Open access • Journal Article • DOI:10.1109/TVCG.2014.2346922

# #FluxFlow: Visual Analysis of Anomalous Information Spreading on Social Media

— Source link

Jian Zhao, Nan Cao, Zhen Wen, Yale Song ...+2 more authors

**Institutions:** University of Toronto, Massachusetts Institute of Technology, IBM, University of Pittsburgh ...+1 more institutions

**Published on:** 06 Nov 2014 - IEEE Transactions on Visualization and Computer Graphics (IEEE)

**Topics:** Social media, Visual analytics, Information visualization, Interactive visual analysis and Data visualization

# #FluxFlow: Visual Analysis of Anomalous Information Spreading on Social Media

Jian Zhao, Nan Cao, Zhen Wen, Yale Song, Yu-Ru Lin, and Christopher Collins

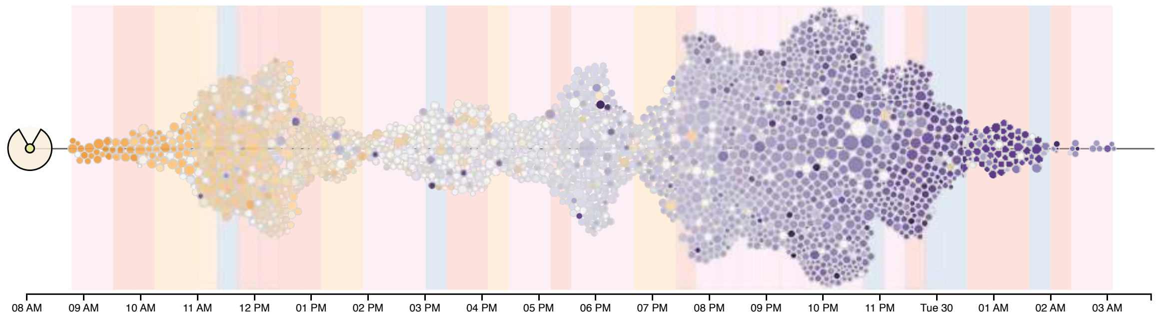


Fig. 1. An overall visualization of the top 100 ranked anomalous retweeting threads during the 2012 Hurricane Sandy. The circles indicate the participating Twitter users in the threads, and the background colors represent the hidden state variables generated by the model, implying the nuances of information spreading patterns. Generally during this 18-hour time period, the anomaly scores of users change from low (brown) to high (purple), and there are three large peaks in user volumes. The last one expresses a plateau of 3 hours with the hidden state staying in mainly "pink", state that frequently appears in abnormal threads (see Section 6).

**Abstract**—We present FluxFlow, an interactive visual analysis system for revealing and analyzing anomalous information spreading in social media. Everyday, millions of messages are created, commented, and shared by people on social media websites, such as Twitter and Facebook. This provides valuable data for researchers and practitioners in many application domains, such as marketing, to inform decision-making. Distilling valuable social signals from the huge crowd's messages, however, is challenging, due to the heterogeneous and dynamic crowd behaviors. The challenge is rooted in data analysts' capability of discerning the anomalous information behaviors, such as the spreading of rumors or misinformation, from the rest that are more conventional patterns, such as popular topics and newsworthy events, in a timely fashion. FluxFlow incorporates advanced machine learning algorithms to detect anomalies, and offers a set of novel visualization designs for presenting the detected threads for deeper analysis. We evaluated FluxFlow with real datasets containing the Twitter feeds captured during significant events such as Hurricane Sandy. Through quantitative measurements of the algorithmic performance and qualitative interviews with domain experts, the results show that the back-end anomaly detection model is effective in identifying anomalous retweeting threads, and its front-end interactive visualizations are intuitive and useful for analysts to discover insights in data and comprehend the underlying analytical model.

**Index Terms**—Retweeting threads, anomaly detection, social media, visual analytics, machine learning, information visualization

## 1 Introduction

Over the recent years, the surge of social media, such as Twitter and Facebook, has significantly advanced the way that people publish, acquire, and share news and information. All day long, millions of messages are created, commented on, and disseminated by over one billion active social media users [10]. Such publicly available texts as well as their propagation patterns among people provide great potential for researchers and practitioners in a variety of fields, such as political science and marketing, to make data-informed decisions.

While there is abundant information on social media, not every posting is equally valuable, important or informative. The first challenging question is: which particular message streams are worth looking into? To be efficient, analysts aim to identify *anomalous* information spreading patterns within the vast and noisy social media data [38]. While very popular trends and newsworthy topics can be easily captured, there exist a wide variety of anomalous conversational threads that are neither bursty enough to trigger trend-detectors nor popular enough to make the news, but have considerable impact on certain people and applications. For example, during 2011 London riots, misinformation spread in social media after an initially peaceful march protesting about the police response to the fatal shooting of Mark Duggan, which significantly fueled the riots, but the government underestimated the risks of these rumors at the beginning [1].

There have been some attempts in developing various algorithms to model and measure information diffusion patterns on social media, thus further suggesting anomalous events and messages [24, 36, 40, 47]. However, since the social media datasets are usually complicated and highly dynamic, it is still difficult for analysts to trust or make use of the results without an in-depth understanding of the automatic methods [28]. For example, one may ask questions about: why certain messages are selected by the algorithm, how they differ from others, and what the abstract variables mean in the model? Hence, there exists a need to involve human supervision in the analysis of anomalous information spreading.

To address the above challenges, we propose FluxFlow, an interactive visualization system for analyzing anomalous information spreading on social media. More specifically, we use micro-blogs captured from Twitter as our source of input and trace the dissemination of in-

- *Jian Zhao is with University of Toronto. E-mail: jianzhao@dgp.toronto.edu.*
- *Nan Cao (contact author) and Zhen Wen are with IBM J. Watson Research Center. E-mails:{nancao,zhenwen}@us.ibm.com.*
- *Yale Song is with MIT. E-mail: yalesong@csail.mit.edu.*
- *Yu-Ru Lin is with University of Pittsburgh. E-mail: yurulin@pitt.edu.*
- *Christopher Collins is with UOIT. E-mail: christopher.collins@uoit.ca.*

*Manuscript received 31 Mar. 2014; accepted 1 Aug. 2014. Date of publication 11 Aug. 2014; date of current version 9 Nov. 2014.*
*For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.*
*Digital Object Identifier 10.1109/TVCG.2014.2346922*

formation through the "retweet" feature, forming a huge repository of retweeting threads. FluxFlow incorporates advanced machine learning algorithms based on the one-class conditional random fields (OCCRF) model [40], to detect anomalous conversational threads in Twitter. We use OCCRF because the data has the "one-class" nature [14], i.e., little knowledge about true anomalies, and highly time-dependent structures (the user retweeting behaviors). In addition to the temporal dispersal patterns of retweeting threads, FluxFlow leverages other important data features as the model input, including those derived from the tweet contents, user attributes, and social network structure.

FluxFlow offers a set of novel visualization designs for presenting the analytical results of the model, allowing users to further comprehend the data with interactive visual explorations. We propose a flexible timeline visualization for retweeting threads by packing small circles representing users without overlaps, which reveals not only overall temporal patterns but also the attributes of participating users (Fig. 1). Multiple coordinated views are applied in FluxFlow to visually summarize and reveal many important aspects of information spreading in Twitter, such as topics, sentiment content, temporal dynamics of the spreading process, and the relationships and connections among threads and authors. This allows analysts to browse and compare different retweeting threads in FluxFlow, such as anomalous ones and others, from several perspectives. Additionally, FluxFlow provides visual access to some low-level information generated by the model, such as hidden variables, to facilitate a deeper understanding about how the algorithm works.

Our key contributions in this paper include: 1) a novel visualization system for the interactive exploration of anomalous retweeting threads with various visual representations and view perspectives, 2) an integrated analysis module consisting of various machine learning algorithms to characterize important features of retweeting threads and perform anomaly detection by applying OCCRF to Twitter data, and 3) a case study and quantitative evaluations that demonstrate the effectiveness of our visualizations and models with real Twitter data captured during Hurricane Sandy and Boston Marathon Bombing.

## 2 Related Work

In this section, we review algorithmic analyses of information spreading and summarize prior approaches to visualizing streaming social media data.

### 2.1 Analysis of Information Spreading

In the social science domain, there has been extensive research on studying the phenomena of information propagation, for example, the "two-step flow" theory of communications [27], stating that ideas flow from mass media to opinion leaders, and from them to a wider population. A recent study on Twitter has also indicated considerable evidence of this theory at work [45]. Moreover, researchers attempt to quantitatively model and measure the process of information spreading on micro-blog platforms [24, 36], and even make predictions of the propagation speed and scale [47].

