



E-ware: a big data system for the incremental discovery of spatio-temporal events from microblogs

Imad Afyouni¹ · Aamir Khan² · Zaher Al Aghbari¹

Received: 10 March 2021 / Accepted: 6 June 2022 / Published online: 25 June 2022
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract

Event detection from social media aims at extracting specific or generic unusual happenings, such as, family reunions, earthquakes, and disease outbreaks, among others. This paper introduces a new perspective for the incremental extraction and clustering of social events from big social data streams. We present ‘*E-ware*’, a scalable and efficient big data platform that integrates data stream and geospatial processing tools for the incremental extraction and dissemination of spatio-temporal events. We introduce a pure incremental approach for event discovery, by developing unsupervised machine learning and NLP algorithms and by computing events’ lifetime and spatial spanning. Our incremental clustering technique employs temporal sliding windows, in order to update the discovered topic clusters with the upcoming social streams (i.e., tweets). The system integrates an efficient spatio-temporal index for fast retrieval and updates of evolving event clusters. We conduct experiments over Twitter datasets to measure the effectiveness and efficiency of our system. The results demonstrate that *E-ware* has a major advantage for real-time incremental detection and tracking of events, both spatially and temporally. This leads to the development of unparalleled smart city applications, such as event-enriched trip planning, epidemic disease evolution, and proactive emergency management services.

Keywords Social data mining · Event detection · NLP · Incremental processing · Spatio-temporal scope · Stream data management

1 Introduction

Over the last two decades, social media has emerged as a great support for understanding the behavior of users and communities. The interaction of users through social media, such as Twitter and Flickr, has paved the way for productive insights and discoveries. People’s interests, feedback, check-ins, and events are among the hot topics discussed on daily bases over social media sites (Chauhan et al. 2020). The content generated by users is massive and rich; therefore,

researchers, stakeholders, and authorities can build applications to extract insightful spatio-temporal information about live events of interest (EoI).

A ‘social event’ can be commonly defined as the occurrence within a specified space and time of a real-world unusual happening (Huang et al. 2018). For example, when a social attraction occurs (e.g. a new festival or a start of a shopping discount season) in the city of Dubai, a large number of tweets will be posted about such a social event. Social events usually comprise family reunions, promotions, incidents, announcements, or natural hazards, among others. The dynamic updates of such events by the live communities in social media lay the ground for developing plenty of intelligent location-based services (LBSs). These SBSs can support a variety of domain applications, such as trip planning, emergency management, transportation, navigation, city exploration, education, and crime intelligence (Savaş and Topaloğlu 2017; SAVAŞ and TOPALOĞLU 2019).

Existing studies on event detection aim at detecting specific or generic types of events, but generally focus on extracting the main topic or subtopics related to global

✉ Imad Afyouni
iafyouni@sharjah.ac.ae

Aamir Khan
aamirkhan.wc@gmail.com

Zaher Al Aghbari
zaher@sharjah.ac.ae

¹ Department of Computer Science, University of Sharjah, Sharjah, UAE

² Faculty of Computer Science, Dalhousie University, Halifax, Canada

events (Ai et al. 2018). However, extracting other features that describe the evolution and spanning of such events over space and time need further investigation. In addition, existing techniques can find events from snapshots of historical social data in offline mode, but fail to consider the incremental detection and continuous processing of such events. For instance, small-scale events (e.g., accident or traffic jam) may be alive for a few hours, whereas large-scale events (e.g., storm or election) may span for several days or weeks, and over a larger geographical extent. Furthermore, existing event detection systems do not fully support big data stream processing, which is mandatory to achieve a scalable and worldwide event extraction and visualization.

This paper presents a new perspective for the incremental event discovery in which the spatio-temporal evolution of dynamic social events is tracked. We propose ‘*E-ware*’, a fully-fledged big data system for the incremental extraction and tracking of socially-enabled spatio-temporal events. An unsupervised learning model is developed, where events are detected by identifying anomalous topics that are bursty in a given time period. An incremental spatio-temporal clustering is then performed to group similar semantically-related topics by considering tweet textual features and the temporal patterns of the sentences. This first phase results in clusters of similar topics, where the newsworthiness of an event is calculated based on the significance of co-occurring topics, and by considering the textual and temporal analysis of related co-occurring subtopics.

An incremental clustering technique is developed by employing a sliding window at the collection and preprocessing phases, which incrementally updates the discovered event clusters with the upcoming social streams (i.e., tweets). A newly detected topic is either associated with an existing topic cluster or becomes a seed for a new cluster. Later, detected topic clusters will undergo a hierarchical spatial ‘*de-clustering*’ in order to obtain the final event clusters that are tagged with spatial and temporal components. The clustered events are indexed by a spatio-temporal index structure to facilitate searching and retrieval of events over evolving space and time. The *E-ware* platform comprises a scalable architecture that lays the ground for efficient mining of big social data streams, by leveraging cutting-edge big data and stream management tools (e.g. Spark, Kafka, Apache Nifi, GeoServer, etc.). *E-ware* provides a seamless integration and visualization of clustered events on a worldwide map, thus allowing for a unique city exploration enriched with live spatio-temporal events.

This research study aims at providing a deeper understanding of the surrounding social context of mobile users, by continuously disseminating and monitoring extracted spatio-temporal events. In other words, we need to answer the following requests:

- *what is happening around right now?*
- *when and where this event has occurred?*
- *for how much time this event has been alive, and what is the current spatial spanning of that event?*

This system should leverage an incremental and continuous processing for event extraction and integration, and allow for determining the spatio-temporal evolution of such events.

Our platform lays the ground for the development of unparalleled smart city applications including smart trip planning, tracking and prediction of major events, such as epidemic disease evolution (e.g., COVID-19 as an example), and proactive emergency management services. When compared to related literature, the main contributions of this work are summarized as follows:

1. Developing an incremental clustering technique to extract social events and to monitor their spatio-temporal evolution.
2. Designing stream processing algorithms to detect updated anomalies of spatio-temporal topics based on historical data patterns.
3. Implementing a fully-fledged big data system, referred to as ‘*E-ware*’, that integrates cutting-edge data management and processing tools for the incremental event discovery.
4. Evaluating our *E-ware* platform with respect to efficiency and effectiveness of results. We assess the intrinsic properties of our algorithms for the incremental detection, scalability, and clustering accuracy, among others.

To the best of our knowledge, none of the existing approaches fully support the incremental clustering and continuous monitoring of spatio-temporal events. In addition, our platform supports the detection of unspecified usual happenings at local and global scales in a big data and scalable environment. The remainder of this paper is organized as follows. Section 2 discusses the related work on event extraction from social media in the big data era, while Sect. 3 provides an overview of the system and describes its salient components. Our data preprocessing and ingestion methodology is discussed in Sect. 4. Section 5 introduces the details behind the spatio-temporal event detection technique, and the spatio-temporal indexing scheme within the big data framework. Section 6 highlights some implementation details, and then, Sect. 7 presents the evaluation and discussion on results. Finally, concluding remarks are highlighted, showing the potential of this research.

2 Related work

Event and anomaly detection from social media can provide deeper insights about user's and community interactions on a variety of unspecified topics of interest Atefeh and Khreich (2015), Zhang et al. (2015). This section presents the main related work on event detection and data stream processing techniques.

2.1 Event of interest (EoI) mining

Social events of Interests (EoI), which will be referred to simply as events in the rest of the paper, can be observed as the representation of the real-world happenings at a given location and time. These happenings can be classified based on the *thematic* (e.g., festival or sport events), *temporal*, *spatial*, and other learning features (e.g., user profiles and social links) Afyouni et al. (2022), Wei et al. (2019), Xin and MacEachren (2020), Savaş et al. (2015). Discovering and disseminating events from diverse online social networks and with a variety of modes (e.g., text, image) have been the focus in many research studies, such as crypto-currencies Savaş (2021), crime intelligence SAVAŞ and TOPALOĞLU (2019), politics Adedoyin-Olowe et al. (2016), traffic analysis Alkouz and Al Aghbari (2020), and fashion analysis Wazarkar and Keshavamurthy (2020). Existing works on event detection aim at detecting specific Cui et al. (2017), Gao et al. (2018), Wang and Ye (2018), Alkouz and Al Aghbari (2020), Derbas et al. (2020) or generic types of events Rehman et al. (2020), but generally focus on extracting the main topic and sub-topics of such events. However, extracting other features that describe the evolution and spanning of such events over space and time need further investigation Zhou and Chen (2017). In addition, existing techniques can find events from snapshots of the social data streams ignoring the incremental and continuous development of such events Zhou et al. (2020). For instance, small-scale events (e.g., accident or traffic jam) may be alive for a few hours, whereas large-scale events (e.g., storm or election) may span for several days or weeks, and over a larger geographical extent. Furthermore, existing event detection systems do not fully support big data stream processing, which is mandatory to achieve a scalable and worldwide event extraction and visualization.

Approaches and models for event extraction from social media mainly include feature-pivot (based on temporal features of data), document-pivot (i.e., classify documents on based a given similarity measures, such as, TF-IDF or

Cosine similarity), and topic modeling (e.g., Latent Dirichlet Allocation) Ai et al. (2018), Jelodar et al. (2019). Research on event detection originates from the Topic Detection and Tracking (TDT) field of study Allan et al. (1998). Event extraction usually comprises three stages: (1) data filtering and preprocessing; (2) data representation that involves evaluating the significance of words in incoming streams or data batches; and (3) clustering phase. Different clustering methods can be applied, such as, hierarchical clustering McInnes et al. (2017), density-based Lee (2012), and Incremental clustering Li et al. (2017), Ozdakis et al. (2017). Examples of other data representation techniques and features can also be found in Massoudi et al. (2011), Unankard et al. (2015), Hasan et al. (2016). Online clustering and Naïve Bayes classifiers have also been proposed in the TwitterStand framework Sankaranarayanan et al. (2009). TwitterNews+ was proposed in Hasan et al. (2016), which implements an incremental clustering for detecting news from Twitter. Guille et al. introduced a model named MABED to detect topics based on mentioning anomaly by identifying bursty words in a given time slice Guille and Favre (2015). Clustering is then performed to group similar words based on their co-occurrence in related tweets. However, most of the extracted clusters are abstract topics without considering the spatial dimension of the topics. The authors in Liu et al. (2019) introduced a model for event detection without manual labeling in the training data, by injecting a bias in the Neural Network with an Attention Mechanism. A comprehensive survey summarizing all recent related work in this topic can be found in Afyouni et al. (2022).

