From Anomaly Detection to Rumour Detection using Data Streams of Social Platforms

Nguyen Thanh Tam[1], Matthias Weidlich[2], Bolong Zheng[3], Hongzhi Yin[4],
Nguyen Quoc Viet Hung[5], Bela Stantic[5]

[1] École Polytechnique Fédérale de Lausanne, [2] Humboldt-Universität zu Berlin,
[3] Huazhong University of Science and Technology, [4] University of Queensland, [5] Griffith University

tam.nguyenthanh@epfl.ch, matthias.weidlich@hu-berlin.de, zblchris@gmail.com,
db.hongzhi@gmail.com, quocviethung.nguyen@griffith.edu.au (Corresponding author), b.stantic@griffith.edu.au

ABSTRACT

Social platforms became a major source of rumours. While rumours can have severe real-world implications, their detection is notoriously hard: Content on social platforms is short and lacks semantics; it spreads quickly through a dynamically evolving network; and without considering the context of content, it may be impossible to arrive at a truthful interpretation. Traditional approaches to rumour detection, however, exploit solely a single content modality, e.g., social media posts, which limits their detection accuracy. In this paper, we cope with the aforementioned challenges by means of a multi-modal approach to rumour detection that identifies anomalies in both, the entities (e.g., users, posts, and hashtags) of a social platform and their relations. Based on local anomalies, we show how to detect rumours at the network level, following a graph-based scan approach. In addition, we propose incremental methods, which enable us to detect rumours using streaming data of social platforms. We illustrate the effectiveness and efficiency of our approach with a real-world dataset of 4M tweets with more than 1000 rumours.

PVLDB Reference Format:
Nguyen Thanh Tam, Matthias Weidlich, Bolong Zheng, Hongzhi Yin, Nguyen Quoc Viet Hung, Bela Stantic. From Anomaly Detection to Rumour Detection using Data Streams of Social Platforms. *PVLDB*, 12(9): 1016-1029, 2019.
DOI: https://doi.org/10.14778/3329772.3329778

1. INTRODUCTION

Social platforms became widely popular as a means for users to share content and interact with other people. Due to their distributed and decentralised nature, content on social platforms is propagated without any type of moderation and may thus contain incorrect information. Wide and rapid propagation of such incorrect information quickly leads to *rumours* that may have a profound real-world impact. For instance, in April 2013, there was rumour about two explosions in the White House, injuring also Barrack Obama [55]. The rumour was fuelled by content posted using a hacked Twitter account associated with a major new agency. The resulting panic had major economic consequences, such as a $136.5 billion loss at the stock market. This incident highlights the need for early and accurate *rumour detection*, in particular on social platforms.

It is notoriously hard to detect rumours [47]. Posts on social platforms are short and lack semantics. For instance, tweets have a limited number of characters, and comprise slang and spelling mistakes. Hence, traditional techniques to assess the credibility of (long, well-written) documents are of limited use for social platforms. Also, user interactions at unprecedented scale lead to rumours spreading quickly. Earliness of rumour detection is as important as detection accuracy. Moreover, social platforms are dynamic. Content is posted continuously, so that rumour detection cannot exhaustively collect data before giving results, but needs to work with streaming data. Finally, posts on social platforms are contextual. A post in isolation may not provide sufficient information for rumour detection. Instead, modalities such as user backgrounds, hashtags, cross-references, and user interactions must be considered to improve detection accuracy.

Several debunking services such as snopes.com have been established to expose rumours and misinformation. They harness collaborative user efforts to identify potential rumours, which are then verified by experts. Due to such manual processing, the number of potential rumours that can be assessed is limited and significant time is needed for verification, which motivated work on automated rumour detection. Given the short length of posts on social platforms, rumour detection is often approached by grouping posts that relate to a single event [30]. This does not work in an online setting, though, since the posts related to an event are not available a priori.

Traditional rumour detection techniques tend to rely solely on the textual information of posts, potentially combined with features on post authors and their relations. However, focusing on one or two modalities of posts on social platforms is insufficient. For instance, users posting rumour-related content are often ignored by other users, which is not directly visible in features that capture solely the characteristics of a single user. In another example, posts circulating among a group of users that believe in conspiracy theories are likely to refer to rumours. Without information from outside the group, it is impossible to know whether these posts are related to a rumour.

Against this background, we argue for a novel approach to rumour detection that identifies anomalies on social platforms by comparing data *between peers* and *with the past*. Such anomalies can be observed for different modalities (e.g., users, tweets) and at varying levels of granularity. For example, a sudden increase or decrease in

Proceedings of the VLDB Endowment, Vol. 12, No. 9
ISSN 2150-8097.
DOI: https://doi.org/10.14778/3329772.3329778

the number of followers of a user may be related to the user spreading rumours. Also, within a group of users, the credibility of one user being significantly lower than their peers may stem from the propagation of rumours. Moreover, relations between entities (e.g., users, posts, hashtags, links) may hint at anomalies, e.g., differences in time and location mentioned in a tweet and in a linked article.

