vertical solid lines, and weekly events by horizontal lines.

We noticed that when the value of n is too small then the algorithm tends to combine multiple events into a single event at higher lattice nodes, whereas when the value is too large then the algorithm tends to split single events into multiple events at the lower lattice nodes. This happens only for certain events, often in the case when major events coincide (for example, on the same day in different cities), then for the small value of n the method chooses to aggregate them. At the same time, for a large value of n the method would push too many of the events down the lattice, splitting up individual events as multiple events over cities and times. There needs to be a balance—for different examples, we get different results. So for single-term functions like suicide and shooting, we are seeing individual events being split by a city as they react at different times in different locations regardless of the value of n. For sports and team games, we are seeing multiple events aggregated higher in the lattice.

Hence for the cases where it is expected that the selected terms are more frequent in tweets and multiple events can be found, a larger value is recommended. The large value will force the algorithm to output many individual events

TABLE 6
List of Events Detected by the Proposed Method using Terms Election, Elected, Electing, Vote, Poll

Lattice Nodes	#Events	(Label) Time - Location	%O(%E)	χ^2	Frequently Mentioned Terms
[Hour, City]	0	—	—	—	—
[Day, City]	2	(a) 7 Nov 17 - New York City	5.20 (0.8983)	199.57	virginia, #electionday, governor, sink, democrat, danica
		(b) 1 Oct 17 - Barcelona	17.84 (1.4163)	151.77	police, spanish, referendum, catalonia, bullets, rubber
[Hour, State]	0	—	—	—	—
[Week, City]	4	(c) Week 4, Jan 18 - Seoul	7.74 (0.9912)	394.01	#iheartawards, #btsarmy, #teambts, #bts, #bestboyband
		(d) Week 3, Jan 18 - Seoul	7.52 (1.0009)	330.32	#iheartawards, #btsarmy, #bts, #bestboyband
		(e) Week 50, Dec 17 - Mumbai	5.23 (0.7932)	240.4	#tvpersonality, #hellyshah, #biggboss, gujarat, modi
		(f) Week 47, Nov 17 - Manila	5.46 (0.7533)	175.92	#missuniverse, #philippines, rachel, peters, thailand
[Day, State]	0	—	—	—	—
[Hour, Country]	1	(g) 11PM 12 Dec 17 - USA	10.49 (0.8512)	250.84	alabama, jones, democrat, senator, moore, pedophile
[Week, State]	0	—	—	—	—
[Day, Country]	3	(h) 2 Dec 17 - USA	2.54 (0.9173)	233.57	gop, senate, republican, mccain, medicare, #goptaxscam
		(i) 19 Dec 17 - USA	1.98 (0.8496)	142.17	gop, senate, republicans, #goptaxscam, virginia, moore
		(j) 18 Dec 17 - India	5.42 (0.8103)	135.3	bjp, gujarat, congress, win, voted, modi, victory
[Hour, *]	0	—	—	—	—
[Week, Country]	1	(k) Week 3, Jan 18 - South Korea	7.52 (1.0009)	330.32	#iheartawards, #bestfanarmy,#btsarmy,#bts
[Day, *]	2	(l) 10 Jan 18	1.77 (0.8403)	200.27	#iheartawards, #bestfanarmy, #bestsolobreakout
		(m) 22 Jan 18	1.61 (0.8664)	134.79	#btsarmy, #bestboyband, republicans, president
[Week, *]	2	(n) Week 43, Oct 17	1.16 (0.8869)	149.35	kenya, #amas, parliament, catalan, gop, soul, hillary
		(o) Week 49, Dec 17	1.11 (0.8618)	126.9	moore, win, #hellyshah, roy, tax, bill, senate, gujarat

%O and %E represent percentage of Observed and Expected tweets.