As the abundance of data to explore on social media can quickly become overwhelming, another main line of research is to develop ways of efficiently identifying useful and valuable information. In a general sense, this means discovering data that is different, unusual or unexpected. In other words, detecting anomalies [14]. For example, Diakopoulos et al. proposed a message uniqueness metric to detect unusual and relevant news events on social media [18]. By considering both spatial and temporal domains, Chae et al. used a seasonal trend decomposition procedure to extract abnormal topics and events [13].

The aforementioned anomaly detection approaches aim to detect unusual *points*, such as outlier peaks in a sequences, for example time-series of topics or events. In this paper we identify abnormal *sequences* (*retweeting threads*) in mass information spreading on social media. Recent advanced machine learning algorithms, such as OCCRF [40], have been proposed to detect anomalous sequences. However, analytic models such as OCCRF and Latent Dirichlet Allocation (widely used in topic modeling), generate abstract scores and latent variables which can be challenging for a human to interpret. FluxFlow integrates interactive visualization techniques and a set of analytical algorithms, including the OCCRF model, to visually summarize many aspects of the data as well as the latent variables, providing a high level overview of the detected threads as well as an exploratory interface of the underlying model states. FluxFlow is the first attempt to apply OCCRF to anomalous retweeting thread detection.

### 2.2 Visual Analytics of Social Media Data

Due to the large size, complexity, and noisy character of data available on social media, researchers have leveraged visualization techniques to assist people with the exploration and analysis. Schreck and Keim provide a broad overview of this area [38]. One of the most important aspects, and our focus here, is the temporal dimension of social media datasets. Thus, below we summarize influential visual analytics techniques for streaming text data, with a focus on social media, in three areas: events, topics, and information spreading. A survey about more general timeline visualization techniques can be found in [6].

Many visualizations have been proposed to visualize time-series events from news resources and document collections. For example, CloudLines describes an incremental visualization to display dynamic event streams as a line of circles with different sizes and opacities governed by an importance function [29]. Luo et al. developed a visual analytics system to detect events from documents with temporal references and visualize them using a bubble-shape representation [30]. As for the exploration of events associated with microblogs, TwitInfo describes an algorithm to detect peaks of high tweet activity and then highlights them in a timeline visualization [32]. LeadLine incorporates multiple sources of online media to characterize different attributes of events including the time, content, location, and people [20]. Other event properties, such as the affective information, have also been investigated. For instance, Adams et al. use the horizontal position and background colors to indicate the mood information of events that are shown as pictures in a 2D view [5].

Another type of information which has been of analytic interest is the evolving topics reflected in social media text. For visualizing text corpora, ThemeRiver [23], which shows the temporal variations of "themes" (similar to topics) in large document collections using a smooth stacked graph, has inspired many of the modern visualization designs. For example, Visual Backchannel uses a similar approach for representing dynamic tweets keywords varying in time [19]. Along the same line, TextFlow proposes a sophisticated layout algorithm to show merging and splitting patterns among evolving topics [17]. Xu et al. added a storyline-style visualization indicating the roles of opinion leaders atop a ThemeRiver graph showing topic competition over time [46]. Beyond stream-style techniques, other types of visual representations of temporal topic variations have been explored. Eddi presents different topics as tag clouds and introduces a new topic assignment schema using searching engines as a distributed knowledge base [8]. HierarchicalTopics organizes a large number of topics into a tree structure where users can make further changes to the hierarchy based on their mental model of the topic space [21].

Recently, several visualizations have been developed to show the patterns of information spreading on social media. Viégas et al. combined node-link views and circular treemaps to visualize the information flow of sharing behaviors on Google+ [43]. Whisper uses a sunflower metaphor to represent spatiotemporal information diffusion on Twitter [12]. There are also some interesting websites worth noting, although not published in academic papers. For example, Project Cascade tracks information propagation on Twitter by providing a 3D visualization that can transform into different 2D charts [33]. Using Riot Rumours, users can drag a time slider to see how rumors unfold using a dynamic circle packing layout [35]. Finally, Revisit displays retweeting threads using a focus+context timeline visualization [41].

Our FluxFlow design has been inspired by many of the above systems. However, most of the previous works have concentrated on visualizing temporal events and topics extracted from social media, rather than the actual information dissemination process, such as conversational threads, which is our focus in this paper. While several recent attempts have been made to monitor information diffusion

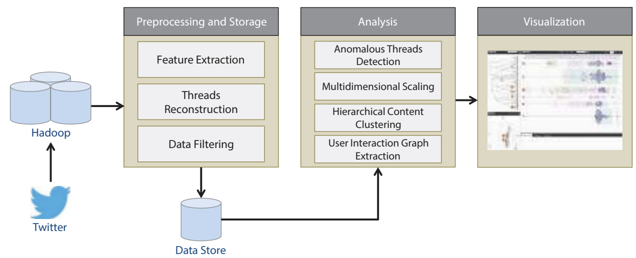


Fig. 2. Overview of FluxFlow system architecture.

patterns on social media as introduced above [12, 43], these systems simply present users with all the data, which can be overwhelming. We contribute a focus on valuable information, such as anomalous threads, through a machine-learning-based comprehensive analysis of large and noisy social media datasets.

## 3 System Overview

The FluxFlow system is designed for detecting, exploring and interpreting anomalous conversational threads in Twitter, consisting of three major components (Fig. 2): a data preprocessing and storage module, a data analysis module, and a visualization module.

The data storage and preprocessing module leverages Apache Hadoop [2] on a cluster, containing three components: data filtering based on user interests (e.g., keywords), retweeting thread reconstruction from the raw tweets, and thread feature extraction. All these components are implemented based on Map-Reduce to support efficient parallel processing of big data. The output is stored in a database designed to support online queries.

In the analysis module, FluxFlow assigns an anomaly score for each retweeting thread and ranks them in a non-increasing order. To further understand the abnormality, we computed contextual information to illustrate: 1) how threads are distributed in the anomaly feature space, 2) how messages under similar topics spread in different ways, and 3) how Twitter users in these threads interact with each other, based on several algorithms such as multidimensional scaling (MDS) [9] and hierarchical topic clustering.

The visualization module displays anomalous threads and their contextual information with various views. As Fig. 5 shows, FluxFlow represents anomalous threads with interactive timelines, the hierarchical clustering of threads in a tree, and their feature-space distributions in a zoomable MDS view. Some other views, including the features view, states view, and raw tweets view are also provided to help analysts explore the data at a lower level.

## 4 Detecting Anomalous Retweeting Threads

In this section, we introduce the techniques of detecting and interpreting anomalous retweeting threads. We first describe the one-class conditional random fields (OCCRF) model for sequential anomaly detection. We then show how this model can be used with Twitter data by extracting relative features. Finally, we put the detected threads back into context to facilitate the interpretation of detected anomalies.

### 4.1 One-Class Conditional Random Fields Model

The problem of detecting anomalous retweeting threads can be cast as *sequential anomaly detection* [14]. It is a challenging task for the following two important reasons: 1) temporal dependency—we need to capture how information is spread over time, and 2) one-class nature—there is little to no example (or even a clear definition) of true anomalies, and the best we can assume is that most of the retweeting thread examples we have are normal.

The second problem is further complicated by the fact that, although we obtain some examples of true anomalies over time, they do not represent the underlying distribution of the anomalous class accurately. In order to identify anomalous retweeting threads successfully, we need a mechanism able to capture anomalous information spreading patterns even without knowing which sequences are anomalous.

To this end, we use OCCRF [40], a recently developed technique that is able to detect sequential anomalies without the guidance of true anomalous examples. It is shown that the OCCRF significantly outperforms traditional anomaly detection algorithms on tasks like identifying detecting insider threats in an organizational network, without using any labeled true anomaly examples. Part of our contribution is the evaluation of this model on detecting real-world anomalous retweeting threads collected from Twitter.

The OCCRF computes an anomaly score of a sequence by measuring how its information spreading pattern is different from a set of (unlabeled) training examples. Specifically, an anomaly score of, e.g., a retweeting thread, $\mathbf{x} = [x_1, \cdots, x_T]$ of length $T$ is defined as the difference between a user-set parameter value $\rho = [0,1)$ and the probability margin measure

$$score = [\rho - \Delta(\mathbf{x};\mathbf{w})]_+ , \tag{1}$$

where each $x_i \in \mathbf{x}$ is a feature vector representing the $i$-th datum in the sequence. In a retweeting thread $\mathbf{x}$, a data item is a retweeting post consisting of two parts: the retweeting message and the user who retweets it. Therefore, each $x_i$ contains both message and user features. $[\cdot]_+$ is a hard-threshold operator that discards any negative value; $\mathbf{w}$ is a model parameter vector (see below); and $\Delta(\mathbf{x};\mathbf{w})$ is the probability margin measure defined as

$$\Delta(\mathbf{x};\mathbf{w}) = p(y = +1|\mathbf{x};\mathbf{w}) - p(y = -1|\mathbf{x};\mathbf{w}). \tag{2}$$

This probability margin computes the difference between the probabilities of a thread $\mathbf{x}$ being normal ($\mathbf{y} = +1$) and abnormal ($\mathbf{y} = -1$). The parameter $\rho$ controls the sensitivity of the algorithm (the higher the more sensitive, and more threads are identified as anomalous).