Incorporating spatial and temporal features to extracted events represent another important challenge that can help in tracking and predicting the evolution of such events over space and time. The authors in Lee et al. (2018) have presented an approach for event analysis to examine their spreading and popularity based on hashtag correlation and temporal pattern mining. A spatio-temporal extension of the density-based clustering (DBSCAN) algorithm was proposed in Huang et al. (2018). This approach clusters tweets across space and time with multiple geographical scales, while using LDA for topic modeling. The authors in Rehman et al. (2017b, 2017a, 2020), have developed a technique for event detection with multiple spatial granularities, such that accidents and traffic conditions are shown at a local scale, while job alerts, elections and natural disasters are shown at higher map scales. Although their work considers spatial extent detection in a hierarchical manner, but the approach developed is not incremental, does not consider the temporal

Table 1 Comparison of event detection related papers

Reference	Purpose	Technique	Spatial	Temporal	Thematic	Semantic	Localization	Scalability	Incremental	Periodicity
Rehman et al. (2020)	Generic	Clustering and multiscale detection	✓		✓		✓	✓		Batch
Guille and Favre (2015)	Generic	Clustering using topic co-occurrence and temporal dynamics		✓	✓					Batch
Liu et al. (2019)	Generic	Event classification without manual labeling, using biased neural networks with attention			✓	✓				Batch
Ahuja et al. (2019)	Generic	Probabilistic approach to detect breaking news (major events only) and their spatio-temporal properties	✓	✓	✓		✓			Batch
Hasan et al. (2019a)	Specific: detect breaking news	Incremental clustering, random indexing based term vector model with LSH		✓				✓	✓	Near-real-time
Peng et al. (2021)	Generic	Pairwise popularity GCN and Heterogeneous DBSCAN for social event classification		✓	✓	✓		✓		Near-real-time
Belcastro et al. (2021)	Specific: discovering sub-events after a disaster	Classic classification and DBSCAN clustering	✓		✓					Batch
Dabiri and Heaslip (2019)	Specific: traffic event detection	CNN+LSTM for classification, word embeddings and geocoding	✓		✓					Batch
George et al. (2021)	Generic	Poisson model and quadtree structure for multiscale event detection	✓		✓		✓			Batch
Han et al. (2019)	Specific: monitoring major events	Power-law distribution for spatio-temporal data		✓	✓		✓			Batch
Shah and Dunn (2019)	Specific: event localization and ranking in specific domains	Hierarchical patterns, spatio-temporal lattice	✓	✓	✓		✓	✓		Near-real-time
This Paper	Generic	Spatio-temporal clustering and indexing, term-based anomaly, incremental processing using sliding windows and sentence embeddings, big data pipeline	✓	✓	✓	✓	✓	✓	✓	Near-real-time

evolution of events, and uses a supervised learning method, i.e., bag of words, to detect events of interest.

Table 1 summarizes the recent related work, and evaluates them against our paper. In this table, we consider several criteria while assessing the different research studies, such as, the purpose whether generic or specific, spatial, temporal, thematic, and semantic features. The semantic feature denotes the use of word embeddings and other techniques for content similarity. Other criteria include the event localization, scalability and big data management, the adoption of incremental processing, and finally the periodicity standing for batch or stream processing. Ahuja et al. (2019) proposed a model for spatio-temporal event detection (STED) by employing a probabilistic approach to discover events by their associated topic, occurring time, and spatial occurrence from news and Twitter data sources Ahuja et al. (2019). Although this latest work present several advantages and has a similar objective to our study, its focus was on detecting and monitoring the global events that were discussed on news, rather than a generic model to discover all types of events. Also, there was no discussion on how to maintain a continuous processing of data streams in order to update event clusters. Another recent work on spatio-temporal event detection has introduced the principle of incremental processing over temporal slices (i.e., hours, days, and weeks), and spatial resolutions (i.e., cities, regions, and countries) Shah and Dunn (2019). However, this work only considered and monitored specific event domains (e.g., elections, sports), rather than a general purpose event detection technique. A Power-law distribution model applied to spatio-temporal data was presented in Han et al. (2019). Two algorithms were introduced, where a basic version could only represent time-series data at multiple spatial resolutions, while an advanced version could apply semantic similarity over tweet content to generate more meaningful events. George et al. (2019) used quad-tree structure for hierarchical partitioning of space, and the Poisson distribution model to detect streams' density. The study presented in Belcastro et al. (2021) is specific-purpose detection technique for discovering sub-events after a disaster, which uses classic classification and DBSCAN clustering. This approach does not take the temporal and incremental features into account and cannot be generalized to all types of events. Peng et al. (2021) introduced a model using pairwise popularity GCN and Heterogeneous DBSCAN for social event classification. However, this model does not consider the spatial dimension that we believe is crucial for event-based recommendation systems.

As illustrated in Table 1, most of the recent work could not address all criteria towards achieving our goal in

building a scalable and incremental processing technique for spatio-temporal event detection. This paper presents a unique approach that takes advantage of the latest technologies in data stream management, to develop an incremental and continuous approach for spatio-temporal event discovery. Our system automatically updates the spatial and temporal scopes of extracted event clusters with upcoming data streams, which are divided based on temporal slices. The spatial distribution of events is also calculated in a hierarchical manner, in order to understand the significance of such events based on users' interaction.

2.2 Performance and scalability perspectives

Implementing large-scale event detection requires digesting large volumes of data streams, and should consider big data management and near-real-time processing techniques. Traditional data analysis algorithms and techniques do not scale in high computational complexity with large datasets in social media. The rise of parallel and distributed computing, mainly leveraged by the MapReduce paradigm Chu et al. (2007), has enabled an unprecedented use of big data mining tools and machine learning techniques in a variety of domains. For instance, Apache Hadoop and Spark, are tangible implementations of the MapReduce paradigm. Other distributed file systems that are commonly using the MapReduce paradigm are Apache Pig, Apache HDFS, and Stratosphere. Besides, data stream processing is leveraged by several open-source tools including Apache Kafka, Apache Storm, Spark Streaming, and Flink. Libraries for machine learning are Apache Mahout, SparkMLlib, and MLBase. Lastly, Apache Nutch, Apache Zeppelin, Pentaho, and SparkR used for the development of business intelligence applications and data analysis.

Data stream management allows for a continuous manipulation and processing of unbounded streams coming from real-time sources. Data stream processing is usually classified in two forms Hare et al. (2015): (i) Native stream processing, where records are processed as soon as they arrive on the fly; and (ii) Micro-batch processing. The aim in these approaches is to maintain high-throughput responses with the least delay, while processing real-time incoming data. Moreover, an efficient retrieval and updates of dynamic events require a spatio-temporal indexing scheme, especially with regard to the evolution of their spatio-temporal properties. Several spatial and spatio-temporal indexing schemes have been proposed in non-relational distributed databases Fox et al. (2013), Liu et al. (2014), Yu et al. (2015), Whitby et al. (2017). A real-time trend detection and monitoring model from social media was recently presented

in Fedoryszak et al. (2019). However, their focus was on extracting trending topics rather than individual spatio-temporal events. Fuzzy clustering with an adaptive classification of tweets using Apache Spark was also presented in Hasan et al. (2019b).

3 System overview

We present ‘E-ware’, a fully-fledged big data system that handles unstructured data from social networks, extract topic anomalies from incoming streams, and then extract spatio-temporal events of interest (EoI) in a continuous and incremental manner. E-ware determines spatial and temporal extents of detected events. The salient components in our proposed pipeline are illustrated in Fig. 1. A brief overview of each component is presented as follows.

- **Data acquisition:** This module focuses on acquiring geotagged data from Twitter. The acquired stream of tweets are unstructured data that is preprocessed in real-time. We demonstrate our platform using Twitter data stream because of the worldwide popularity of this social media, and the accessibility of Twitter API. Our crawler collects data in micro-batches of a few minutes each to facilitate anomaly detection in late stages. In the future, this approach can be extended by designing different crawlers of geotagged textual streams from other sources and wrap it into different Kafka topics, in order to enrich the event content. Currently, E-ware only collects geotagged tweets, which means many of the tweets that are not geotagged may contain valuable information
- **Data preprocessing and ingestion:** This module involves preprocessing, packaging, and ingestion of streams in micro-batches within sliding windows, thus leveraging the continuous streaming and processing of data. Successive temporal slices are generated, while taking results from historical data streams into account.
- **Incremental event detection:** Extracting social events, such as job announcements, concerts, promotions or breaking news, is performed in different steps. This module first detects topics of interest based on word anomalies in close temporal spikes (i.e., within the same sliding window). Topics will then be checked for redundancy, and will be associated to closely related subtopics. Events are later determined by associating spatial and temporal components to extracted topics. The incremental discovery of geo-social events is performed by updating event clusters with newly digested micro-batches. Incoming streams from new batches will be integrated in existing clusters if they share topics, space, and time, or otherwise will be establishing new ones.
- **Ingestion of spatio-temporal indexed events:** A spatio-temporal indexing scheme is implemented to allow for fast update and retrieval if detected events. Querying dynamically changing attributes, such as the spatial and temporal properties is challenging in this context, since events are evolving based on the newly ingested streams. E-ware integrates a scalable spatio-temporal index for managing dynamic event clusters, thus efficiently accom-

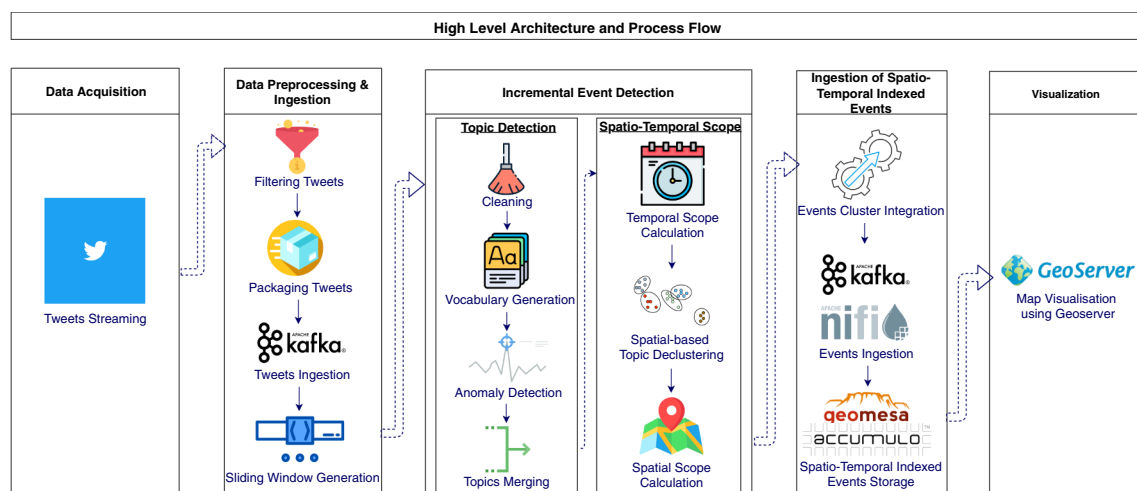


Fig. 1 System architecture

modating incoming data streams into newly extracted events, or within existing clusters.