TABLE 7
List of Events Detected by the Proposed Method Using Terms Match, Game, Team, Player

Lattice Nodes	#Events	(Label) Time - Location	%O(%E)	χ^2	Frequently Mentioned Terms
[Hour, City]	1	(a) 10PM 29 Oct 17 - Los Angeles	12.42 (1.5202)	27.24	#worldseries, tonight, baseball, houston, #dodgers
[Day, City]	2	(b) 8 Oct 17 - Los Angeles	3.26 (1.4242)	23.94	nfl, colts, football, leaving, stunt, charlottesville
		(c) 9 Oct 17 - New York City	2.64 (1.4837)	20.19	pence, nfl, mike, jerry, kneel, yankees, queen, cowboys
[Hour, State]	0	—	—	—	—
[Week, City]	1	(d) Week 43, Oct 17 - Philadelphia	3.49 (2.2269)	15.05	eagles, simmons, final, love, yards, philly
[Day, State]	1	(e) 1 Oct 17 - Texas	3.31 (1.6681)	16.41	nfl, kneeling, cowboys, texas, rams, thursday, garland
[Hour, Country]	0	—	—	—	—
[Week, State]	1	(f) Week 39, Oct 17 - Texas	3.31 (1.6681)	16.41	nfl, football, kneeling, cowboys, final, texas, watson
[Day, Country]	7	(g) 28 Oct 17 - USA	2.63 (1.6865)	55.64	nfl, won, #worldseries, gurriel, dodgers, nba, astros
		(h) 21 Oct 17 - USA	2.55 (1.6924)	46.23	football, tonight, lonzo, yankees, nfl, jackson
		(i) 22 Oct 17 - USA	2.57 (1.7421)	45.13	falcons, giannis, nba, atlanta, start, final
		(j) 27 Jan 18 - India	4.20 (1.6372)	40.19	#iplauction, ipl, indian, savind, shikhar, star
		(k) 14 Oct 17 - USA	2.38 (1.6944)	31.91	nfl, hertha, berlin, solidarity, german, football
		(l) 7 Jan 18 - USA	2.26 (1.7536)	20.09	nfl, season, saints, back, won, football, panthers
		(m) 29 Dec 17 - USA	2.16 (1.6937)	16.36	nfl, chris, fund, charlottesville, paychecks, bowl, football
[Hour, *]	2	(n) 6PM 8 Oct 17	3.56 (1.9105)	16.32	mike, colts, nfl, football, stadium, cowboys, kahoot
		(o) 9PM 29 Oct 17	3.49 (1.9239)	16.28	#worldseries, #indvnz, football, cricket, hockey, nfl
[Week, Country]	0	—	—	—	—
[Day, *]	0	—	—	—	—
[Week, *]	1	(p) Week 44, Nov 17	1.94 (1.8224)	14.32	football, nfl, #worldseries, league

%O and %E represent percentage of Observed and Expected tweets.

separated from each other. Similarly, for the case where it is expected that the selected terms would catch single event such as shooting, a small value of n is recommended. The small value will force the algorithm to output only some specific events.

4.2 Tweet-based Detection Functions

Unlike single and multi-term detection functions, these detection functions use the entire tweet to compute a detection score instead of selecting any specific terms a priori. We selected sentiment analysis algorithm to extract sentiment signals of the tweets and used those signals to detect events. Sentiment analysis of written texts is widely studied in natural language processing [55], [56]. We consider sentiment in a simple form—positive or negative affect—and apply SentiStrength [57], a widely-used open source Java library designed for computing sentiment of tweets. SentiStrength takes a tweet and labels it with two scores, one indicating positive sentiment (from 1 to 5, least positive to most positive), and one indicating negative sentiment (from 1 to 5, least negative to most negative). We developed a detection function f, which computes the sentiment of the tweet t and returns either P, N or Nu, where P means positive tweet, N means negative tweet and Nu means neutral tweet.

The detection function is applied first on individual tweets, and then the tweets are aggregated on hourly bases for each city to construct final features. Unlike single and multi-term functions, here we prepare two separate set of final features. In the first set of features, we use the aggregation function g that computes fraction of positive tweets ($O_p = \frac{\sum D}{N}$) and use it as an observed variable O. In the second set of features, we use the aggregation function g that computes fraction of negative tweets ($O_n = \frac{\sum D}{N}$) and use it as an observed variable O. We run our event detection algorithm for both sets of features separately and combined the events detected from both of these experiments.