In this paper, we present models and methods to realise the idea of detecting rumours based on anomalies. To this end, we follow a data management approach: We ground rumour detection in algorithms that work on a generic graph representation of social data, thereby achieving a solution that is applicable for any type of social platform. We first show how to identify anomalies locally, by assessing entities and relations of a social platform in comparison to their peers and to their past. Yet, acknowledging the inherent randomness of social platforms, anomalies are then viewed at a broader scale. To conclude on the spread of rumours, which is deemed more important than their classification [47], we incorporate the vicinity of local anomalies.

Our contributions and the structure of the paper (following a discussion of some background in §2) are summarised as follows:

- *Social Platform Model and Rumour Detection (§3).* Based on a model for social platforms, we develop a general process to detect rumours based on local and global anomalies.
- *Local Anomaly Detection (§4).* We propose a non-parametric method for anomaly detection at the level of individual entities, based on differences between (i) current and past observations related to an entity, and (ii) the entity and its peers.
- *Global Anomaly Detection (§5).* We lift anomaly detection to groups of entities, taking into account relations between them.
- *Streaming Setting (§6).* We show how to apply our approach for streaming data by incrementally computing anomaly scores on the local and global level.

An evaluation of our approach with more than 4M real-world tweets, spanning more than 1000 rumours, is presented in §7. We review related work in §8 and conclude in §9.

2. BACKGROUND

Anomalies in social media. Abnormal propagation of information on social platforms can be classified as different types of anomalies, including hypes, fake news, satire news, disinformation, misinformation, and rumours [57]. For hypes, information is propagated in cascades that accidentally 'blow-up' on social platforms, e.g., related to popular events. Rumours, in turn, originate from the fact that people tend to exaggerate what they dislike [6]. Their veracity needs to be assessed, which is commonly done by assigning a trust score to entities, such as users and posts [2].

Here, we focus on detecting rumours. While hypes and rumours share some characteristics, they differ in how information is propagated. In hypes, information is spread randomly and chaotically. As revealed in a recent survey [47], however, rumours are propagated in a channelled manner, spreading 'farther, faster, and deeper' through interactions of actual users rather than bot accounts.

Type of anomalies differ in their sets of indicative signals. For example, detection of hypes (e.g., breaking news) focuses on peak volume of social posts and sharing activities [36, 37]. Spam detection of online reviews, in turn, uses user signals, such as average rating, number of reviews, and selectivity [51]. Our approach for rumour detection looks at inconsistency signals, exemplified below.

Twitter as an example. While we use Twitter as an example of a social platform throughout the paper, our model is applicable to other social platforms [40], as it is based on a universal graph representation (§3), generic statistical measures to compute anomalies (§4), and a graph-based anomaly detection algorithm (§5).

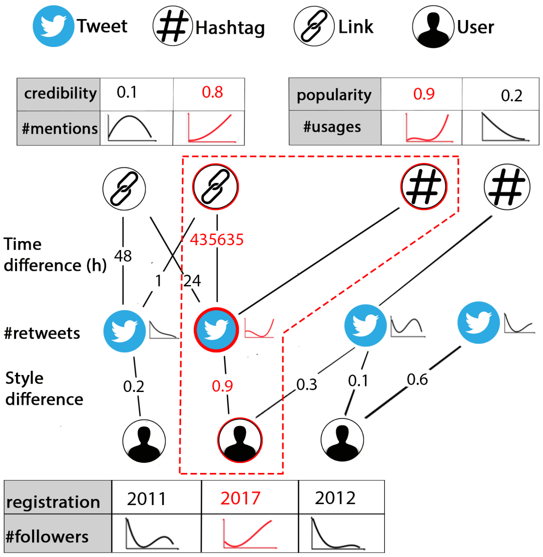


Figure 1: Multi-modal social graph

Consider a snapshot of Twitter social graph, as shown in Fig. 1. It includes users, tweets, hashtags, and linked articles. Each entity has different features, e.g., a user has a registration date and a number of followers. Entities are connected by relations. For instance, the relation between a tweet and an article indicates that the content of the tweet contains a link to that article. Moreover, each relation has an attribute value, e.g., the tweet-article relation has an attribute that indicates the difference between the publication dates of the tweet and the article, respectively.

Rumours are often manifested in anomalies related to entities and their relations. In Fig. 1, one may observe that the highlighted user has a registration date that is significantly newer than those of related users. At the same time, the number of followers is very high, compared to the historical record of the user. Other entities in this example are also suspicious, due to anomalies. For the highlighted tweet, the number of retweets is suddenly higher than in the past, as is the number of mentions for the highlighted linked article.