- **Visualization:** This component creates a unique knowledge layer to digital maps by overlaying extracted events from social media in the form of dynamic attractions that span over space and time. This dynamic knowledge layer can play a vital role in enriching traditional location-based services from many perspectives.

The next sections present details on the process flow and algorithms towards building the big data platform for incremental social event detection.

4 Data preprocessing and ingestion

The data streams are collected from Twitter. Different platforms such as Instagram and Flickr can be incorporated by preparing similar Kafka topics as input, and by performing multi-source data fusion at the beginning of the pipeline or after the event detection process. The Data Preprocessing and Ingestion phase involves three major steps: Filtering, Packaging and Ingestion.

Filtering plays a key role in getting the right data. In our case, we ignore the tweets that do not match our criteria, by choosing the English language and geotagged tweets. Tweet text is split into words using tokenization, and irrelevant words are filtered out using stop words. Since these words cannot be considered as topics, stop words are used to filter out irrelevant content from tweets. The

packaging step involves extracting the required fields for event detection, especially those related to spatial and temporal attributes.

Users' posts include several non-text components, such as URLs, emojis, digits, punctuation, and user mentions. URLs are hyperlinks to other web resources such as images, videos, or websites. On the other hand, emojis are pictograms that users include in the posts to represent their emotions about the topic. However, these emojis are irrelevant to traffic events. Moreover, digits and punctuation with the posts do not contribute to the detection of traffic events. Similarly, user mentions are irrelevant to event detection. Therefore, the posts are preprocessed to clean them from these non-text components. After being cleaned, stop words are removed from the posts since they do not contribute to event detection. Then, the remaining words of a post are stemmed to remove the ambiguity of the different variations of each word. That is words are transformed to their roots.

For ingestion, we use Apache Kafka to build a fault-tolerant Big Data pipeline. The Kafka producer reads streams from twitter and publish them to Kafka topics. The data from Kafka is consumed in real-time. The tweet properties are packaged into a Kafka payload. Kafka payload is published to a Kafka topic. Each Kafka payload has a similar structure to the following JSON format:

```
{'date': '2019-10-05 16:42:55', 'latitude': 37.32034997, 'longitude': -122.00979481, 'text': "I'm at Cupertino High School in Cupertino, CA. https://t.co/vSgoVzuzgC", 'tweet_id': '118052377565952'}
```

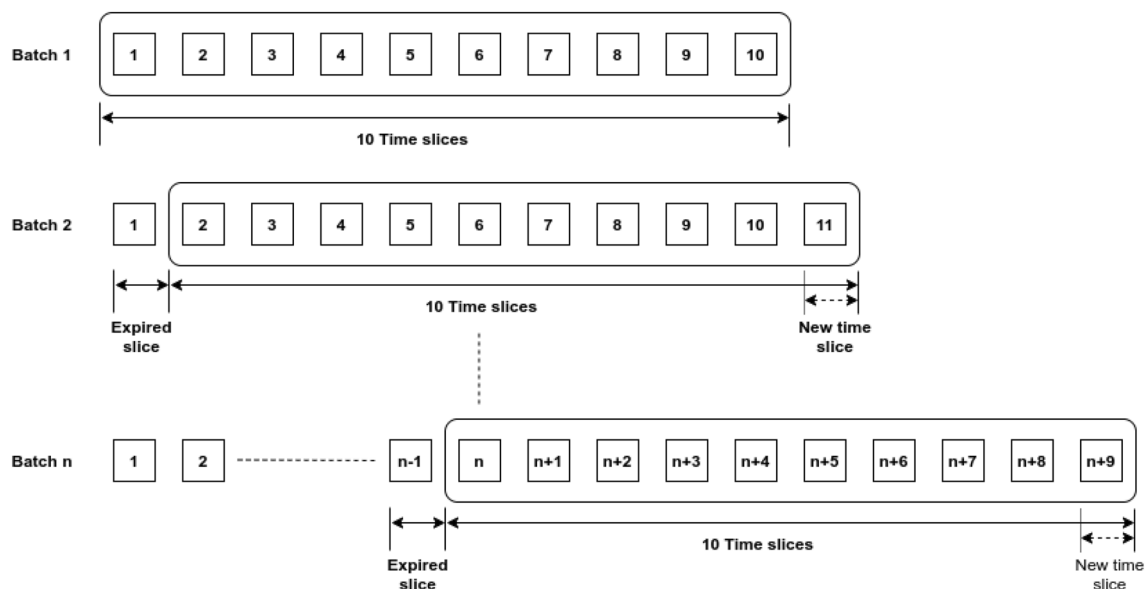


Fig. 2 Continuous processing of data streams. An example with 10 time slices (micro-batches)

This published payload is later consumed from Kafka to perform event detection. Kafka makes it possible to perform all of the above steps in near real-time. Additional components for data crawling and filtering can be added for new data sources without modifying other components, due to the fact that Kafka allows multiple topics to be consumed separately and concurrently.

The final stage after ingesting tweets into Kafka topics consists in constructing the sliding window as illustrated in Fig. 2. The continuous and incremental processing of data streams allows us to maintain the upcoming events and to merge existing clusters with new streams discussing the events. This can drastically change the performance of our event detection system from accuracy and efficiency perspectives. Our sliding windows consist of subdivided time slices, also referred to as ‘micro-batches’, each containing several minutes of data streams. This helps in maintaining some statistics on word frequency and to enable anomaly detection among micro-batches. The sliding window generation will be based on the following parameters: (1) *tsl*: Time-slice length, default to 12 (min), and 2) *nbtsb*: Number of time-slices in a batch, default to 10. The next window will be automatically adjusted by removing the *i*th slice from the batch, and appending the new time slice as the $(i + nbtsb)^{th}$ slice in the updated window. This also requires that all related files and statistics are being maintained, by updating topic anomaly scores and word frequencies, among others. The details on the management of sliding windows and updates related to new micro-batches are presented in Algorithm 1.

5 Incremental event detection

The Event Detection module consists of two major steps: (1) Incremental Topic Detection and (2) Spatio-Temporal Scope Determination.

5.1 Incremental clustering of spatio-temporal events

Topic detection is a fundamental step of event detection. The burstiness of a word allows us to classify if a word is a topic or not. Unspecified topic detection method involves extraction of unusual word occurrences within a specified temporal range. The basic topic detection presented here extends over the principles introduced in the MABED approach Guille and Favre (2015). As previously mentioned, MABED detects abstract topics only, and cannot retrieve local events accurately, since it cannot capture the whereabouts of detected events. Thus, the topic detected in MABED was a generic topic where anyone from around the world may be talking about. However, our approach extract real spatio-temporal events that occurred within a geographical extent,

and have a temporal evolution attached to them. In addition, the novel incremental processing introduced in our paper, allows for continuous updates over existing clusters, and for smooth integration of newly discovered spatio-temporal events. When running the event detection process, a few parameters are picked up to define the burstiness of words in a given temporal sliding window. The list of parameters is as follows:

- minawf*: Minimum absolute word frequency, default to 10
- maxrwf*: Maximum relative word frequency, default to 0.4
- nbw*: Number of candidate words per event, default to 10

These parameters affect the way an anomaly is detected. *minawf* determines the minimum number of times a given word should appear in the main corpora to be considered as significant. Increasing this number will reduce the number of anomalous words, and consequently the events detected. *maxrwf* denotes the max percentage of that word being used compared to the total number of words used in a certain period. The 0.4 default value means that, if a relative frequency for given word in a given sliding window is less 40%, it is considered to have normal occurrence. We have evaluated E-ware clustering results using different values of the above mentioned parameters. By comparing resulting clusters with the evaluation of the three human subjects, the above default values give the best clustering results. Therefore, the above parameters values are selected for clustering the stream of tweets in all experiments.

The incremental clustering of topics phase integrates word embeddings of tweets to extract relevant and similar topics. The detection process is illustrated in Algorithm 1.

Tweets are read periodically using a certain time interval that the user provides with the *tsl* parameter. Topic detection is then performed on each micro-batch in near real-time, by taking into account the sliding window generated incrementally, thus maintaining up-to-date records on word frequencies and tweet corpora. Different file structures are maintained for every new micro-batch appended to the sliding window as shown in Algorithm 1. They include word-frequency mappings *WF* and *LF_i*, where the key is a tokenized word and the value is its count in the current window, respectively. A word-tweets mapping *WT*, maps words to the list of related tweets. Each tweet content is consumed for *nbtsb* (Number of time-slices in a batch, 10 slices for instance) time-slices and each time-slice contains *tsl* (time-slice length: 15 min for instance) minutes. All tweets are appended to the Corpus of tweets, *CT*. The main steps in Algorithm 1 are described as follows.