Fig. 7 and Table 8 represent the events detected using tweet-based function such as sentiment analysis. We can see that some of the events that are detected by our method using sentiment function are also detected using single or multi-term functions. The advantage of using tweet-based functions such as sentiment is that the knowledge of selecting events-related terms is not required, which may be appropriate for the use cases where broad types of events are of interest.

5 Discussion

In the method we propose here, we model the expected use of language per city per hour. The model is used to compare the observed language use to quantify unexpected changes in the Twitter data stream. These unexpected changes in language use are mapped into a lattice structure where aggregate statistics of tweets—posted from cities, states and countries, and within hours, days, and weeks—are maintained. The method iteratively decides whether to aggregate or disaggregate unexpected differences in a spatiotemporal lattice. Information stored in the lattice nodes can then be queried at any time to identify and rank important events. Because of such ability, the approach can be used to take

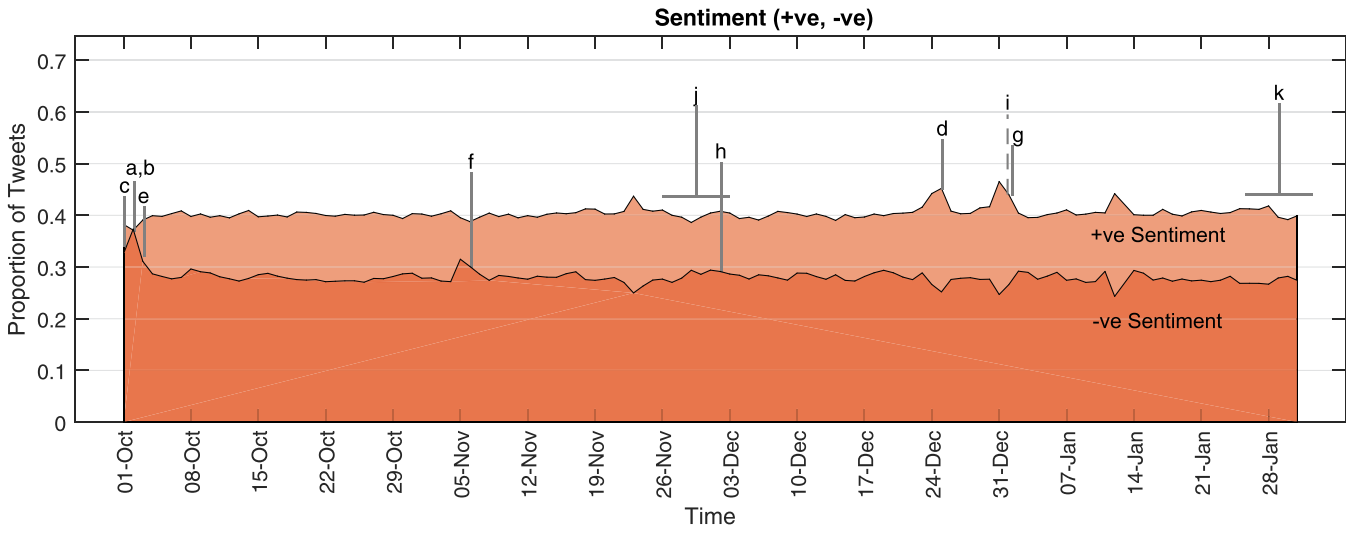

Fig. 7. Daily proportions of positive and negative tweets (shaded areas), with detected events labeled to align with Table 8. Hourly events are represented by vertical dashed lines, daily events by vertical solid lines, and weekly events by horizontal lines.

existing Twitter-based public health surveillance research applications and operationalize them to work in real-time.