The conditional probability distribution of a sequence $\mathbf{x}$ being normal and abnormal is defined as

$$p(y|\mathbf{x};\mathbf{w}) \approx \sum_{\mathbf{h}} \exp F(y,\mathbf{h},\mathbf{x}), \tag{3}$$

$$F(y,\mathbf{h},\mathbf{x}) = \sum_t \phi(y,h_t,\mathbf{x}) + \sum_t \phi(y,h_{t-1},h_t), \tag{4}$$

where $\mathbf{w}$ is a vector of unknown model parameters. It can be determined by solving an optimization problem that assumes *most* of the example sequences are collected from the normal class (thus the term one-class), and formulates an objective function such that it accepts most sequences as normal while keeping the solution space *tight*. More details about solving $\mathbf{w}$ can be found in the original paper [40].

Here, we pay more attentions to $h$ in the above model. Specifically, $\mathbf{h} = [h_1, \cdots, h_T]$ is a set of hidden variables introduced to capture the underlying sub-structure of the sequential data. Each hidden variable is of $H$ dimensional ($H$ is a number given by analyzers; in this paper we set it as 8) and each dimension in such variable represents a “hidden state”. A data item $x$ belongs to multiple states at the same time under different probabilities which are described by a state vector $s$.

In different applications, these hidden states can be interpreted differently. In the case of analyzing retweeting threads, a data item in the sequence, $x_i \in \mathbf{x}$, is described by the features of a retweet message and the user. Considering that the retweeting messages in the same thread remain the same, i.e., all are in forms of “RT + original tweet”, $x$ is actually determined by the user features. Therefore, the “hidden states” of $\mathbf{x}$ can be interpreted as soft communities of users in which users are clustered based on their anomaly features. The transition of the states in a retweeting thread, thus, captures the information spreading pattern among user groups over time.

Once a set of retweeting threads are scored using Equation (1), we sort them by their anomaly score $p(y = -1|\mathbf{x};\mathbf{w})$ in an non-increasing order, generating a ranked list of abnormal retweeting threads.

### 4.2 Applying OCCRF to Twitter

We applied OCCRF model to detect anomalous retweeting threads based on a set of features extracted from Twitter data. Particularly, to characterize Twitter user behaviors, we first built a Twitter user interaction graph based on their interactions with each other (e.g., retweet and mention) [25]. The weight of the link from user $a$ to user $b$ was computed based on the number of retweets and mentions from $a$ to

| Feature (Type) | Description |
|---|---|
| UserFriendsCount (C1) | The user's friends count |
| UserFollowersCount (C1) | The user's follower count |
| UserStatusesCount (C1) | The user's lifetime tweet count |
| RegistrationAge (C1) | The number of days since the user registered |
| FriendsFollowerRatio (C1) | The ratio between users' friends and followers |
| UserAnomalyScore (C1) | The number of the user's interaction in a time window (e.g., 3 hours) divided by the user's monthly average |
| MentionCount (C1) | The number of mentions in the user's tweets of a time window |
| UrlCount (C1) | The number of urls in the user's tweets of a time window |
| HashtagCount (C1) | The number of hashtags in the user's tweets of a time window |
| maxOutTie (C2) | The maximum weight of the links from the user to other users ahead in the thread |
| maxInTie (C2) | The maximum weight of the links to the user from other users ahead in the thread |
| similarity (C2) | The egonet similarity between the user and other users ahead |
| UserIndegree (C2) | The in-degree of the user in his/her egonet |
| UserOutdegree (C2) | The out-degree of the user in his/her egonet |
| RetweetCount (C3) | The retweet count of the orignal tweet in the thread |
| DeviceCount (C3) | The total count of the device where the tweet is from in a period of time (the larger the value indicates a popular device) |
| Interval (C3) | The log of the interval between adjacent tweets |
| TimeOfDay (C3) | Time of the day when the original tweets are retweeted |
| HasQuestionMark (C4) | Whether the tweet contains question mark |
| Lexical emotion (C4) | 220 categories in psychological dictionaries (e.g., "nice", "sweet" for positive emotions) |

Table 1. Anomaly features extracted for a retweeting thread.

$b$. The interaction graph was built using the 10% Twitter feed in 2012, containing 95 million nodes and 1.8 billion edges. In addition, we computed each user's monthly average number of interactions. After that, we extracted a feature vector to represent an incoming retweeting thread, $\mathbf{x}$, in live Twitter streams. We extracted four types of features:

**C1. User profile features.** We extracted user profile statistics such as the counts of followers, friends, status, and so forth. These features indicate how active and influential a user is. In addition to user profile directly provided by Twitter, we computed features proposed in [16] including URL ratio in a user's tweets, hashtag ratio, registration date, etc. These features have been shown useful to detect bots in Twitter, which may be involved in anomalous events such as spreading rumors. Further, we computed a user anomaly score to indicate how much the number of interactions deviates from his/her monthly averages.

**C2. User network features.** Users' EgoNet [1] features such as in-degree and out-degree were extracted based on the interaction graph we built. Such features indicate if they are good at interacting with others and thus more influential. More importantly, we measured the relationship among users in the same retweeting thread, because a strong "clique" of them increases the possibilities of collusion. For example, we computed the maximum weight of a user's incoming and outgoing links from/to all other users ahead of him/her in the thread.

**C3. Temporal features.** We extracted features specific to the current tweet in the thread, such as retweet count at this point and whether this tweet is from a popular device. In addition, we computed the log of the intervals between two adjacent tweets in the sequence. This feature helps to distinguish bursty sequences from slow sequences.

**C4. Content features.** To characterize the content of tweets, we extracted the count of psychological keywords as defined in dictionaries such as LIWC [42], which gives us indicators of the original author's emotion as well as others' response. We hypothesized that anomalous events would trigger anomalous emotional response. Although users can add content when replying, we observed that they seldom do so. Thus, the content feature have little variation across the thread.

The above features express a retweeting thread from different perspectives: user profile and user network features measure the anomaly at the individual level, and the temporal and content features are at the thread level. We extracted 239 features in total, and most of them (220 out of 239) look for psychological keywords defined in dictionaries such as LIWC. These features are summarized in Table 1.

[1]EgoNet: A network which is centered on an individual (the ego) and the people he or she is connected to (the alters).

### 4.3 Interpreting Anomalies in Context

The OCCRF model computes an anomaly score for each retweeting thread without giving any intuition behind it, thus making the results difficult for analysts to interpret or trust. Therefore, FluxFlow provides several kinds of extra information about the retweeting threads in different context to assist the understanding of those anomaly scores.

The feature differences between an anomalous thread and others can be the most intuitive interpretation of why a thread is considered to be abnormal. A direct comparison of the feature vectors is difficult given the dimension is too high for analysts to capture their similarities or differences. Thus, we employed MDS [9] to provide a 2D overview of thread distributions in the feature space, where thread similarities are revealed by the 2D distances between them. The feature vector of each thread in the MDS projection were defined in two ways: a mean feature vector of all user features $\mathbf{x}$, and a mean state vector of all user states $s$, providing two contexts for different analysis purposes. The first one captures the distribution of threads in raw feature space, and the second one represents threads with the perspective of OCCRF.

A thread may also considered to be abnormal when it disseminates a message differently from the information spreading patterns of other threads under a similar topic, which is also one of the major design considerations of the OCCRF model. To facilitate such comparisons in Twitter data, we clustered the threads hierarchically with a top-down approach based on their topical keywords. More specifically, we extracted a set of high frequency unigrams and bigrams from the tweets as content features, and then applied meanshift [15] for clustering recursively to drill down the dataset in a hierarchy until all cluster sizes are smaller than a given threshold. To allow more insights about the content, we also computed the sentiment of a tweet based on the technique described in [34]. Specifically, we trained a sentiment classifier with the multinomial Naïve Bayes model using the presence of a bigram as a binary feature.

The third type of contextual information worth investigating is the interactions between users, which may also imply why a thread is abnormal. For example, a potential rumor spreader might be densely retweeted by others. In Twitter, users interact with each other via retweeting or mentioning. Retweeting information is only partly provided in Twitter data: all retweets point to the original tweet owner, so it is unknown who retweets whom exactly in a thread. Thus, we extracted all historical user interactions based on the interaction graph discussed in Section 4.2, allowing analysts to identify potential retweeting behaviors between different communities of users.

With all the above data in hand, in the next section we design visualizations to represent following information computed in the analysis module: 1) retweeting threads with OCCRF's rankings and anomaly scores, 2) the hidden states and feature vectors used in OCCRF, 3) MDS projection in the feature space, 4) the hierarchical topic clusters of threads, and 5) historical interactions among users.