Algorithm 1 Incremental Clustering of Spatio-Temporal Events

Data: CS : Corpus Size is the count of all tweets
 CT : Corpus Tweets is the list of all the tweets
 WF : Word Frequency is a list of words and their counts
 LF_i : Local Word Frequency and their counts in ts_i ,
 WT : Word Tweets is a list of words and their tweets
 $W_{ts_i, ts_i+nbt_{sb}}$: Sliding window made of nbt_{sb} time-slices
 DW : Dictionary of words, unique ids, and counts
Result: AT : Abstract topics for the window $W_{ts_i, ts_i+nbt_{sb}}$

```

begin
  set  $CS = 0$ ,  $CT = []$ ,  $AT = []$ ,  $WF = \{\}$ ,  $WT = \{\}$ ; // first time only
   $W_{ts_i, ts_i+nbt_{sb}} = \text{updateSlidingWindow}()$ 
  foreach time-slice  $ts_i \in W_{ts_i, ts_i+nbt_{sb}}$  do
    foreach tweet  $t$  in  $ts_i$  do
      preprocess( $t, LF_i, WF, WT$ )
      append( $t, CT, CS$ )
    foreach tuple  $\tau = \langle word, count \rangle \in WF$  do
      if  $count > minawf$  and  $count/CS < maxrwf$  then
         $wid = \text{assignUIId}(\tau)$ 
         $DW.append(\tau, wid)$ 
      foreach tuple  $\rho = \langle wid, word, count \rangle \in DW$  do
         $A[i] = \text{anomaly}(i, LF_i, TotalF)$ 
         $\Delta = \Sigma A[i]$ ; // magnitude of anomaly
         $DW.append(\tau, A, \Delta)$ 
      foreach tuple  $\rho = \langle wid, word, count, A, \Delta \rangle \in DW$  do
        if  $\Delta > \xi$  then
           $AT.add(\rho)$ 

```

- Data preprocessing of the current time-slice (lines 4–8). This preprocessing phase is explained in Sect. 4, where the current sliding window $W_{ts_i, ts_i+nbt_{sb}}$ is maintained with up-to-date file structures CT , WF , WT , and CS . After preprocessing, a vocabulary generation phase is performed, where we pick up the words that have the potential to become a Topic. Words and their associated counts are stored in a WF object for the whole word frequency, and LF_i for the local word frequency in the ts_i time slice.
- Anomaly detection: A topic can be considered as an unusual occurrence of a series of interconnected words that evolve over a limited temporal scope. Topic detection starts by recognizing bursty words that have high

mentioning frequency, which is considered as anomalous with respect to other individual words in the vocabulary WF . There are various ways of detecting anomalies. Measuring anomaly can be performed by estimating the expected frequency of mentions associated to a tuple $\tau = \langle word, count \rangle$ on each temporal interval ts_i based on the normal behavior, and following a probabilistic approach. Therefore, we can estimate the probability $P(\text{mention in the } i\text{th time-slice})$ of mention frequency in the $i\text{th time-slice}$, by considering the normal distribution over the overall sliding window with a large enough data portion. The accuracy of detecting anomalies depends heavily on the number of time slices considered in this computation as shown later in the experiments.

- Lines 9–13 explains the conditions for which the anomaly is considered as significant. This is based on the current mention frequency that needs to be greater than the minimal absolute expectation. The $minawf$ and $maxrwf$ are used to filter out the words that have low frequencies and high relativity respectively. The minimum Absolute Frequency ($minawf$) is the least number of times a word should occur in the tweets corpus to be considered a potential topic. The maximum Relative Frequency ($maxrwf$) is the word frequency relative to the corpus size. $Maxrwf$ can filter out recurrent words that are not related to specific topics. The selected words are stored into a unique dictionary of initial topics DW .
- The Anomaly is calculated for each word in each of tsb time-slices where tsb is inputted by the user (lines 14–18). The sum of all the anomaly scores across tsb time-slices gives a Magnitude of impact (MAG) of the topic. Anomaly is the difference of observation and expectation. Expectation is calculated as the product of tweet count in the current time-slice and the ratio of total mention frequency to the corpus size. On the other hand, observation is the mention frequency in the current time-slice. The calculated anomaly is to be appended to A .
- The initial topics that qualifies to a high magnitude of anomaly based on a given threshold will be added to the abstract topic list AT . Words that have a magnitude score less than ξ are ignored and are not considered as topics. The topics detected so far are single words that represent the basic or abstract level topics. Some of these topics may point to the same event. Merging similar topics and the computation of the temporal scope of detected topics are going to be discussed in the following section and illustrated in Algorithm 2.

5.2 Event extraction and spatio-temporal scope

In order to merge similar topics into a single cluster of related topics, the following steps are developed: (1) Candidate words selection; (2) Selection of the most relevant

accompanying words (i.e., single words from tweets that co-occur with the main topic and within the same temporal range); (3) Detecting duplicated topics; and finally, (4) Merging duplicated topics. The process continues by determining the temporal extent of evolving topics, and then by applying spatial de-clustering with spatial and textual tagging in order to discover the final geo-social events within the current sliding window.

Algorithm 2 illustrates the next stage in the event detection process. The aim in the first phase is to refine our initial topics by finding the list of accompanying topics from streams discussing the main topic (lines 3–9). Initially p candidate words are extracted from the stream texts containing the main topic. The correlation weight between topics is measured based on the coefficient introduced in Erdem et al. (2014). This coefficient estimates the gap between two different time-series, by capturing the co-variation direction between topics over time. These candidate words are a topic's co-occurring words. The candidate words are further filtered out based on the parameter k to select the top k accompanying topics by updating their frequencies. The new set of accompanying words will be associated to the main topic, and will be referred to as 'Related Topics' stored in RT . Finally, by encapsulating the related topics into an aggregated abstract topic, the list of tweets discussing these topics is updated into TT_{ids} .

Algorithm 2 Event detection and spatio-temporal scope

Data: $AT = \langle wid, word, count, A, \Delta \rangle$: Abstract Topics
RT: Related Topics; **CT:** Tweet Corpus
WF: Word Frequency is a list of words and their counts
WT: Word Tweets is a list of words and their tweets
Result: \mathcal{FE} : Final Events for the window $W_{ts_i, ts_i + nb_{tsb}}$

```

begin
  set  $RT = []$ ,  $AccW = []$ ,  $TT_{ids} = []$ ,  $GT = []$ 
  //  $TT_{ids}$ : list of tweet identifiers for a given topic
  //  $AccW$ : list of Accompanying Words
  foreach abstract topic  $at \in AT$  do
     $TT_{ids} = retrieveTweets(at, WT)$ 
     $AccW = retrieveAccWords(TT_{ids}, WT)$ 
     $\langle AccW, F \rangle = updateFrequency(AccW, WF)$ 
     $RT = selectTopKCandidates(\langle AccW, WF, k \rangle)$ 
     $TT_{ids} = updateTopicTweets(RT)$ 
     $at.append(RT, TT_{ids})$ 
     $TS = [ts_{start}, ts_{end}] = computeTemporalScope(at)$ 
     $AT.append(at, TS)$ 
  eliminateDuplicatedTopics(AT)
  MergeCorrelatedTopics(AT)
  foreach  $at \in AT$  do
    // get unique tweets from the main topic and subtopics
     $TT_{at} = retrieveGeoTweets(at, CT)$ ;
    // geotagged with topic id
     $GT.append(TT_{at})$ 
  // spatial declustering with textual and temporal indexing
   $geoCls = spatialTopicDeclustering(GT, r, minClusterSize)$ 
  updateClusters( $geoCls$ , AT) // assign topic id, topics, and temporal scope to  $geoCls$ 
   $FE = computeGeoms(geoCls)$ 
   $W_{ts_i, ts_i + nb_{tsb}} = updateSlidingWindow()$ 

```

After forming the newly aggregated topic, the next step is to compute the temporal extent or the life time of this topic (lines 10–11). Using the set of anomalies calculated over the current sliding window, the temporal scope is estimated by extracting the first time slice where the topic anomaly has appeared. Then, the topic will remain alive as long as its magnitude of impact is higher than the minimum threshold. The temporal scope is the Time-To-Live (TTL) factor of the event. Every topic and event will have a start timestamp (date and time) and an end timestamp. Since, the number of anomaly scores calculated for each topic is equal to the tsb time-slices, the temporal scope is estimated between these two time-slices. Discretization done in the early stages helps significantly in estimating the temporal scope.

The start timestamp is the date and time that corresponds to the first occurring time-slice in the tsb time-slices, which has an anomaly score greater than 0. Similarly, the ending timestamp is the date and time that corresponds to the last occurring time-slice in the tsb time-slices, which has an anomaly score greater than 0. The final range $[ts_{start}, ts_{end}]$ is the temporal scope of the Topic. The final abstract topic will now have the following properties: $at = \langle 'wid'$: identifier, ' $word$ ': main topic, ' A ': set of anomalies in the sliding window, ' Δ ': magnitude of impact, ' TS ': temporal scope, ' RT ': related topics, ' TT_{ids} ': list of topic tweet identifiers).

Before determining the spatial scope of an event, we start by cross-matching the aggregated topics generated recently in order to remove any duplicates and to merge similar topics (line 13). Cross-matching is important to avoid having the same topics as main and accompanying topics in the same time. The magnitude of impact and its temporal extent determines whether such a topic should be considered as a main topic. Later, we measure the correlation between the aggregated topics as evolving time-series to determine highly correlated topics, so that they can be merged (line 14). In this case, the main words become the aggregation of the merged topics, and the words characterizing the aggregated topic are the k words that have the highest correlation weight among all other words.

5.3 Spatial de-clustering of aggregated topics

The spatial extent depicts the whereabouts of the extracted event. The spatial information extraction step comes after the topics are detected and merged. Since all the tweets are geo-tagged, their point coordinates help us in estimating the event location. We consider topic-related tweets as the initial input for the final geo-event discovery. These event tweets have the same temporal range attached to the topic. Each topic has related tweets, which may have different coordinates. For example, a topic 'job' may have many different tweets and all of these tweets talk about a job announcement. However, they may talk about different jobs at different

locations. Here, each different job with different location should be considered as a separate event. Spatial Information extraction first removes all duplicated tweets as same tweets may belong to two or more topics. The process of spatial scope determination is explained in more details as follows.

Retrieve unique geo-tweets (Algorithm 2: lines 16–17): For each aggregated topic we had stored the relevant tweets attached to it in TT_{ids} . To avoid data fragmentation we remove the duplicated tweets. During the duplication removal process, we remove the tweets that belong to smaller topic clusters (e.g., if a tweet t belongs to two different topic clusters like ‘job, opening, hiring’ and ‘join’). In this case, we remove the tweet from the topic ‘join’ cluster. Similarly, during the duplication removal process, all the duplicated tweets are automatically removed.