We were unable to undertake a direct comparison with existing approaches by applying them to the same dataset because of differences in the input data we use. Specifically, we did not restrict our data to geo-tagged tweets, geo-tagged tweets made up a small proportion of the tweets in our dataset, and Twitter terms of service preclude the sharing of large volumes of tweets. However, we can compare approaches by considering how the features might impact their use. Our approach builds on other recent approaches that extract temporal signals from streaming Twitter data as events without considering location information [30], [34]. By integrating location information extracted from the Twitter data, our approach finds events by detecting variations in language use by time and location. Other approaches employ machine learning methods to detect events [47], [48], but the need for expert-labelling to train these models may constrain the types of events that can be detected. Our approach uses a pre-specified detection function that assigns scores to tweets based on their content but does not require manually-labelled data. Further, by mapping events into a spatiotemporal lattice and implementing a new approach for aggregating events within that lattice, our

TABLE 8
List of Events Detected by the Proposed Method Using Sentiment Analysis

Lattice Nodes	#Events	(Label) Time - Location (P or N)	%O(%E)	χ^2	Frequently Mentioned Terms
[Hour, City]	0	—	—	—	—
[Day, City]	3	(a) 2 Oct 17 - New York City (N)	43.06 (27.9347)	334.9	vegas, las, shooting, killed, america, police
		(b) 2 Oct 17 - Los Angeles (N)	46.05 (28.3402)	288.11	vegas, las, shooting, news, history, blood, puerto
		(c) 1 Oct 17 - Barcelona (N)	50.66 (24.0981)	148.36	police, spanish, violence, attack, military, rubber
[Hour, State]	0	—	—	—	—
[Week, City]	0	—	—	—	—
[Day, State]	0	—	—	—	—
[Hour, Country]	0	—	—	—	—
[Week, State]	0	—	—	—	—
[Day, Country]	5	(d) 25 Dec 17 - USA (P)	45.92 (39.2854)	212.26	christmas, merry, love, happy, family, holidays
		(e) 3 Oct 17 - USA (N)	34.17 (29.5476)	165.9	vegas, trump, shooting, victims, music, president
		(f) 6 Nov 17 - USA (N)	33.03 (28.7234)	138.8	shooting, texas, church, white, crazy
		(g) 1 Jan 18 - India (P)	53.43 (41.4051)	111.25	year, new, happy, wish, #happynewyear, celebrate
		(h) 2 Dec 17 - USA (N)	33.51 (29.6698)	103.48	tax, bill, trump, senate, republicans, #taxscambill
[Hour,*]	1	(i) 1AM 1 Jan 18 (P)	57.42 (39.2303)	174.48	year,love,#happynewyear, wishing, happiness, tonight
[Week, Country]	0	—	—	—	—
[Day,*]	0	—	—	—	—
[Week, *]	2	(j) Week 47, Nov 17 (P)	41.08 (39.8264)	126.56	like, love, thanksgiving, #missuniverse, #philippines
		(k) Week 4, Jan 18 (P)	40.96 (39.8286)	107.62	like, love, happy, #iheartawards, #bestfanarmy

%O and %E represent percentage of Observed and Expected tweets.

approach is able to capture events that affect a city or broader geographic regions, and events that occur rapidly and disappear within an hour or grow slowly and decay over multiple days.

Our spatiotemporal lattice structure proposed in this paper may also benefit from incorporating aspects of complementary approaches. For example, some approaches combine tweets that share an approximate topic and time to represent an event [13], [31], where events are characterised by event type, event phrase and a named entity but not by a location. Others proposed to use event indexing structures such as multi-layer inverted list (MIL) to support dynamic large-scale event search and update [21]. Events are maintained in the hierarchical MIL and are split using bisecting k-Means clustering if the similarity constraint is violated after new tweets are assigned to existing events in the hierarchical list. Similar to MIL, another event indexing structure is designed to support the fast matching of an incoming tweet to an existing event in the list, such as variable dimensional extendable hash (VDEH) [44]. Given an incoming tweet, VDEH uses the fraction of matched tweets in an event to calculate the similarity between the tweet and an event. It stores highly similar tweets together in one bucket of the hash to accelerate the comparison between the tweet and the events in the list. Our approach can incorporate such types of events matching indexing from these techniques by mapping them into a spatiotemporal lattice where events can be characterised by both time and location as well as by other features such as event type, event phrase and a named entity.