## 5 Visualizing Retweeting Threads

In this section, we first discuss the rationale influencing the overall FluxFlow design, then introduce the main visual encodings used to represent retweeting threads, and finally present the entire interface.

### 5.1 Design Rationale

To design the interface of FluxFlow, we conducted multiple design sessions with three domain experts who belong to a research consortium focusing on anomaly detection and social media understanding. Two of them specialize in machine learning and data mining approaches for analyzing large-scale social media data, and the third one focuses on both computational and visualization methods for understanding social network dynamics. The consortium also holds regular meetings with everyday end-users, such as government analysts and industry practitioners. We discussed with these experts about the challenges in their work, both internal and external. For example, they had difficulties in understanding such as why certain information is treated abnormal by the algorithm, how it differs from the rest, and what the final and intermediate outputs mean in the model. In general, they wanted a system that can visually summarize information diffusion

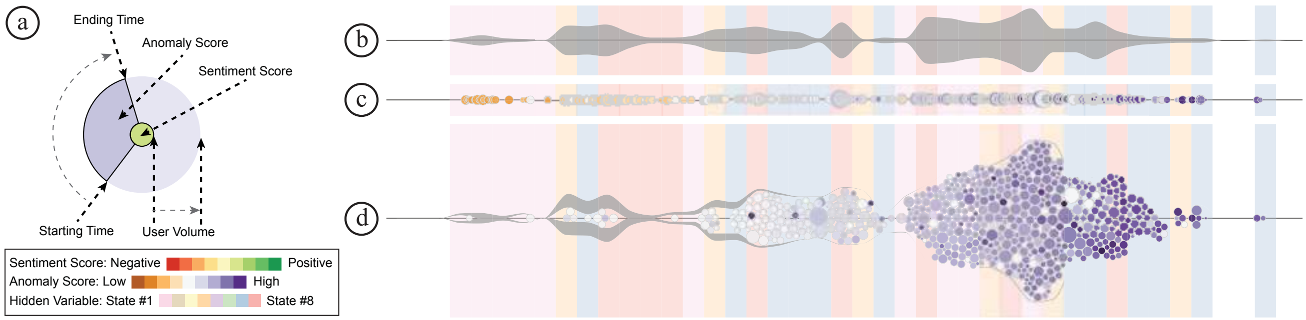


Fig. 3. The main visual encodings in FluxFlow: a) a thread glyph for aggregating the main information, and three thread timeline visualizations for unfolding the temporal patterns with different perspectives, including b) a volume chart, c) a linear circle view, and d) a volume circle view. The backgrounds are color-coded by eight hidden states generated in the model. In d) the volume circle view, users with low anomaly scores are aggregated into the "gray ribbons" in contrast to c) the linear circle view.

patterns to assist the exploration and provide means of identifying and interpreting the anomalies. Based on our consultation with the experts and the previous work, we distilled the following design guidelines for developing visual analytics systems for information spreading.

R1 *Summarizing and aggregating important features of retweeting threads.* Due to the scale and noisiness of data, important attributes of the threads, such as the actual message contents, features used in the anomaly detection, and other meta-data information, should be visually summarized and aggregated when appropriate, to facilitate the discovery of interesting subsets of data [22, 39].

R2 *Indicating characteristics and connections of involving users.* The people who participate in disseminating the tweets is a key facet of information spreading [27]. Thus, the system should present important user characteristics, such as the number of followers, and users' social relationships, to help analysts find influential people and understand the effects of the social network topology.

R3 *Revealing temporal patterns of information spreading.* The process of how a message is propagated among members of the network, indicated through the time dimension of threads, is critically important to analysts. Hence, intuitive visual metaphors for summarizing the trends and other temporal dynamics of information spreading should be included in the system to illustrate the "when" aspect on top of "who says what to whom" [45].

R4 *Facilitating visual data comparisons and correlations.* The key to understanding the patterns of retweeting threads is to compare and correlate them, such as between anomalous threads and others. Hence, the system should facilitate data comparisons through well-designed visual encodings and interactions; data relationships should be also revealed visually, such as identical people participating in different threads.

R5 *Providing diverse data perspectives and views.* At anytime, analysts may require access to multiple aspects of the data, such as the overall thread relations, the temporal trends of a thread, and the detailed features used in the anomaly detection. Therefore, variations of visual representations showing different data perspectives and multiple coordinated views should be supported [7].

R6 *Accessing deep-level information of the model and input.* Apart from the final output provided by the model, some lower level results generated during the analysis process, such as the abstract or hidden intermediate variables, need to be exposed to users when necessary, thus enabling a deeper understanding of the algorithm mechanisms and better human steering of its performance.

### 5.2 Visual Representations of Retweeting Threads

Following the above design rationale, we created a set of visual encodings in FluxFlow for summarizing a retweeting thread, the most important data entity in the input.

#### 5.2.1 Thread Glyphs

We designed a circular glyph to visually summarize important aspects of a retweeting thread in a compact form, as in Fig. 3-a. We selectively

```
Data: A list of circles C_i : (r_i, x_i) sorted ascending by x-constraint x_i
Result: A list of layout circles C_i : (cx_i, cy_i, r_i)
start ← 0, bounds ← [0,0], frontchain ← {};
foreach circle C_i in the input do
    if bounds is empty or C_i intersects with bounds then
        if i − start ≤ 3 then      /* the first three are trivial */
            compute (cx_i, cy_i) to place C_i at an appropriate location;
            update bounds and add C_i to frontchain;
        else                       /* find the best placement location */
            locations ← {};
            foreach circle C_j in frontchain do
                if C_j intersects with [x_i − r_i, x_i + r_i] then
                    attempt to place C_i next to C_j and C_{j+1} on frontchain;
                    add the placement position (lx, ly) to locations;
                end
            end
            sort locations ascending by the distance to (x_i, 0);
            assign (cx_i, cy_i) with (lx_0, ly_0) to place C_i;
            update bounds and add C_i to frontchain as in [44];
        end
    else                           /* start a new placment block */
        start ← i + 1, bounds ← [0,0], frontchain ← {};
    end
end
```

Fig. 4. Circle packing algorithm with horizontal constants.

encoded a number of critical and easily-understood variables associated with a thread, which allows analysts to quickly and intuitively capture the key characteristics (R1), including its overall abnormality, contextual polarity, scale, and temporal information.

More specifically, two numerical scores of the thread, the tweet sentiment score and the thread anomaly score, are encoded with the colors of the inner and outer circles respectively, with two different color schemes selected from [11]: red-green (where red is the most negative), and purple-yellow (where purple is the most abnormal). Further, the number of participating users is mapped to the radius of the outer circle, and the temporal duration of this thread is represented by the wedge on top by placing its starting and ending timestamps with a clock metaphor, where the global timeline is the full circle.

#### 5.2.2 Thread Timelines

To unfold the temporal aspects of retweeting threads (R3), we further developed three timeline visualizations to provide different data perspectives (R5), including a *volume chart*, a *linear circle view*, and a *volume circle view* (Fig. 3). The background of timeline views (that can be made invisible as shown in Fig. 6) is used for color-coding transition patterns of the hidden states generated by OCCRF (R6).

The volume chart (Fig. 3-b) shows the temporal trends of user volume in a retweeting thread using Bézier curves, which can be further extended to graphs such as ThemeRiver [23] to display different types of users, e.g., males and females, if the data is available. The linear circle view (Fig. 3-c) is designed to precisely illustrate the timestamp of each retweet event, displaying each individual user as a small circle on the time axis, where size and color indicate the number