Secondly, we integrate a spatial clustering technique applied to the coordinates of all tweets to cluster those related to the same topic in a close proximity (lines 19–21). We have used HDBSCAN clustering algorithm since it supports the haversine distance that computes distances on a sphere between geo-locations. Also, with HDBSCAN, there is no need to indicate the number of resulting clusters in advance, as it processes data points hierarchically. Two main parameters are used to characterize how HDBSCAN should perform the hierarchical clustering: r for radius, which specifies the spatial range in meters and helps in merging clusters within a specified distance, and the $min - cluster - size$ that specifies the minimum size allowed for a group of data points to form a cluster. These parameters are manipulated and tested in order to assess the performance of our event detection approach as a whole, and in particular, the clustering accuracy of final events.

We refer to this process as *de-clustering* because it is classifying tweets that belong to one topic based on the spatial dimension. Since we have only considered the geo-tagged tweets (latitude and longitude), each tweet will have a spatial attribute associated with it. Every topic has related tweet(s). If related tweets count is more than 1 then spatial clustering can help with the identification of events from these topics. For eg: The topic cluster ‘view, opening, hiring, read, latest, job, sales’ has some of the following related tweets:

- Text: Can you recommend anyone for this #Labor job in Owatonna, MN? Click the link in our bio to see it and more. Loader/... <https://t.co/...> Latitude: 44.0888533, Longitude: -93.2181477
- Text: We’re hiring in #Fargo, ND! Click the link in our bio to apply to this job and more: baker at Caribou Coffee #Hospitality. Latitude: 46.8578484, Longitude: -96.8614608
- Text: Nervous to apply for a job like “SALES ASSOCIATE in SAN ANTONIO, TX” at Dollar General

Corporation? Apply even if y... <https://t.co/...> Latitude: 29.3857395, Longitude: -98.4690975

Clearly, the above 3 related tweets are talking about different events even though they belong to the same context. Rather than considering a random tweet as an event, we can consider all the tweets as different events.

To form an event, we group the tweets by *topic* and *spatial cluster id*. Each of these groups is assigned an id called the *event id*. In the above example: three different tweets at three different locations that belong to the topic cluster ‘view, opening, hiring, read, latest, job, sales’, will be formed as three different events with different event ids.

As a result of the different phases for incremental event extraction, the following properties will represent our final events:

event – id: indexed unique identifier representing an event

topics: keywords that represent an event

tweet – ids: identifiers of tweets

tweet – texts: texts of tweets

importance – score: magnitude of the event

start – date: start date and time of the temporal scope of event.

end – date: end date and time of the temporal scope of event.

Duration: the duration or lifetime of an event.

geom (longitude, latitude): spatially indexed point coordinates

From the above process, we get spatio-temporal clusters that are tagged by topics. An event is formed for each of these clusters, and is assigned an event id. The final event clusters are determined with a set of related topics, temporal evolution and spatial clustering. These extracted events are published to another topic in Kafka in a continual manner for the pipeline to work smoothly.

5.4 Spatio-temporal event indexing

Once the events are detected, we need to distinguish existing clusters from the newly created ones. If the detected event already exists in database, we need to update its spatio-temporal properties; otherwise, we create and spatio-temporally index the new event. New events are published as Kafka topics and existing ones are updated in database.

Developing an efficient pipeline requires integrating a spatio-temporal index to store and query dynamic data points that represent our extracted events. The geometry-type attribute is used for the geohash-based spatial index, while the date-time attribute will complement the third dimension to index latitude, longitude, and timestamp for event point data. We take advantage of Geomesa Hughes et al. (2015), an open source platform that leverages distributed storage and scalable processing of spatio-temporal data.

Apache Accumulo database is used underneath to implement a robust big data pipeline.

The ingestion of spatio-temporal indexed events involves two major steps: (i) Events Cluster Integration, (ii) and Events Ingestion into a Spatio-Temporal Database. Once we get the active list of events, we can update the event properties in the Geomesa database with the new detected properties. After the event cluster integration, we spatio-temporally index the newly detected events using the geohash indexing scheme into Geomesa-Accumulo database.

Events cluster integration:

Event cluster integration is the process of updating the event information of an existing cluster in database. This process occurs when events are detected again in the next sliding windows. This is done to ensure that events do not duplicate in database. Each event has properties like spatial scope, temporal scope, duration, tweet identifiers and magnitude of impact, which may be updated based on the latest event detection results. For example, the temporal scope of an event has a *start* and *end-date* parameters, where *end-date* should be updated if an existing event was found again. This will also affect the duration of the event. The magnitude could also be increased because if event was detected again, this means that more people were talking about this event. In addition to that, event's spatial scope can also be adjusted if required and new related tweet-texts could be added in the existing list.

After all events are detected in the current sliding window, they are queried in the active list of events in Geomesa. The active list of events comprise those events that are still alive. Once we get the active list of events then we can update the event properties in the Geomesa database with the new detected properties. The events are deactivated based on the *TTL* parameter and a threshold *th* representing the number of time slices. If the event does not appear in the next *th* time-slices, then the event will be deactivated.

Event spatio-temporal ingestion:

After the events cluster integration, we add the newly detected events to Geomesa-Accumulo database. The Apache Kafka and Apache NiFi tools are used for the ingestion pipeline. Since we already use a 'topic' to store streaming tweets in Apache Kafka, we built other 'event' topics to store newly detected events. The NiFi process is built to automate the dataflow between the components. The term 'dataflow' here depicts the automated and managed flow of information from Kafka into Geomesa-Accumulo database. The Geomesa-Accumulo database schema is defined before any data is transferred from Kafka to Geomesa through NiFi.

6 Implementation details

In this section, we discuss the implementation of E-ware as a big data Processing system, and the event visualization module.

6.1 E-ware big data processing

The proposed system was setup on Ubuntu 18.04 operating system in a standalone mode. The device used was a MSI GS40-6QE laptop with the following key specifications: Intel Core i7-6700HQ processor, 16GB DDR4-2133 RAM, and 1TB Samsung 840 Evo Solid-State Drive. We used the following versions of the tools to setup the proposed system: Java version 8, Python version 3.6, Zookeeper version 3.4.10, Apache Kafka version 2.11-2.1.1, Apache NiFi version 1.8.0, Hadoop version 2.8.4, Apache Accumulo version 1.9.2, Geomesa-Accumulo version 2.3.1, and Geoserver version 2.14.4. It is worth mentioning that all the required tools are open source.

This dataset consists of tweets with geolocations collected using the Twitter Streaming API. We use Twitter's streaming API to collect data in real-time. Tweets are streamed from Twitter Streaming API and are fed into Apache Kafka. The data from Kafka is consumed in real-time. Python was used to write the Kafka Consumer. Python's tweepy library was used to stream the data from Twitter.

For NiFi to upload the events data to Geomesa-Accumulo store, we connected two processors, Kafka and Geomesa-Accumulo in the NiFi's user interface. We defined the required specifications for NiFi to understand the data from Kafka, and required converters to convert the data from Kafka for Geomesa to understand it. This was the ingestion pipeline.

6.2 Event visualization

Event-enriched map visualization is performed to illustrate our resulting dynamic knowledge layer. E-ware integrates Geoserver for web-based map browsing, in order to query and visualize extracted events. Figures 3 and 4 show samples of extracted events visualized based on extracted clusters with unspecified events illustrated on the left side pane. Each cluster has a mention of the number of tweets it encapsulates. Figure 5a also illustrates the details of each cluster as the set tweets included, while Fig. 5b presents the detailed properties for each event. Finally, topic-based visualizations can be demonstrated in E-ware, where all events are categorized within major unspecified topics as illustrated in Fig. 6. In this figure,

multiple events from different locations, and at different period can share the same topic, so they are grouped together only based on textual features.

7 Evaluation

We present the evaluation of E-ware and the set of experiments conducted with respect to efficiency and effectiveness of our proposed system. We analyzed the different phases from preprocessing, to event detection, up to visualization of events on map. For evaluation purposes, we collected 3,122,304 tweets during three weeks in December, 2020.

7.1 Event clustering accuracy

To calculate the accuracy of our approach, we selected a subset of the dataset (130,096 tweets) to facilitate verification of clustering results. All the considered tweets for these experiments are geo-tagged and collected from NYC, USA. Additionally, we only considered English language tweets for our experiments.

For the accuracy measures, we randomly took some event clusters and manually annotated the ground truth data. For Efficiency, we allowed our big data system collect and process tweets in real-time.

We first compared our approach against the most relevant related work that consider generating spatio-temporal event clusters out of Twitter data. We also implemented two other approaches that have some similarities, by testing their codes and reproducing the results on the same datasets: the snow

challenge winner Ifrim et al. (2014), which detects events with spatial coordinates using aggressive filtering and hierarchical clustering; and MABED Guille and Favre (2015) where topic detection has been the major focus based on anomaly detection. Our objective is also to evaluate our platform with the closest related works as presented in Table 1 (Ahuja et al. 2019; Rehman et al. 2020; Belcastro et al. 2021; Dabiri and Heaslip 2019; George et al. 2021; Shah and Dunn 2019). It should be noted that other approaches that do not consider the spatial feature in event detection are discarded from this evaluation process. Also, the STED approach proposed in Ahuja et al. (2019), does not present any performance evaluation with respect to the accuracy of their event detection technique. Moreover, the approaches presented in Belcastro et al. (2021), Dabiri and Heaslip (2019), Shah and Dunn (2019) are monitoring specific types of event detection, which cannot be fairly compared against the generic approaches that consider unsupervised learning techniques. Furthermore, Rehman et al. (2020) and George et al. (2021) consider a pure spatial approach with no temporal dimension included. As a result, Table 2 illustrates a summary of performance evaluation with respect to these related work, but it is worth considering these differences, as we could not find any related work that covers all the features presented in this paper. Table 2 the precision, clustering accuracy and F1 score for each approach. In our case, we considered some default parameters for the spatial clustering, with five as the minimum cluster size, and no radius limitation.