The approach has limitations that may be resolved through further research. In certain scenarios the approach tends to combine multiple overlapping events into a single event at higher lattice nodes, especially when the value of n is too small. Future work in the area might consider integrating clustering methods in the event detection module to separate events combined at higher lattice nodes. For example, community detection methods might be implemented to consider the structure among users posting tweets with certain scores, and topic modelling might be implemented to consider separation of language among tweets with certain scores. Also, we have not directly compared the results against existing methods. The reason is that the events produced by our proposed approach are not directly comparable to the events produced by existing methods because existing methods produce events at a single level of the spatiotemporal lattice whereas our approach produces events across multiple levels of the spatiotemporal lattice. Another limitation of the approach is the proportion of available tweets that were used to construct models of the language in each of the application domains we chose, which limits the number of cities we could include and the types of events that we could detect. Over a period of 203 days we retrieved 730.6 million tweets, of which 30 percent (219.3 million) were labeled as English-language tweets. Of these, 72.2 percent (158.4 million) of the tweets were from users with location information available. Of these, we were able to assign 7.3 percent (11.7 million) of all English-language tweets to one of 100 cities. Newer approaches to location inference may improve the approach by increasing the proportion of tweets that can be assigned to these cities [2]. This may improve the performance of the model by making it possible to extend to a broader range of cities and to rarer events.

6 Conclusion

Identifying spatiotemporal patterns from social media data streams is useful for a variety of applications, which may include disaster response, tracking the impact of news, outbreak detection and other forms of public health surveillance. We proposed an approach that models the expected rate of terms use per city per hour to produce a model to detect the magnitude of unexpected changes in the use of terms. It maps the unexpected changes in Twitter data streams into a lattice structure, where various aggregate statistics of tweets posted from cities and countries, and within hours, days, and weeks are maintained in the lattice nodes, which are used to quickly detect events at various spatiotemporal granularities in near real-time. It ranks events by importance and iteratively decides whether to aggregate or disaggregate unexpected differences in a spatiotemporal lattice. The output is a ranked list of events that are defined by a list of matching tweets posted within a constrained period of time and location. The advantage of the approach we propose here is its flexibility—it can be used for events represented by relatively infrequent terms through to events represented by unexpected changes in the use of relatively common terms. By implementing the approach with the publicly-available Twitter stream, we demonstrated that the approach is capable of ranking events that are a mix of those localised within a city or an hour to events that grow and dissipate over extended periods and geographic regions. The results indicate that the approach sometimes aggregates distinct events that occur simultaneously within a state or country. This suggests that the approach could be improved by incorporating term-based clustering of tweets prior to aggregation in the lattice to help distinguish between overlapping events.

Acknowledgments

The study was funded by the National Health & Medical Research Council Project Grant APP1128968.

References

[1] Z. Shah, P. Martin, E. Coiera, K. D. Mandl, and A. G. Dunn, "Modelling spatiotemporal variation of positive and negative sentiment on twitter to improve the identification of localised deviations," *J. Med. Internet Res.*, vol. 20, no. 4, pp. 1–16, 2019.

[2] X. Zheng, J. Han, and A. Sun, "A survey of location prediction on twitter," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1652–1671, Sep. 2018.

[3] M. Hasan, M. A. Orgun, and R. Schwitter, "A survey on real-time event detection from the twitter data stream," *J. Inf. Sci.*, vol. 44, no. 4, pp. 443–463, 2018.

[4] D. T. Nguyen and J. E. Jung, "Real-time event detection for online behavioral analysis of big social data," *Future Gener. Comput. Syst.*, vol. 66, pp. 137–145, 2017.

[5] T. Sakaki, M. Okazaki, and Y. Matsuo, "Tweet analysis for real-time event detection and earthquake reporting system development," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 4, pp. 919–931, Apr. 2013.

[6] W. Dou, X. Wang, W. Ribarsky, and M. Zhou, "Event detection in social media data," in *Proc. IEEE VisWeek Workshop Interactive Vis. Text Analytics-Task Driven Analytics Social Media Content*, 2012, pp. 971–980.