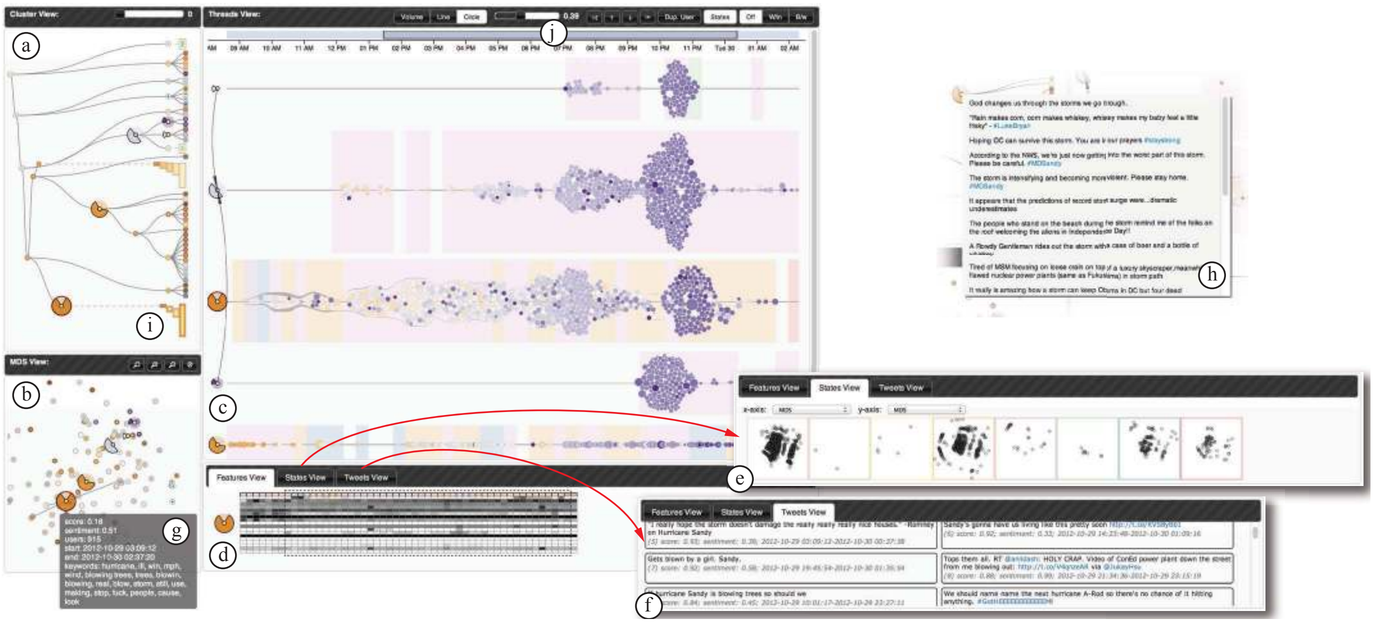


Fig. 5. The FluxFlow visual interface contains four interactively coordinated UI components, including a) a cluster view, b) a MDS view, c) a threads view, and a detail information panel with three subviews: d) a features view, e) a states view and f) a tweets view. Extra information such as the meta-data of a thread or the tweet contents can be assessed through g) informative tooltips and h) context menus. The analyst can also perform flexible exploration of retweeting threads at multiple scales, such as i) aggregating tree branches in the cluster view, and j) zooming the timelines using the time window in the threads view.

of followers and anomaly score of that particular user respectively. To avoid visual clutter, techniques inspired by Cloudlines [29] can be applied to control the circle opacity and size based on their importance.

We also propose a novel visual representation for retweeting threads, the volume circle view (Fig. 3-d). Important thread participants are displayed as circles (using the same encoding as the linear circle view) that are densely packed without overlaps along a timeline. Less important users are aggregated into two gray ribbons similar to the volume chart (R1). In FluxFlow, an analyst can define a threshold of the user anomaly score to determine which visual forms to show (users greater than the threshold are shown as circles). However, the visualization is not restricted to this particular importance measure. This volume circle view combines the benefits of both previous views, indicating the temporal trends of retweet volume with the overall shape and the information of individual users with circles. It is interesting to note that when the anomaly score threshold is set to 1, the volume circle view smoothly transforms into the volume chart by aggregating all users into the ribbons.

To place user circles in the volume circle view, we developed a greedy layout algorithm based on a circle packing approach (Fig. 4). Our layout algorithm extends the approach of Wang et al. cite-Wang2006 by accommodating horizontal constraints (user time-axis positions) when packing circles, so that the temporal trends of retweeting threads are preserved. While other layout algorithms could be applied to achieve similar results, such as a force-directed layout with collision detection, we chose a circle packing approach because it is efficient and the resulting layout is deterministic, unlike unstable force-directed layouts which generate different results on each run.

## 5.3 FluxFlow Interface

According to the aforementioned design rationale, we developed the front-end interface of FluxFlow as a web application (Fig. 5). FluxFlow consists of four interactively coordinated UI components that serve different analytical purposes (R5), including: a) a cluster view, b) a MDS view, c) a threads view, and a detail information panel with three subviews containing d) a features view, e) a states view and f) a tweets view. Brushing and linking techniques are applied to relate visualization objects across different views. Also, the FluxFlow design follows a consistent visual language with smooth animations.

### 5.3.1 Seeing the Big Picture

With the context information described in Section 4.3, FluxFlow offers two overviews to allow users to capture the general picture of data from different perspectives (R1). The cluster view groups all retweeting threads hierarchically based on features extracted from the tweet texts; and the MDS view summarizes the relationships of threads in a high-dimensional feature space used for the anomaly detection.

The cluster view (Fig. 5-a) reveals the content similarities among retweeting threads in a dendrogram, where each internal tree node represents an aggregate of related retweeting threads, allowing users to navigate the data in a sense of topics and keywords. Initially all nodes in the cluster view are represented with small circles where their interior and outline colors are mapped to the thread sentiment and anomaly scores. Several interactive features are integrated to facilitate the exploration of this clustering tree. First, to accommodate the navigation of large hierarchies, we developed a compact visual representation of tree branches based on the visualization proposed by Zhao et al. [48]. As indicated in Fig. 5-i, each bin corresponds a level of the branch with its height mapped to the number of nodes and vertical position governed by the centroid node locations. This visualization summarizes the general shape of the branch as well as the information of nodes in each level. Second, by manipulating the slider on the toolbar, FluxFlow allows a user to quickly collapse all nodes below certain tree level, shown in the compact forms, which is suggested as the *above traversal* paradigm in [22].

In the MDS view (Fig. 5-b), FluxFlow shows the distributions of threads with MDS projection from the high-dimensional anomaly feature space, allowing users to identify outliers and visually compare the threads at a higher level (R4). The MDS view applies consistent visual encodings to the thread nodes as the cluster view, and supports several basic interactions such as zooming and panning. In FluxFlow, both the leaf and internal threads in the clustering hierarchy are shown in the MDS view; however, a user can choose to hide all the internal nodes for other analysis purposes.

In addition, FluxFlow includes several mechanisms to coordinate these two overviews, to better assist users' comprehension of data. For example, nodes in collapsed branches in the cluster view are shown semi-transparently in the MDS view. Hovering over a parent node in the MDS view displays links to its child nodes (Fig. 5-b). Both views provide informative tooltips (Fig. 5-g) and context menus (Fig. 5-h)

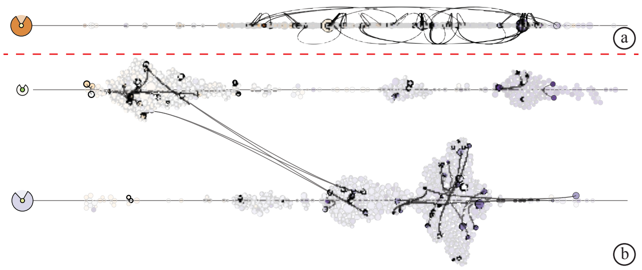


Fig. 6. User social network connections overlaid on top of thread timelines: a) user links of one thread in the linear circle view, and b) user links between and within two threads in the volume circle view.

to access extra information of the thread nodes, such as the tweet consents, exact score values and starting/ending timestamps.

### 5.3.2 Looking into Individual Threads

When the user identifies something interesting by interacting with the cluster view and the MDS view, she can double-click a node in either to unfold the timeline visualization of that thread in the threads view, and the selected node will be displayed as a thread glyph accordingly in the two overviews, as shown in Fig. 5.

Since the thread glyph and timeline designs are not restricted for visualizing just one thread, both leaf and internal nodes can be added to the threads view, providing a multi-level visual exploration and comparison of data with aggregations (R1 and R4). The parent-child relationships between thread nodes are also indicated with arrows connecting the thread glyphs on the left of the view (Fig. 5-c). Moreover, for each thread timeline, the analyst can highlight retweets from each of its sub-threads by hovering over the descendant nodes in either the cluster view or the MDS view. For more detail, the analyst can dive into a thread timeline by splitting it into multiple child threads in-place in the threads view, using a toolbar button.

FluxFlow supports a number of interactions related to temporal exploration of retweeting threads, allowing users to discover the trends and other temporal dynamics of information spreading (R3). For example, multi-scale navigations along the time dimension, such as zooming and panning, can be easily operated by directly brushing and dragging a time window on the axis on top of the thread view (Fig. 5-j). In addition to this absolute time axis, the user can also align all the retweeing threads to the same starting point in time to perform side-by-side comparisons in a relative manner (R4).

Another important aspect of concern to analysts is the users involved in the retweeting process (R2), and thus FluxFlow integrates several functions to facilitate the process of discovering user relationships. First, duplicated users within the same thread or across different threads can be highlighted by toggling a button on the toolbar, which not only allows thread comparisons at the user-level but also the identification of suspicious users in the case that a particular user appears in multiple anomalous threads. Second, FluxFlow can further reveal the user social connections at the intra- or inter-thread level by overlaying links on the timeline views (Fig. 6), based on the results of our computations described in Section 4.3. To avoid visual clutter, hierarchical bundling of the links is applied by first clustering the starting and ending user nodes based on their layout positions, and the vertical order of thread timeline views can be adjusted when necessary.