The results show that the aggressive filtering approach of SnowChallenge presents a very high precision, but fails to score well in recall and F1 measures due to the huge loss

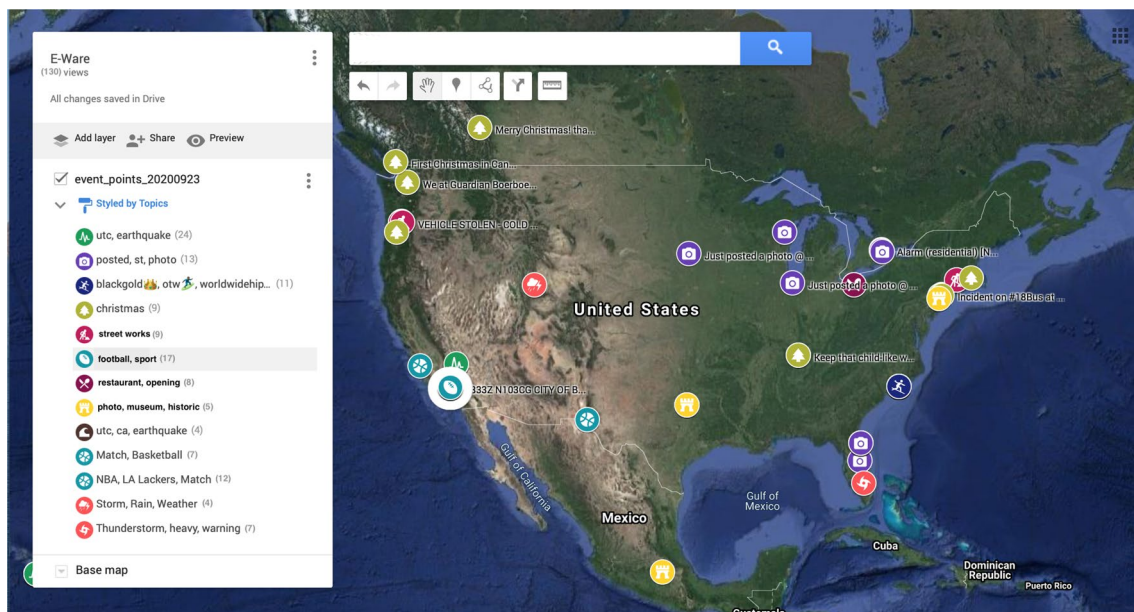
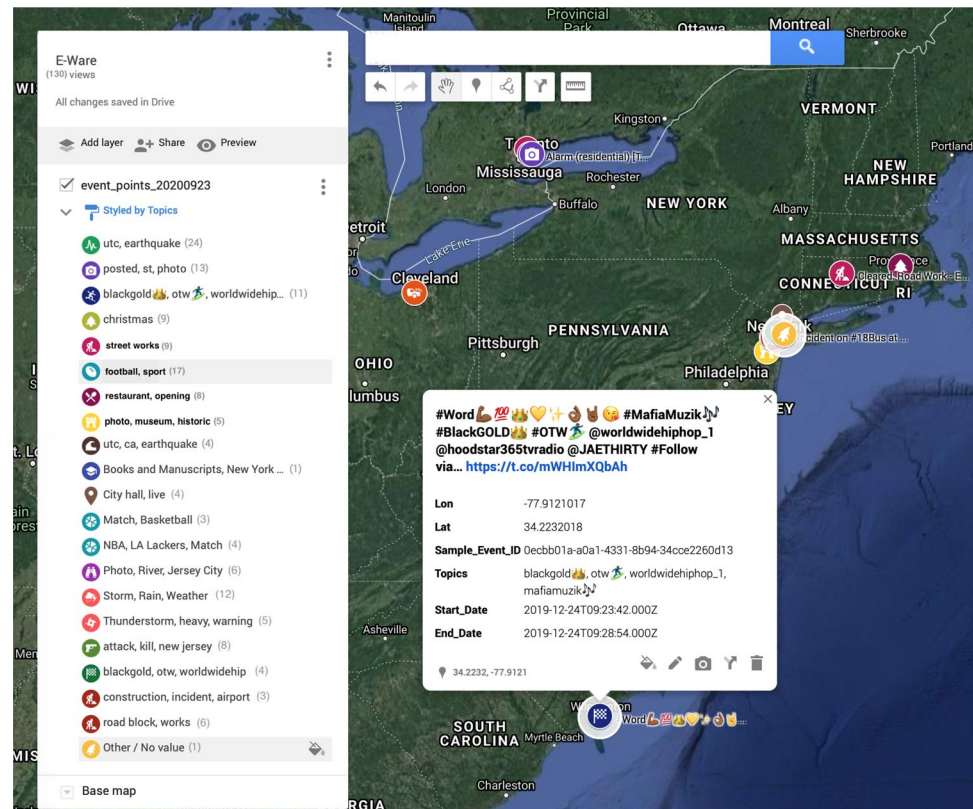


Fig. 3 Map-based event visualization with event clusters showing on the left side pane

Fig. 4 Example visualization of extracted event details



of false negatives, which is caused by aggressive filtering. MABED, on the other hand, scores average on both precision and recall, due to the fact that the approach aims at detecting high level topics rather than spatio-temporal events. Therefore many real life events can be merged within one topic, thus reducing its accuracy. our approach in E-ware scores well in the different measures, and can compete with the best approaches even though there exists no benchmark where we compare all approaches on the same basis. The clustering accuracy measure reflects the quality our extracted clusters in terms of how many tweets in the cluster are actually talking about the same event. The results show that our clustering technique works very well and the number of outliers within clusters is very low.

7.2 Effect of events' spatial clustering radius

To measure the effect of modifying the radius of event clusters on the performance of event detection, we computed the Precision, Recall and F1 of the resulting event clusters. For this experiment, we fixed the minimum cluster size to 5 and the number of time slices to 10. These performance measure were computed at the end of the tenth time slice on the manually annotated dataset of 130,096 tweets. The manual annotations of clusters was performed independently by three human subjects.

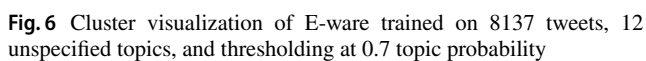
Figure 7 shows the computed Precision, Recall and F1 at three different radii, 0.5 km, 1 km and 5 km. Note that Precision and F1 performed best at 1 km. This is expected since as the radius becomes too large, cluster grow large, which in turn increases the possibility of more irrelevant events being included in the cluster. Furthermore, when the radius becomes too small the clusters are caused to be fragmented. In these fragments, the false positives become more dominant, which affect the homogeneity of the clusters and thus lowers the Precision.

On the other hand, Recall decreases as the radius becomes larger, i.e. clusters become larger. When clusters become larger, small clusters may be encapsulated in big ones, and this may result in an increased number of false negatives. Therefore, the clustering recall is negatively affected. Nonetheless, as clusters become larger they tend to include more relevant events, which improves the precision measure as shown in Fig. 7.

In Fig. 8 we show the effect of modifying the radius of events' spatial clusters on the Accuracy of clustering and tweet classification. As the radius increases, the clusters become larger and thus they become more likely to contain irrelevant events (false positives). Therefore, the accuracy of clustering drops as shown in Fig. 8a. Similarly, the accuracy of tweet classification describes how many raw tweets were actually correctly clustered within the event clusters. The degradation of performance shown in Fig. 8b is due



(b) Event detailed information

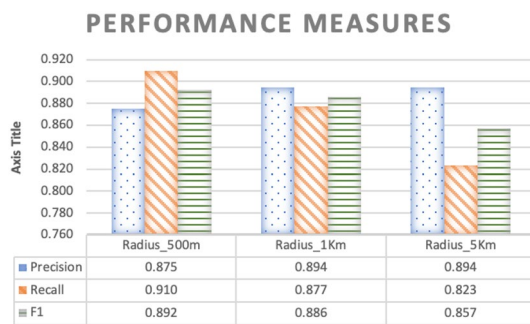


The effect of events’ spatial radius on the purity of clusters, which can be explained by the percentages of True Positives (TP) and False Positive (FP) in a cluster. Figure 9 depicts the average TP and FP of the resulting clusters at different radii (0.5km, 1 km and 5 km). The result shows that at 1 km radius, the clusters have the least percentage of FP and highest TP, i.e. purest clusters. This result is consistent with that of Fig. 7, which shows that at 1 km, the clusters have the highest Precision and F1 values.

 Springer

Table 2 Summary on performance evaluation

Reference	Qualitative (Effectiveness)			Features (coverage)	Clustering Accuracy
	Precision	Recall	F-Score		
Ozdikis et al. (2017)	0.81	0.59	0.66	Incremental, temporal, but not spatial	
Martin and Goker (2014)	0.98	0.34	0.50	Baseline, spatial only	NA
Guille and Favre (2015)	0.66	0.77	0.71	Thematic and temporal, but not spatial	NA
Rehman et al. (2020)	0.82	0.80	0.81	Spatial only	0.91
Han et al. (2019)	0.82	0.78	NA	NA	
George et al. (2021)	0.89	0.667	NA	Spatial only	
Liu et al. (2020)	0.6891	0.7833	0.7332	Thematic and temporal only	NA
Ahuja et al. (2019)	NA	NA	NA	Spatio-temporal but on major events only, not incremental	NA
This paper (E-ware)	0.88	0.85	0.86	Spatio-temporal with generic event detection and incremental processing	0.95

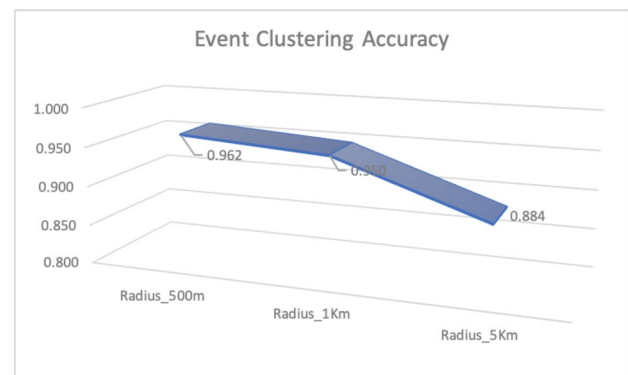
**Fig. 7** Effect of varying the event clustering radius on the performance measures: precision, recall and F1

extend E-ware in the future to report the extent, or radius, of the event (cluster), especially if the event is relatively spanning a wide area.