[7] A. Hermida, "Twittering the news: The emergence of ambient journalism," *Journalism Practice*, vol. 4, no. 3, pp. 297–308, 2010.
[8] A. L. Hughes and L. Palen, "Twitter adoption and use in mass convergence and emergency events," *Int. J. Emergency Manage.*, vol. 6, no. 3–4, pp. 248–260, 2009.
[9] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe, "Predicting elections with twitter: What 140 characters reveal about political sentiment," in *Proc. Int. AAAI Conf. Weblogs Social Media*, 2010, vol. 10, pp. 178–185.
[10] X. Ji, S. A. Chun, and J. Geller, "Epidemic outbreak and spread detection system based on twitter data," in *Proc. Int. Conf. Health Inf. Sci.*, 2012, pp. 152–163.
[11] A. G. Dunn, D. Surian, J. Leask, A. Dey, K. D. Mandl, and E. Coiera, "Mapping information exposure on social media to explain differences in hpv vaccine coverage in the united states," *Vaccine*, vol. 35, no. 23, pp. 3033–3040, 2017.
[12] M. Salathé and S. Khandelwal, "Assessing vaccination sentiments with online social media: implications for infectious disease dynamics and control," *PLoS Comput. Biol.*, vol. 7, no. 10, 2011, Art. no. e1002199.
[13] A. Ritter, O. Etzioni, S. Clark et al., "Open domain event extraction from twitter," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 1104–1112.
[14] F. Atefeh and W. Khreich, "A survey of techniques for event detection in twitter," *Comput. Intell.*, vol. 31, no. 1, pp. 132–164, 2015.
[15] A.-M. Popescu and M. Pennacchiotti, "Detecting controversial events from twitter," in *Proc. 19th ACM Int. Conf. Inf. Knowl. Manage.*, 2010, pp. 1873–1876.
[16] M. Osborne et al., "Real-time detection, tracking, and monitoring of automatically discovered events in social media," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics Syst. Demonstrations*, Jun. 2014, pp. 37–42. [Online]. Available: https://www.aclweb.org/anthology/P14-5007, doi: 10.3115/v1/P14-5007.
[17] H. Becker, M. Naaman, and L. Gravano, "Beyond trending topics: Real-world event identification on twitter," in *Proc. 5th Int. AAAI Conf. Weblogs Social Media*, 2011, pp. 438–441.
[18] S. Phuvipadawat and T. Murata, "Breaking news detection and tracking in twitter," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. Intelligent Agent Technol.*, 2010, pp. 120–123.
[19] M. Osborne, S. Petrovic, R. McCreadie, C. Macdonald, and I. Ounis, "Bieber no more: First story detection using twitter and wikipedia," in *Proc. SIGIR Workshop Time-Aware Inf. Access*, 2012.
[20] S. Petrović, M. Osborne, and V. Lavrenko, "Streaming first story detection with application to twitter," in *Proc. Human Language Technologies: Annu. Conf. North American Chapter Association Comput. Linguistics*, 2010, pp. 181–189.
[21] H. Cai, Z. Huang, D. Srivastava, and Q. Zhang, "Indexing evolving events from tweet streams," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 11, pp. 3001–3015, Nov. 2015.
[22] E. Meij, W. Weerkamp, and M. De Rijke, "Adding semantics to microblog posts," in *Proc. 5th ACM Int. Conf. Web Search Data Mining*, 2012, pp. 563–572.
[23] S. Guo, M.-W. Chang, and E. Kiciman, "To link or not to link? a study on end-to-end tweet entity linking," in *Proc. Conf. North American Chapter Association Comput. Linguistics: Human Language Technol.*, 2013, pp. 1020–1030.
[24] W. Magdy, H. Sajjad, T. El-Ganainy, and F. Sebastiani, "Bridging social media via distant supervision," *Social Netw. Anal. Mining*, vol. 5, no. 1, 2015, Art. no. 35.
[25] C. Li, A. Sun, and A. Datta, "Twevent: segment-based event detection from tweets," in *Proc. 21st ACM Int. Conf. Inf. Knowl. Manage.*, 2012, pp. 155–164.
[26] F. Alvanaki, M. Sebastian, K. Ramamritham, and G. Weikum, "Enblogue: Emergent topic detection in web 2.0 streams," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2011, pp. 1271–1274.
[27] R. Parikh and K. Karlapalem, "ET: Events from tweets," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 613–620.
[28] J. Weng and B.-S. Lee, "Event detection in twitter," in *Proc. 5th Int. AAAI Conf. Weblogs Social Media*, 2011, vol. 11, pp. 401–408.
[29] Y. Gao, J. Sang, T. Ren, and C. Xu, "Hashtag-centric immersive search on social media," in *Proc. 25th ACM Int. Conf. Multimedia*, 2017, pp. 1924–1932.
[30] G. Stilo and P. Velardi, "Efficient temporal mining of micro-blog texts and its application to event discovery," *Data Mining Knowl. Discovery*, vol. 30, no. 2, pp. 372–402, 2016.