### 5.3.3 Revealing Deep-Level Information

Sometimes the user wants to know further about the analysis behind the visualization and the raw data to help her make better decisions, and thus the visualization cannot be a black box of the analysis process (R6). As introduced in Section 4.1, generally OCCRF extracts features from the involving users in retweeting threads and perform anomaly detection with eight hidden states capturing the sub-structures of users spreading the information. Thus we designed a number of ways to visually uncover deeper information about the model.

For example, the hidden state transitions shown as the background of thread timeline views (Fig. 5-c) can reveal the internal stage of OCCRF. By comparing the state transition patterns across threads, the user is able to obtain more knowledge about how the model relies on these states, i.e., user community sub-structures, to perform anomaly detection. To further look into the state variables, the analyst can leverage the features view which summarizes the temporal variations of feature vectors described in Section 4.2 with a heatmap-like visualization (Fig. 5-d). A coupled zooming mechanism with the threads view is also incorporated to enable the multi-scale exploration.

From a different perspective of viewing these abstract state variables (R5), the states view indicates how states are tied to tweet users by displaying the MDS projections of all users from the high-dimensional feature space (Fig. 5-e). Additionally, the user can set the axes to represent specific features, forming a scatter-plot of users with the correlations between features. The distributions of users in these charts can be viewed as signatures of the states characterizing the features, which helps the analyst understand what each of the abstract variables might mean.

## 6 Evaluation

We assessed the effectiveness of FluxFlow's analytical models and visualizations using two 10% Twitter feed datasets collected during two significant events: 2012 Hurricane Sandy and 2013 Boston Marathon Bombing. In this section, we report the quantitative performance of our anomaly detection model over the two datasets, describe one analysis use case developed based on our interviews with domain experts, and discuss their general comments about FluxFlow.

### 6.1 OCCRF Evaluation in Twitter

The Hurricane Sandy dataset contains 52 million tweets during Oct 29, 2012; and the Boston Bombing dataset contains 242 million tweets from Apr 15 to Apr 19, 2013. Since it is infeasible to collect all misinformation during these two events, we chose to evaluate the accuracy instead of recall for our approach. We compared the OCCRF approach with One-Class SVM (OCSVM) [37], one of state-of-the-art unsupervised anomaly detection methods. We chose OCSVM as the baseline because it can achieve comparable performance against other existing methods (e.g., HMM, Active Outlier, etc.) [40]. Further, to make OCSVM more comparable, we concatenated the features of data points within a time window into one long feature vector to introduce the time-dependency. For the union of the top 500 anomalous retweeting threads detected by both models, we asked three annotators to label whether a sequence is misinformation based on reports after the events, such as [3, 4]. The comparison of the accuracy at top-K is shown in Fig. 7. We can observe that OCCRF can effectively detect rumors and significantly outperforms the baseline.

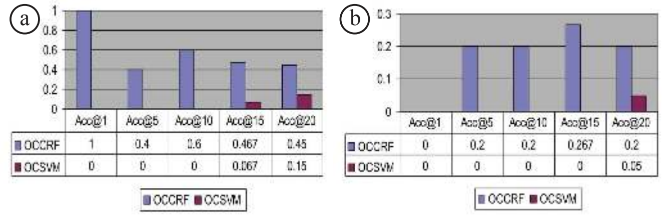


Fig. 7. Accuracies of OCCRF and OCSVM in correctly detecting rumors in the top-K retweeting threads ranked by the models in datasets (i.e., Acc@K): a) Hurricane Sandy, and b) Boston Bombing.

Moreover, we performed a preliminary qualitative comparison of the four types of features used in OCCRF (Table 1). Following the leave-one-out methodology, we used only three types of features in each round and asked the annotators to evaluate the model outputs. The results show that temporal and user network features were the most important. In contrast, the content features are the least influential, which might be because it is very difficult to reliably analyze the short, noisy and informal Twitter text.

### 6.2 Case Study: Hurricane Sandy

In 2012, Hurricane Sandy impacted people's lives in several countries, including the United States and Canada. During the event, a vast

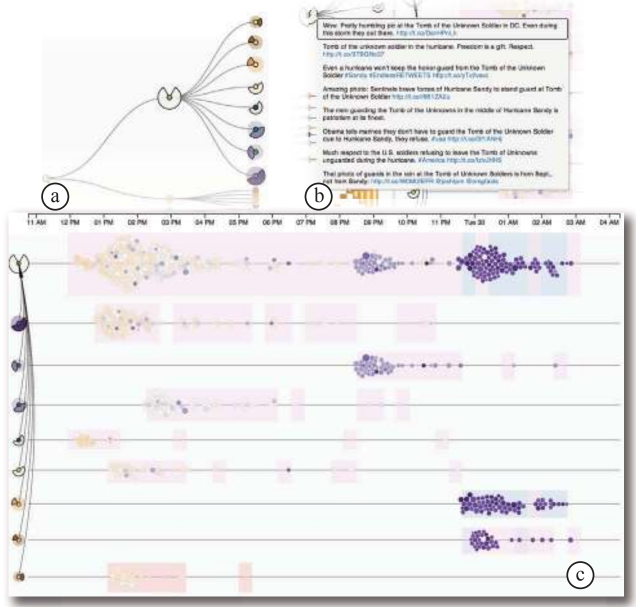


Fig. 8. Retweeting threads about the topic of “a picture of the Tomb of Unknown Soldier” during the Hurricane Sandy event: a) thread nodes indicating the topic in the cluster view, b) raw tweets shown in the context menu, and c) thread timelines in the threads view.

amount of information was spread between people through social media, of which many messages contained misinformation. The goals of this case study include exploring the data in Twitter, identifying anomalous conversational threads, and examining the internal mechanisms of the analytics model. To fully demonstrate the features of FluxFlow, we synthesized the following use case based on observations and comments from in-depth interviews with the same three domain experts who we worked with to derive the design requirements (see Section 5.1).

**Bottom-up approach.** An analyst loads the top 100 abnormal retweeting threads into FluxFlow, as ranked by anomaly score. She knows from experience that a most rumour threads have an anomaly score above 0.25. From the raw tweets view, she sees the range of anomaly scores is from 0.98 to 0.03, indicating that the top 100 should be sufficient for identifying misinformation. The analyst decides to start from exploring original individual retweeting threads in data. Thus, she first choses to show only the leaf threads in the MDS view, and uses the colors and positions of the thread glyphs to select a number of potential anomalous ones to importor into the threads view. Within the selected threads displayed as timelines, one with very high anomaly score (0.97) and interesting content catches her eye: *“Wow. Pretty humbling pic at the Tomb of the Unknown Soldier in DC...”*, which is retweeted 113 times. Thus she minimizes other thread timelines into the volume view or linear circle view to save space. Next, through the cluster view (Fig. 8-a), the analyst finds its parent thread whose tweets actually all talk about *“a picture of the Tomb of Unknown Soldier”*, lasting about 14.5 hours and involving a total of 407 users, based on the context menu (Fig. 8-b). Then she adds the parent node to the threads view as well, and the volume circle view of this thread indicates there exist multiple peaks in user volume and the user anomaly scores generally transit from low to high along time. To further drill down for this topic, the analyst loads all the children threads into the threads view by pressing the “expand” button on the toolbar (Fig. 8-c). In one child thread that is full of “purple” abnormal users, she even reads: *“Obama tells marrines they don’t have to guard the Tomb of the Unknown Soldier. They refuse.”*, which sounds very bizarre but is retweeted by 63 users within 3 hours. This was proven to be a rumor afterwards — the photo was actually shot in September [3].

**Top-down approach.** On the other hand, the analyst explores the dataset from a higher level by manipulating the cluster view. Since there are too many threads in the hierarchy, she drags the scrollbar on top of the view to collapse nodes below a certain level. After

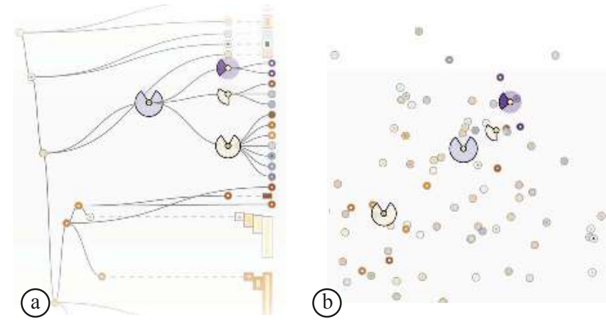


Fig. 9. Exploring one thread of interest and its three child nodes with sub-topics: a) the cluster view and b) MDS view.

some exploration, the analyst identifies one thread with relatively high anomaly score (0.62), and from the tooltip she can see a few interesting keywords of the tweets, such as “power plant”, “home” and “swim”. After unfolding the timeline of this node in the threads view, the analyst finds that, except a few users at the beginning, this thread contains many users with high anomaly scores. A further visual aggregation of users with low anomaly scores into ribbons indicates that most of the users have scores above 0.75. In general, the user volume seems to form three peaks, with a large burst in the end (the 2nd thread in Fig. 5-c). As expected, when the analyst expands the thread node in the cluster view, three child threads appear, indicating the three sub-topics may relate to the peaks in the thread timeline. With various visual encodings of the thread glyphs, she see that two of them last very little time, including one with a significantly higher anomaly score (Fig. 9-a). Additionally, these two threads appear much closer in the MDS view, indicating their strong similarity in the feature space (Fig. 9-b). By hovering over those child nodes, which highlights the corresponding user circles of the timeline, the analyst further observes that the final burst is caused by the two shorter threads (the 1st and 4th threads in Fig. 5-c). The tooltips and context menus indicate that one is about *“power outage in North America”* (152 users involved) and the other is related to *“a shark swimming down the street”* (107 users involved). Both have been indicated as rumors later [3].