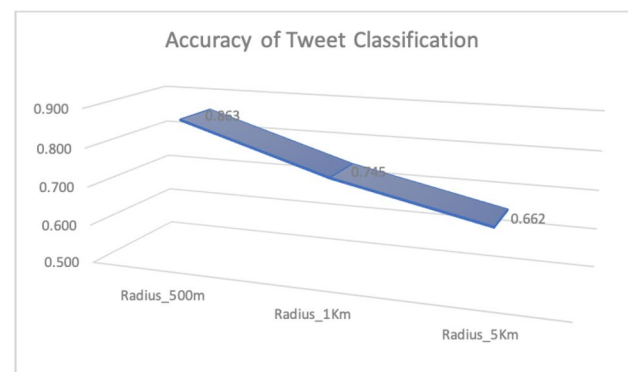
7.3 Effect of events' clustering membership

To study the effect of varying the cluster minimum size, i.e. minimum number of events per cluster, on event detection, we computed the following performance measures, Precision, Recall and F1. In this experiment, we deactivated the radius limit and fixed the number of time slices to 10. Figure 10 shows that as the minimum cluster size increases, the precision of clustering increases. This is due to the fact that setting small minimum cluster size will allow clusters to be fragmented with relevant events being placed in different clusters. However, as the minimum cluster increases, more relevant events can be contained in each cluster, That in turn increases the precision of clustering. The recall is impacted negatively by having bigger clusters in a similar way to what was described in Sect. 7.2.

Figure 11 shows the effect of modifying the minimum cluster size on the accuracy of clustering. With small minimum cluster sizes, the clusters small and its events are



(a) Clustering Accuracy



(b) Tweet Classification

Fig. 8 Effect of varying the radius in spatial clustering

similar. Consequently, the accuracy of clustering increases. As the minimum of cluster size increase, clusters become larger and possibility of including more irrelevant events increases and thus the accuracy of clustering drops down as depicted by Fig. 11.



Fig. 9 Effect of varying the event clustering radius on the True Positive and False Positive values of clustering

7.4 Effects of number of time slices

In these experiments, we varied the temporal window of the stream, i.e. number of time slices, of tweets to study the effect of the length of the temporal window on the performance of event detection. For this experiment, we fixed the cluster radius to 1 km and the minimum cluster size to 5. In Fig. 12, the precision and F1 score of the event detection increase as the proposed system, E-ware, considers more time slices. That is because a longer temporal window is likely to contain more of the true events and therefore E-ware was able to detect them.

Figure 13 shows that the percentage of True Positives increases as the length of the temporal window increases. That is, the number of correctly identified events increases when more time slices are considered in the clustering phase. This is logical since a shorter temporal window may either not include some events or partially include some true events, i.e. some tweets of a true event are not cut off by the short temporal window. At the same time, the percentage of False Positives decreases as the number of time slices increases. The results in Fig. 13 confirm our findings in Fig. 12.

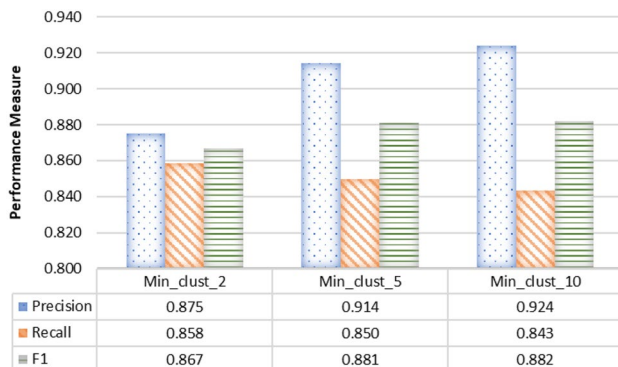
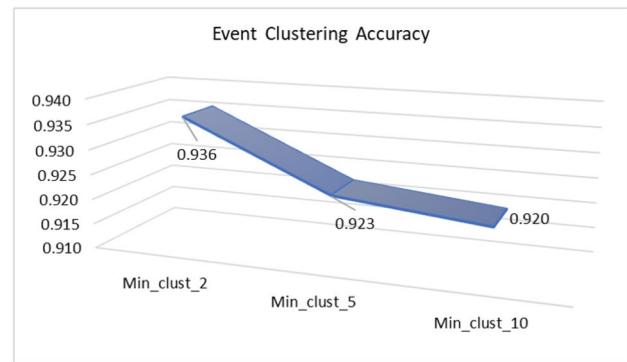
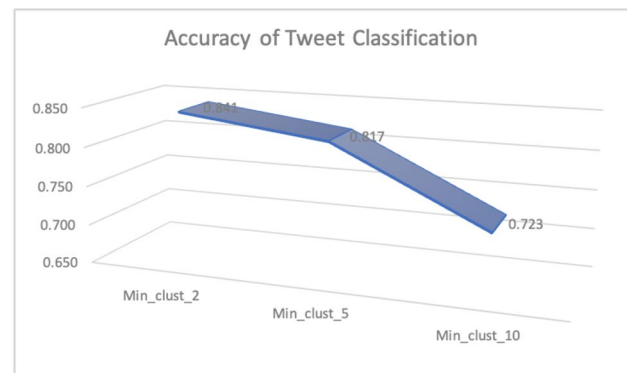


Fig. 10 Effect of varying the event clustering minimum size on the performance measures: Precision, Recall and F1



(a) Clustering Accuracy



(b) Tweet Classification

Fig. 11 Effect of varying the minimum cluster size in spatial clustering

The results above demonstrates that our platform can achieve good performance in terms of incremental event detection accuracy, clustering quality, and also in terms of efficiency to compute in near-real time over continuous sliding windows of data streams.

8 Conclusion

We presented E-ware, a fully-fledged system that integrates a scalable architecture for big social data mining, with an incremental approach to continuously extract the spatio-temporal evolution of event clusters from upcoming data streams. Other approaches only consider one snapshot of historical data, and miss the temporal or spatial component when extracting events. Unsupervised machine learning and NLP algorithms were developed in this research in order to leverage a continuous and incremental event detection process, while taking into account the spatial and temporal dimensions of evolving events. A spatio-temporal indexing scheme was employed using the Geomesa-Accumulo system for the fast retrieval of dynamic events. An extensive evaluation of the developed platform were investigated with respect

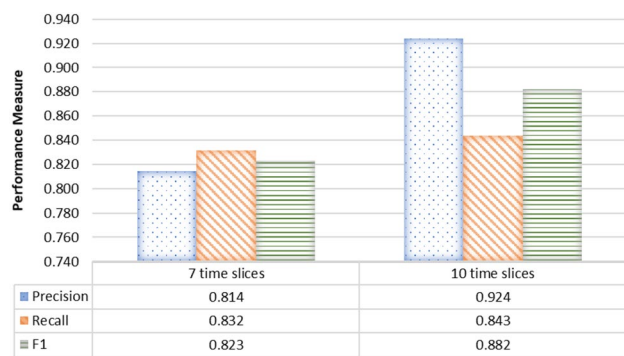


Fig. 12 Effect of varying the temporal window on the performance measures: Precision, Recall and F1

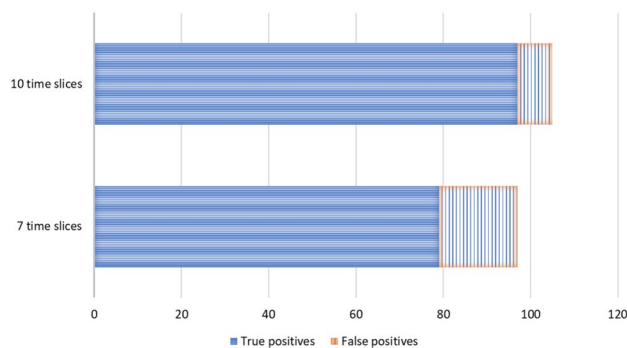


Fig. 13 Effect of varying the temporal window on the True Positive and True Negative values of clustering

to effectiveness and scalability perspectives. E-ware lays the ground for the development of unparalleled smart city applications and location-based services, including smart city exploration on digital maps, trip planning based on dynamic events, and forecasting and tracking of natural disasters (e.g., Floods) and epidemic diseases towards enhancing emergency responses. For instance, among the many applications E-ware can offer, is the development of a smart event-based trip planning system that detects dynamic preference-aware ‘hot’ attractions for travel route recommendation, by considering live spatio-temporal travel constraints. That is hot attractions will be detected as dynamic events from social media, and then combined with the static attractions, such as museums and parks, into one list of attractions. Then, the event-based trip planning system can utilize this list to recommend a number of attractions (dynamic and/or static) based on users’ preferences to visit. Another example of applications can be providing better emergency responses in case of COVID-19 pandemic. For example, a knowledge layer can be built on top of E-ware to recommend a more efficient management of local community interactions, by mining the current hot areas under risk of infection, while taking infected users’ patterns into consideration. That is hot

COVID-19 areas that represent spikes in the stream of tweets will be detected as events. By tracking these hot areas, the system can predict future hot areas of infections. Such information can help health authorities in the respective countries to quickly establish vaccination centers, disease testing centers, mobile hospitals, etc., to combat the spread of the pandemic in potential hot areas.