[31] C.-H. Lee, T.-F. Chien, and H.-C. Yang, "An automatic topic ranking approach for event detection on microblogging messages," in *Proc. IEEE Int. Conf. Syst. Man Cybernetics*, 2011, pp. 1358–1363.
[32] M. Mathioudakis and N. Koudas, "Twittermonitor: Trend detection over the twitter stream," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2010, pp. 1155–1158.
[33] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: Real-time event detection by social sensors," in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 851–860.
[34] Q. Li, A. Nourbakhsh, S. Shah, and X. Liu, "Real-time novel event detection from social media," in *Proc. IEEE 33rd Int. Conf. Data Eng.*, 2017, pp. 1129–1139.
[35] K.-S. Kim, H. Ogawa, A. Nakamura, and I. Kojima, "Sophy: A morphological framework for structuring geo-referenced social media," in *Proc. 7th ACM SIGSPATIAL Int. Workshop Location-Based Social Netw.*, 2014, pp. 31–40.
[36] K.-S. Kim, I. Kojima, and H. Ogawa, "Discovery of local topics by using latent spatio-temporal relationships in geo-social media," *Int. J. Geographical Inf. Sci.*, vol. 30, no. 9, pp. 1899–1922, 2016.
[37] T. Cheng and T. Wicks, "Event detection using twitter: A spatio-temporal approach," *PloS One*, vol. 9, no. 6, 2014, Art. no. e97807.
[38] Y. Huang, Y. Li, and J. Shan, "Spatial-temporal event detection from geo-tagged tweets," *ISPRS Int. J. Geo-Inf.*, vol. 7, no. 4, 2018, Art. no. 150.
[39] Y. Wang, M. S. M. Pozi, G. Yasui, Y. Kawai, K. Sumiya, and T. Akiyama, "Visualization of spatio-temporal events in geo-tagged social media," in *Proc. Int. Symp. Web Wireless Geographical Inf. Syst.*, 2017, pp. 137–152.
[40] Z. Liu, Y. Huang, and J. R. Trampier, "Spatiotemporal topic association detection on tweets," in *Proc. 24th ACM SIGSPATIAL Int. Conf. Advances Geographic Inf. Syst.*, 2016, Art. no. 28.
[41] T. Hua, F. Chen, L. Zhao, C.-T. Lu, and N. Ramakrishnan, "Automatic targeted-domain spatiotemporal event detection in twitter," *GeoInformatica*, vol. 20, no. 4, pp. 765–795, 2016.
[42] T. Sugitani, M. Shirakawa, T. Hara, and S. Nishio, "Detecting local events by analyzing spatiotemporal locality of tweets," in *Proc. 27th Int. Conf. Advanced Inf. Netw. Appl. Workshops*, 2013, pp. 191–196.
[43] X. Zhou and C. Xu, "Tracing the spatial-temporal evolution of events based on social media data," *ISPRS Int. J. Geo-Inf.*, vol. 6, no. 3, 2017, Art. no. 88.
[44] X. Zhou and L. Chen, "Event detection over twitter social media streams," *Int. J. Very Large Data Bases*, vol. 23, no. 3, pp. 381–400, 2014.
[45] J. Krumm and E. Horvitz, "Eyewitness: Identifying local events via space-time signals in twitter feeds," in *Proc. 23rd Sigspatial Int. Conf. Advances Geographic Inf. Syst.*, 2015, Art. no. 20.
[46] R. Li, S. Wang, and K. C.-C. Chang, "Multiple location profiling for users and relationships from social network and content," *Proc. VLDB Endowment*, vol. 5, no. 11, pp. 1603–1614, 2012.
[47] R. Li, K. H. Lei, R. Khadiwala, and K. C.-C. Chang, "Tedas: A twitter-based event detection and analysis system," in *Proc. IEEE 28th Int. Conf. Data Eng.*, 2012, pp. 1273–1276.
[48] L. Zhao, F. Chen, C.-T. Lu, and N. Ramakrishnan, "Spatiotemporal event forecasting in social media," in *Proc. SIAM Int. Conf. Data Mining*, 2015, pp. 963–971.
[49] D. Teske, "Geocoder accuracy ranking," in *Process Design for Natural Scientists*. Berlin, Germany: Springer, 2014, pp. 161–174.
[50] A. Rahimi, T. Cohn, and T. Baldwin, "Twitter user geolocation using a unified text and network prediction model," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process.*, Jul. 2015, vol. 2, pp. 630–636. [Online]. Available: https://www.aclweb.org/anthology/P15-2104, doi: 10.3115/v1/P15-2104.
[51] K. Massoudi, M. Tsagkias, M. De Rijke, and W. Weerkamp, "Incorporating query expansion and quality indicators in searching microblog posts," in *Proc. Eur. Conf. Inf. Retrieval*, 2011, pp. 362–367.
[52] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.
[53] Z. Shah et al., "Computing Hierarchical Summary from Two-dimensional Big Data Streams," *IEEE Trans. Parallel Distrib. Syst.*, vol. PP, no. 99, p. 1, 2017, doi: 10.1109/TPDS.2017.2778734.
[54] Z. shah, A. N. Mahmood, M. G. Barlow, Z. Tari, X. Yi, and A. Zomaya, "Computing hierarchical summary from two-dimensional big data streams," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 4, pp. 803–818, Apr. 2018.