**Connection and Comparison.** Further, the analyst wants to see if there is any relations between the participating users in the two “rumor” topic threads explored in the above two approaches. Thus, she opens these two anomalous retweeting threads in the threads view, and turns on the “show duplicated users” and “show user connection” functions (Fig. 6-b). Identical users in those two threads are highlighted with black outlines and the estimated social network connections are shown as links on top. The analyst observes that across the two rumor retweeting threads, there are many overlaps, and a few users have close social network relationships, implying that those users might merit further investigation into their rumor-spreading behavior.

Next, the analyst wants to thoroughly understand the model mechanisms by comparing retweeting threads, such as rumors vs. ordinary tweets, and high anomaly score threads vs. low ones. Thus, in addition to the above 3 identified rumors, she samples some other interesting threads with the help of the cluster view and MDS view, in total 6 threads (2 in each of the high, medium, and low anomaly score levels). To facilitate the visual comparison of thread timelines, the analyst aligns all threads to the same horizontal position by shifting their starting times (Fig. 10-a). From the overall shapes of user volumes, threads that have short-time bursts or long tails are likely to be assigned larger anomaly scores (first two threads in Fig. 10-a). Also it is interesting that user anomaly scores, measuring the user activity deviations, do not have determinative effects alone on the abnormality of threads. For example, in the 2nd and 3rd threads of Fig. 10-a, the thread anomaly score and user anomaly scores tend to disagree. Thus future studies are needed to better understand the OCCRF model.

**Deeper insights of the model.** By comparing the threads’ background colors, which encodes the transitions of hidden state variables in the model, the analyst finds an interesting pattern that highly anomalous retweeting threads mostly remain in the “pink” state, and

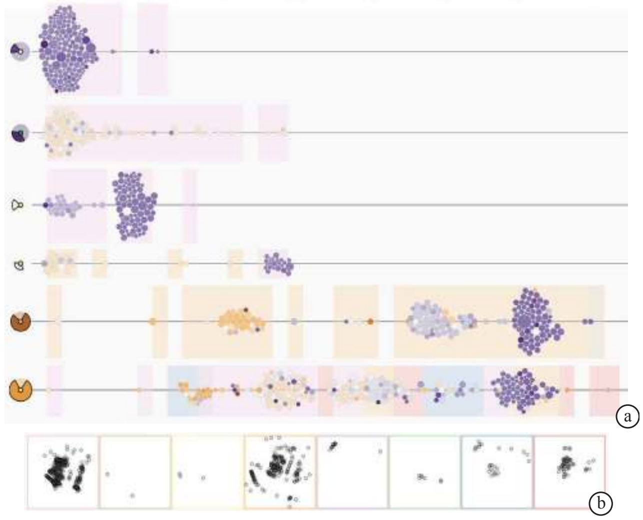


Fig. 10. An analyst is trying to better understand how the anomaly detection model works using a) visual comparison of threads with different abnormality scores, and b) examination of the MDS projection patterns of the hidden states. In a), the first three are identified rumors with topics about "power outage", "the Tomb of Unknown Soldier picture" and "shark swimming in the street" respectively.

those with lower anomaly scores are likely to have more variations of state colors (Fig. 10-a). The misinformation seems to have stronger correlation with the hidden states. For example, the 3rd thread in Fig. 10-a is a rumor (i.e., *"a shark swimming down the street"*) which is in the "pink" state though its anomaly score is not very high (0.57). However, the 4th thread, with similar anomaly score but different set of states, is not a rumor. To further examine the meaning of those abstract states, the analyst opens the states view to observe the MDS projections of users associated with different state variables, where the "pink" state seems to have different projection patterns from others (Fig. 10-b). Now the analyst wants to dig further about the underlying features of the analytical model in FluxFlow. Thus she explores the temporal feature variations of a couple of threads using the feature view. From the heatmaps of feature values, the analyst discovers that threads with higher anomaly scores tend to have larger values in features measuring the activities of users such as the status count, followers count, out degree, and so on.

### 6.3 General Comments from Domain Experts

All domain experts were impressed by the overall FluxFlow design, mentioning that the visualization was intuitive and aesthetically pleasing and that the interactions and animations were smooth. They appreciated FluxFlow as a research tool for exploring and understanding the detected anomalous retweeting threads. The experts particularly liked the volume circle view and its interactive aggregation feature, commenting that "*You can see the overall trends and user [anomaly score] distributions and easily compare the threads, which is very helpful for identifying the tipping points.*" Moreover, they thought that showing the duplicated users and social interaction graphs on top of the thread timlines were critical as "*it shows how the same users or a group of related users acted in different threads*". The experts also mentioned it would be nicer to show the "chain" of retweeting in the threads, if such data is available.

The information context we provided for interpreting anomalous retweeting threads, such as the cluster view and MDS view, were also appreciated by the experts, who commented: "*I can find outliers from the MDS view that provides extra information complementary to anomaly scores. [...] The cluster view helps to organize the threads hierarchically and to browse them with similar content.*" One expert suggested it would be convenient to display an overview of threads when zooming the MDS view. Another said allowing the interactive selection of different features to form the MDS projection could be more powerful. In addition, the experts thought the features view and states view were useful for them to deeply inspect the data and the model. For example, one said "*it helps to illustrate what's the states underlying the model. [...] I see different groups of people in different states.*" For improvements, one expert mentioned that it would be better "if two threads' feature view can be put side-by-side" for low-level comparisons.

## 7 Discussion

There exist some limitations of the current prototype that we would like to address. First, the anomaly detection process of OCCRF could be long and tedious, due to large data scales. In our experiment, OCCRF can train the model from 3000 retweeting threads in around 2 hours, using 40 cores (Intel Xeon 2.13GHz) and 32GB memory on a Linux sever. While we could pre-compute the data, it lacks the flexibility needed for some tasks such as real-time monitoring. Second, since user interaction data is only partially available in Twitter, we estimated the social connection graph based on users' mentioning and retweeting behaviors from historical datasets, which assumes that the graph structure did not change. Third, despite that FluxFlow incorporates a number of visualizations to allow an in-depth comprehension of the anomaly detection model, such as the features view and states view, there are still many kinds of low-level information to show for helping analysts develop better algorithms, such as correlations of feature vectors and interactions of model parameters.

There are several interesting directions for generalizing and extending our current system. First, while FluxFlow is built with OCCRF, its visualization component can stand alone to serve a more general tool for visual exploration of information propagation on social media (Fig. 5). For example, the threads view is generalizable in timeline-based exploration of conversational threads. Moreover, the multi-scale representation and interaction in the cluster view is flexible enough to be applied in navigating any hierarchical data, such as phylogenetic trees in biology. The proposed circle packing algorithm used in the threads view is also applicable for object layout in other timeline visualizations. Furthermore, if the analytics computation can be improved for real-time processing, the thread timeline views can be extended to represent dynamic retweeting data with techniques such as CloudLines [29] and Visual Sedimentation [26]. Lastly, to further facilitate the exploration of retweeting threads, we can easily incorporate FluxFlow with interactive filtering techniques based on keywords or geo-locations such as that in SenseSpace2 [31].

## 8 Conclusion and Future Work

We have presented FluxFlow, a novel visual analytics system for the interactive exploration of anomalous information spreading on social media. FluxFlow systemically incorporates a set of algorithms to characterize retweeting threads, perform anomaly detection in those threads, and produce an effective visual interface, consisting of several novel visualizations and multiple coordinated views, to present the model outputs.

Through quantitative evaluation of the model and qualitative interviews with domain experts, study results indicated that FluxFlow's anomaly detection algorithm is efficient in identifying misinformation, and the visualization is useful for analysts to discover insights and comprehend the model.

In the future, we will further investigate anomaly detection models for Twitter conversational threads and improve the current algorithm to allow a faster analysis. In addition to emotional features, it is interesting to integrate other content features (e.g., topics and semantic information) to the current anomaly detection. However, it may significantly increase the dimensionality of the feature space. We plan to supplement FluxFlow with real-time monitoring abilities for anomalous information detection with live social media data streams, thus allowing people to make immediate actions and decisions. We will also develop visualizations to further reveal underlying mechanisms of complicated machine learning models, and extend the current thread timeline representations with other visualization techniques to accommodate live and dynamic data.