References

- Adedoyin-Olowe M, Gaber MM, Dancausa CM, Stahl F, Gomes JB (2016) A rule dynamics approach to event detection in twitter with its application to sports and politics. *Expert Syst Appl* 55:351–360
- Afyouni I, Aghbari ZA, Razack RA (2022) Multi-feature, multi-modal, and multi-source social event detection: A comprehensive survey. *Inf Fusion* 79:279–308
- Ahuja A, Baghudana A, Lu W, Fox EA, Reddy CK (2019) Spatio-temporal event detection from multiple data sources. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp 293–305. Springer
- Ai W, Li K, Li K (2018) An effective hot topic detection method for microblog on spark. *Appl Soft Comput* 70:1010–1023
- Alkouz B, Aghbari ZA (2020) Snsjam: road traffic analysis and prediction by fusing data from multiple social networks. *Inf Process Manag* 57(1):102139
- Allan J, Papka R, Lavrenko V (1998) On-line new event detection and tracking. In: *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pp 37–45
- Atefeh F, Khreich W (2015) A survey of techniques for event detection in twitter. *Comput Intell* 31(1):132–164. ISSN 0824-7935
- Belcastro L, Marozzo F, Talia D, Trunfio P, Branda F, Palpanas T, Imran M (2021) Using social media for sub-event detection during disasters. *J Big Data* 8(1):1–22
- Chauhan P, Sharma N, Sikka G (2020) The emergence of social media data and sentiment analysis in election prediction. *J Ambient Intell Hum Comput*, pp 1–27
- Chu C-T, Kim SK, Lin Y-A, Yu Y, Bradski G, Olukotun K, Ng AY (2007) Map-reduce for machine learning on multicore. In: *Advances in neural information processing systems*, pp 281–288
- Cui W, Wang P, Du Y, Chen X, Guo D, Li J, Zhou Y (2017) An algorithm for event detection based on social media data. *Neurocomputing* 254:53–58
- Dabiri S, Heaslip K (2019) Developing a twitter-based traffic event detection model using deep learning architectures. *Expert Syst Appl* 118:425–439
- Derbas N, Dusserre E, Padró M, Segond F (2020) Eventfully safapp: hybrid approach to event detection for social media mining. *J Ambient Intell Humaniz Comput* 11(1):87–95
- Erdem O, Ceyhan E, Varli Y (2014) A new correlation coefficient for bivariate time-series data. *Physica A* 414:274–284
- Fedoryszak M, Frederick B, Rajaram V, Zhong C (2019) Real-time event detection on social data streams. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp 2774–2782. ACM
- Fox A, Eichelberger C, Hughes J, Lyon S (2013) Spatio-temporal indexing in non-relational distributed databases. In: *2013 IEEE International Conference on Big Data*, pp 291–299. IEEE
- Gao Y, Wang S, Padmanabhan A, Yin J, Cao G (2018) Mapping spatiotemporal patterns of events using social media: a case study of influenza trends. *Int J Geogr Inf Sci* 32(3):425–449

- George Yasmeen, Karunasekera Shanika, Harwood Aaron, Li Kwan Hui (2019) Spatio-temporal event detection using poisson model and quad-tree on geotagged social media. In: *2019 IEEE International Conference on Big Data (Big Data)*, pp 2247–2256. IEEE
- George Y, Karunasekera S, Harwood A, Lim KH (2021) Real-time spatio-temporal event detection on geotagged social media. *Journal of Big Data* 8(1):1–28
- Guille A, Favre C (2015) Event detection, tracking, and visualization in twitter: a mention-anomaly-based approach. *Soc Netw Anal Min* 5(1):18
- Han Yi, Karunasekera Shanika, Leckie Christopher, Harwood Aaron (2019) Multi-spatial scale event detection from geo-tagged tweet streams via power-law verification. In: *2019 IEEE International Conference on Big Data (Big Data)*, pp 1131–1136. IEEE
- Hare J, Samangoeei S, Niranjan M, Gibbins N (2015) Detection of social events in streams of social multimedia. *International Journal of Multimedia Information Retrieval* 4(4):289–302
- Hasan Mahmud, Orgun Mehmet A, Schwitter Rolf (2016) Twit-ternews+: A framework for real time event detection from the twitter data stream. In: *8th International Conference on Social Informatics, SocInfo 2016*, pp 224–239. Springer, Springer Nature
- Hasan M, Orgun MA, Schwitter R (2019a) Real-time event detection from the twitter data stream using the twit-ternews+ framework. *Information Processing & Management* 56(3):1146–1165
- Hasan RA, Alhayali RAI, Zaki ND, Ali AH (2019b) An adaptive clustering and classification algorithm for twitter data streaming in apache spark. *Telkomnika* 17(6):3086–3099
- Huang Y, Li Y, Shan J (2018) Spatial-temporal event detection from geo-tagged tweets. *ISPRS Int J Geo Inf* 7(4):150
- Hughes James N, Annex Andrew, Eichelberger Christopher N, Fox Anthony, Hulbert Andrew, Ronquest Michael (2015) Geomesa: a distributed architecture for spatio-temporal fusion. In: *Geospatial Informatics, Fusion, and Motion Video Analytics V*, volume 9473, p 94730F. International Society for Optics and Photonics
- Ifrim Georgiana, Shi Bichen, Brigadir Igor (2014) Event detection in twitter using aggressive filtering and hierarchical tweet clustering. In: *Second Workshop on Social News on the Web (SNOW), Seoul, Korea, 8 April 2014*. ACM
- Jelodar H, Wang Y, Yuan C, Feng X, Jiang X, Li Y, Zhao L (2019) Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey. *Multimedia Tools and Applications* 78(11):15169–15211
- Lee C-H (2012) Mining spatio-temporal information on microblogging streams using a density-based online clustering method. *Expert Syst Appl* 39(10):9623–9641
- Lee H, Abdar M, Yen NY (2018) Event-based trend factor analysis based on hashtag correlation and temporal information mining. *Appl Soft Comput* 71:1204–1215
- Li Quanzhi, Nourbakhsh Armineh, Shah Sameena, Liu Xiaomo (2017) Real-time novel event detection from social media. In: *2017 IEEE 33rd international conference on data engineering (ICDE)*, pp 1129–1139. IEEE
- Liu Jiajun, Li Haoran, Gao Yong, Yu Hao, Jiang Dan (June 2014) A geohash-based index for spatial data management in distributed memory. In: *2014 22nd International Conference on Geoinformatics*, pp 1–4. <https://doi.org/10.1109/GEOINFORMATICS.2014.6950819>
- Liu Shulin, Li Yang, Zhang Feng, Yang Tao, Zhou Xinpeng (2019) Event detection without triggers. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp 735–744
- Liu Y, Peng H, Li J, Song Y, Li X (2020) Event detection and evolution in multi-lingual social streams. *Front Comp Sci* 14(5):1–15
- Martin Carlos, Goker Ayse (2014) Real-time topic detection with bursty n-grams: Rgu's submission to the 2014 snow challenge. *CEUR Workshop Proceedings*
- Massoudi Kamran, Tsagkias Manos, de Rijke Maarten, Weerkamp Wouter (2011) Incorporating query expansion and quality indicators in searching microblog posts. In: *Proceedings of the 33rd European conference on Advances in information retrieval*, pp 362–367. Springer-Verlag
- McInnes L, Healy J, Astels S (2017) hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software* 2(11):205
- Ozdikis O, Karagoz P, Oğuztüzün H (2017) Incremental clustering with vector expansion for online event detection in microblogs. *Soc Netw Anal Min* 7(1):56
- Peng H, Li J, Song Y, Yang R, Ranjan R, Yu PS, He L (2021) Streaming social event detection and evolution discovery in heterogeneous information networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15(5):1–33
- Rehman Faizan Ur, Afyouni Imad, Lbath Ahmed, Basalamah Saleh (2017a) Understanding the spatio-temporal scope of multi-scale social events. In: *Proceedings of the 1st ACM SIGSPATIAL Workshop on Analytics for Local Events and News*, pp 1–7. ACM
- Rehman Faizan Ur, Afyouni Imad, Lbath Ahmed, Khan Sohaib, Basalamah Saleh M., Mokbel Mohamed F. (2017b) Building multi-resolution event-enriched maps from social data. In: *Proceedings of the 20th International Conference on Extending Database Technology, EDBT 2017, Venice, Italy, March 21–24, 2017.*, pp 594–597. OpenProceedings.org
- Rehman FU, Afyouni I, Lbath A, Khan S, Basalamah S (2020) Building socially-enabled event-enriched maps. *GeoInformatica* 24(2):371–409
- Sankaranarayanan Jagan, Samet Hanan, Teitler Benjamin E., Lieberman Michael D., Sperling Jon (2009) Twitterstand: News in tweets. In: *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '09*, pp 42–51, New York, NY, USA. ACM. ISBN 978-1-60558-649-6
- Savaş S (2021) Analysis of the social media impact on the popularity of crypto-currencies. In: *16th International Conference on Computer Science and Engineering*, pp 67–72
- Savaş S, Topaloğlu N, Güler O (2015) The determination of user's preferences on some domain names in turkey: a survey application. *International Journal of Informatics Technologies* 8(2):51–58
- Savaş Serkan, Topaloğlu Nurettin (2017) Crime intelligence from social media: a case study. In: *2017 IEEE 14th International Scientific Conference on Informatics*, pp 313–317. IEEE
- SAVAŞ SERKAN, TOPALOĞLU NURETTİN (2019) Data analysis through social media according to the classified crime. *Turkish Journal of Electrical Engineering & Computer Sciences* 27(1):407–420
- Shah Z, Dunn AG (2019) Event detection on twitter by mapping unexpected changes in streaming data into a spatiotemporal lattice. *IEEE Transactions on Big Data*, pp 1–16
- Unankard S, Li X, Sharaf MA (2015) Emerging event detection in social networks with location sensitivity. *World Wide Web* 18(5):1393–1417
- Wang Z, Ye X (2018) Social media analytics for natural disaster management. *Int J Geogr Inf Sci* 32(1):49–72
- Wazarkar S, Keshavamurthy BN (2020) Social image mining for fashion analysis and forecasting. *Appl Soft Comput* 95:106517
- Wei Hong, Anjaria Janit, Samet Hanan (2019) Learning embeddings of spatial, textual and temporal entities in geotagged tweets. In: *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp 484–487. ACM
- Whitby Michael A, Fecher Rich, Bennight Chris (2017) Geowave: Utilizing distributed key-value stores for multidimensional data. In:

- International Symposium on Spatial and Temporal Databases*, pp 105–122. Springer
- Xin Yanan, MacEachren Alan M (2020) Characterizing traveling fans: a workflow for event-oriented travel pattern analysis using twitter data. *International Journal of Geographical Information Science*, pp 1–20
- Yu Jia, Wu Jinxuan, Sarwat Mohamed (2015) Geospark: A cluster computing framework for processing large-scale spatial data. In: *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, p 70. ACM
- Zhang X, Chen X, Chen Y, Wang S, Li Z, Xia J (2015) Event detection and popularity prediction in microblogging. *Neurocomputing* 149:1469–1480
- Zhou H, Yin H, Zheng H, Li Y (2020) A survey on multi-modal social event detection. *Knowl-Based Syst* 195:105695
- Zhou X, Chen X (2017) Tracing the spatial-temporal evolution of events based on social media data. *ISPRS Int J Geo Inf* 6(3):88

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.