[55] F. N. Ribeiro, M. Araújo, P. Gonçalves, M. A. Gonçalves, and F. Benevenuto, "Sentibench-a benchmark comparison of state-of-the-practice sentiment analysis methods," *EPJ Data Sci.*, vol. 5, no. 1, pp. 1–29, 2016.
[56] K. Ravi and V. Ravi, "A survey on opinion mining and sentiment analysis: Tasks, approaches and applications," *Knowl.-Based Syst.*, vol. 89, pp. 14–46, 2015.
[57] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, "Sentiment strength detection in short informal text," *J. Association Inf. Sci. Technol.*, vol. 61, no. 12, pp. 2544–2558, 2010.

Zubair Shah received the BS degree in computer science from the University of Peshawar, Pakistan, the MS degree in computer system engineering from Politecnico di Milano, Italy, and the PhD degree from the University of New South Wales, Australia. He has been a lecturer from 2012-2013 at the City University of Science & Technology, Peshawar, Pakistan and a research fellow from 2017-2019 at Australian Institute of Health Innovation, Macquarie University, Australia. He is currently an assistant professor with the Division of ICT, College of Science and Engineering, Hamad Bin Khalifa University, Qatar. His research interests include health analytics, big data analytics, machine learning, and data mining and summarization. He has published his work in various IEEE Transactions and A-tier international journals and conferences.

Adam Dunn received the PhD from The University of Western Australia, in 2007, and has been working in the area of health informatics since 2008. He is currently an associate professor with the Centre for Health Informatics, Australian Institute of Health Innovation, Macquarie University, Australia. He leads research streams in clinical research informatics, where his team uses machine learning and network science to identify and mitigate biases in the design, reporting, and synthesis of clinical trials and their results; and public health informatics, where his team uses machine learning and network science to measure the spread of health misinformation in news and social media, and seeks to empower people with tools to improve health literacy.