## REFERENCES


[1] http://www.telegraph.co.uk/news/uknews/law-and-order/10556041/Mark-Duggan-well-balanced-or-confrontational-the-29-year-old-whose-death-left-London-burning.html.

[2] http://hadoop.apache.org/.

[3] http://www.snopes.com/photos/natural/sandy.asp.

[4] http://www.snopes.com/politics/conspiracy/boston.asp.

[5] B. Adams, D. Phung, and S. Venkatesh. Eventscapes: visualizing events over time with emotive facets. In *Proc. of the ACM Int'l Conf. on Multimedia*, MM '11, pages 1477–1480, 2011.

[6] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Springer, 2011.

[7] M. Q. W. Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Proc. of Advanced Visual Interfaces*, pages 110–119, 2000.

[8] M. S. Bernstein, B. Suh, L. Hong, J. Chen, S. Kairam, and E. H. Chi. Eddi: interactive topic-based browsing of social status streams. In *Proc. of the ACM Sym. on user interface software and technology*, UIST '10, pages 303–312, 2010.

[9] P. J. F. G. Borg. *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005.

[10] S. Brain. Facebook statistics. http://www.statisticbrain.com/facebook-statistics/.

[11] C. Brewer and M. Harrower. Colorbrewer: Color advice for maps. http://colorbrewer2.org/.

[12] N. Cao, Y.-R. Lin, X. Sun, D. Lazer, S. Liu, and H. Qu. Whisper: Tracing the spatiotemporal process of information diffusion in real time. *IEEE Trans. on Visualization and Computer Graphics*, 18(12):2649–2658, 2012.

[13] J. Chae, D. Thom, H. Bosch, Y. Jang, R. Maciejewski, D. Ebert, and T. Ertl. Spatiotemporal social media analytics for abnormal event detection and examination using seasonal-trend decomposition. In *IEEE Conf. on Visual Analytics Science and Technology*, VAST'12, pages 143–152, 2012.

[14] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, 2009.

[15] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.

[16] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia. Detecting automation of twitter accounts: Are you a human, bot, or cyborg? *IEEE Trans. Dependable Sec. Comput.*, 9(6):811–824, 2012.

[17] W. Cui, S. Liu, L. Tan, C. Shi, Y. Song, Z. Gao, H. Qu, and X. Tong. Textflow: Towards better understanding of evolving topics in text. *IEEE Trans. on Visualization and Computer Graphics*, 17(12):2412–2421, 2011.

[18] N. Diakopoulos, M. Naaman, and F. Kivran-Swaine. Diamonds in the rough: Social media visual analytics for journalistic inquiry. In *IEEE Sym. on Visual Analytics Science and Technology*, pages 115–122, 2010.

[19] M. Dork, D. Gruen, C. Williamson, and S. Carpendale. A visual backchannel for large-scale events. *IEEE Trans. on Visualization and Computer Graphics*, 16(6):1129–1138, 2010.

[20] W. Dou, X. Wang, D. Skau, W. Ribarsky, and M. Zhou. Leadline: Interactive visual analysis of text data through event identification and exploration. In *IEEE Conf. on Visual Analytics Science and Technology*, VAST'12, pages 93–102, 2012.

[21] W. Dou, L. Yu, X. Wang, Z. Ma, and W. Ribarsky. Hierarchicaltopics: Visually exploring large text collections using topic hierarchies. *IEEE Trans. on Visualization and Computer Graphics*, 19(12):2002–2011, 2013.

[22] N. Elmqvist and J. Fekete. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *IEEE Trans. on Visualization and Computer Graphics*, 16(3):439–454, 2010.

[23] S. Havre, E. Hetzler, P. Whitney, and L. Nowell. Themeriver: Visualizing thematic changes in large document collections. *IEEE Trans. on Visualization and Computer Graphics*, 8:9–20, 2002.

[24] C.-T. Ho, C.-T. Li, and S.-D. Lin. Modeling and visualizing information propagation in a micro-blogging platform. In *Int'l Conf. on Advances in Social Networks Analysis and Mining*, pages 328–335, 2011.

[25] H. Huang, Z. Wen, D. Yu, H. Ji, Y. Sun, J. Han, and H. Li. Resolving entity morphs in censored data. In *Proc. the Annual Meeting of the Association for Computational Linguistics*, 2013.

[26] S. Huron, R. Vuillemot, and J.-D. Fekete. Visual sedimentation. *IEEE Trans. on Visualization and Computer Graphics*, 19(12):2446–2455, 2013.

[27] E. Katz and P. F. Lazarsfeld. *Personal influence: the part played by people in the flow of mass communications*. Free Press, 1955.

[28] D. Keim, G. Andrienko, J.-D. Fekete, C. Görg, J. Kohlhammer, and G. Melançon. Visual analytics: Definition, process, and challenges. In A. Kerren, J. Stasko, J.-D. Fekete, and C. North, editors, *Information Visualization: Human-Centered Issues and Perspectives*, chapter 7, pages 154–175. Springer Berlin Heidelberg, 2008.

[29] M. Krstajic, E. Bertini, and D. Keim. Cloudlines: Compact display of event episodes in multiple time-series. *IEEE Trans. on Visualization and Computer Graphics*, 17(12):2432–2439, 2011.

[30] D. Luo, J. Yang, M. Krstajic, W. Ribarsky, and D. Keim. Eventriver: Visually exploring text collections with temporal references. *IEEE Trans. on Visualization and Computer Graphics*, 18(1):93–105, 2012.

[31] A. MacEachren, A. Jaiswal, A. Robinson, S. Pezanowski, A. Savelyev, P. Mitra, X. Zhang, and J. Blanford. Senseplace2: Geotwitter analytics support for situational awareness. In *IEEE Conf. on Visual Analytics Science and Technology*, VAST'11, pages 181–190, 2011.

[32] A. Marcus, M. S. Bernstein, O. Badar, D. R. Karger, S. Madden, and R. C. Miller. Twitinfo: aggregating and visualizing microblogs for event exploration. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, CHI '11, pages 227–236, 2011.

[33] NYLabs. Cascade. http://nytlabs.com/projects/cascade.html, 2011.

[34] A. Pak and P. Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *Proc. of Int'l Conf. on Language Resources and Evaluation*, pages 1320–1326, 2010.

[35] R. Procter, F. Vis, and A. Voss. Riot rumours. http://www.theguardian.com/uk/interactive/2011/dec/07/london-riots-twitter/, 2010.

[36] D. M. Romero, B. Meeder, and J. Kleinberg. Differences in the mechanics of information diffusion across topics: Idioms, political hashtags, and complex contagion on twitter. In *Proc. of the Int'l Conf. on World Wide Web*, WWW '11, pages 695–704, 2011.

[37] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, 2001.

[38] T. Schreck and D. Keim. Visual analysis of social media data. *Computer*, 46(5):68–75, 2013.

[39] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proc. of IEEE Sym. on Visual Languages*, pages 336–343, 1996.

[40] Y. Song, Z. Wen, C.-Y. Lin, and R. Davis. One-class conditional random fields for sequential anomaly detection. In *Proc. of the Int'l Joint Conf. on Artificial Intelligence*, IJCAI'13, pages 1685–1691, 2013.

[41] M. Stefaner. Revisit. http://moritz.stefaner.eu/projects/revisit-twitter-visualization/, 2012.

[42] Y. R. Tausczik and J. W. Pennebaker. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of Language and Social Psychology*, 29(1):24–54, 2010.

[43] F. Viégas, M. Wattenberg, J. Hebert, G. Borggaard, A. Cichowlas, J. Feinberg, J. Orwant, and C. Wren. Google+ripples: A native visualization of information flow. In *Proc. of the Int'l Conf. on World Wide Web*, WWW '13, pages 1389–1398, 2013.

[44] W. Wang, H. Wang, G. Dai, and H. Wang. Visualization of large hierarchical data by circle packing. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, CHI '06, pages 517–520, 2006.

[45] S. Wu, J. M. Hofman, W. A. Mason, and D. J. Watts. Who says what to whom on twitter. In *Proc. of the Int'l Conf. on World Wide Web*, WWW '11, pages 705–714, 2011.

[46] P. Xu, Y. Wu, E. Wei, T.-Q. Peng, S. Liu, J. Zhu, and H. Qu. Visual analysis of topic competition on social media. *IEEE Trans. on Visualization and Computer Graphics*, 19(12):2012–2021, 2013.

[47] J. Yang and S. Counts. Predicting the speed, scale, and range of information diffusion in twitter. In *Proc. of the Int'l Conf. on Social Informatics*, pages 216–231, 2010.

[48] J. Zhao, F. Chevalier, C. Collins, and R. Balakrishnan. Facilitating discourse analysis with interactive visualization. *IEEE Trans. on Visualization and Computer Graphics*, 18(12):2639–2648, 2012.