The above local anomalies provide a first signal for rumour detection. Yet, in isolation, these signals are not reliable. For instance, a user sparking a hype will also experience a sudden increase in the number of followers. We therefore need to consider *global anomalies* that comprise connected entities for which local anomalies have been observed. In the example, a rumour-related user is expected to post a rumour-related tweet, which links to a rumour-related article. Moreover, these connections between entities are also meaningful for rumour detection. For instance, in Fig. 1, the time difference between the highlighted tweet and linked article is suspicious, as is the difference between the regular linguistic style of this user (derived from past tweets) and the style of this particular tweet.

In this work, we provide the methods to realise the above idea: We exploit local anomalies and, based thereon, global anomalies among the entities of a social platform to reliably detect rumours.

3. MODEL AND APPROACH

Below, we present a model to capture entities of a social platform and their relations (§3.1). We then define the rumour detection problem (§3.2) and outline our approach to address it (§3.3).

3.1 A Model of Social Platforms

A social platform comprises many entities that are linked to each other by relations.

Entities (nodes). Our model comprises entities of specific types, i.e., modalities, such as tweets, links, users, and hashtags. Entities are modelled using feature vectors, where the features depend on the entity type. For the example in Fig. 1, each user has registration date and number of followers as features. While we limit the discussion to the above modalities in the remainder of this paper, our model is generic in the sense that further modalities such as images and videos [27] can be incorporated.

Relations (edges). Characteristics of entities in isolation are not sufficient to detect rumours. The relations between them provide a richer picture and thus can be expected to be beneficial for rumour detection. Each relation is also modelled by a feature vector, which is specific to the the type (or modality) of the relation. For the example in Fig. 1, each tweet-article relation has the time difference between the publication times of tweets and linked articles.

Multi-modal social graph. A *multi-modal social graph*, or *social graph*, is composed of modalities, entities, and relations between entities. We denote by $D = \{D_1, \ldots, D_n\}$ a set of entity types, while $V = V_1 \cup \ldots \cup V_n$ is a set of entities, such that V_i is the set of entities of type D_i. Similarly, $C \subseteq [D]^2 = \{C_1, \ldots, C_m\}$ is a set of relation types ($[D]^2$ being the 2-element subsets of D), $E = E_1 \cup \ldots \cup E_m$ are sets of relations, where E_i is the set of relations of type C_i.

Based thereon, a social graph is defined as $G = (Q, V, E, f)$, where $Q = D \cup C$ is called the set of modalities of G. The feature information f of entities and relations is used to capture rumour signals in a social graph. Formally, $f = \{f_1, \ldots, f_{n+m}\}$ is a set of mapping functions, where $f_i : Q_i \to \mathbb{R}^{q_i}$ defines an q_i-dimensional feature vector $f_i(x)$ for each element x of the modality Q_i.

The notion of a social graph enables us to address rumour detection with techniques for data management. As such, the developed algorithms are also applicable to data of social platforms that can be transformed to a graph representation [54, 45, 23, 42].

3.2 Rumour Detection

In a social graph, rumours materialise for a subset of its entities. The definition of this subset is not known, so that its identification is referred to as the rumour detection problem. That is, there is some (unknown) function that assigns truth values to entities (regular or rumourous), which shall be approximated.

Problem Statement *Given a social graph $G = (Q, V, E, f)$ and a ground-truth set $R^* \subseteq Q$, the* rumour detection problem *is to find a label function $l : Q \to \{1, 0\}$ to categorize which entities are rumourous, such that detection coefficient is maximized:*

$$\frac{|R^* \cap R|}{|R^* \cup R|} \quad \textit{with } R = \{x \in Q \mid l(x) = 1\}.$$

While the above definition is independent of the type of entity that is considered rumourous, in the remainder, we focus on the detection of rumourous tweets. The reason being that there is no clear-cut truth function to label other entities. For example, users may spread rumours in some tweets, but propagate regular information in others.

3.3 Approach Overview

Addressing the above problem requires us to overcome the tradeoff between accuracy and completeness, which is difficult [12]. A common strategy is to first focus on completeness and subsequently optimize the accuracy of rumour detection. Filtering out false positives is often easier than finding additional true positives.

Following this line, we first strive for completeness by collecting all rumourous signals in data features: The more anomalous a feature of a tweet, the more rumourous it is. However, such a feature-based approach alone will not yield high accuracy of rumour detection.

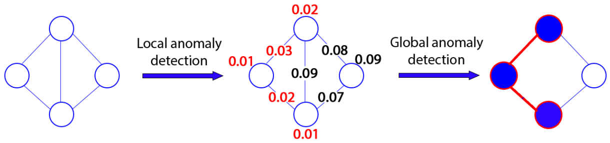


Figure 2: Rumour as Anomaly Detection Process

Since there is always randomness and noise in the data of a social platform, we conclude that a tweet is rumourous only if it is part of a rumourous graph structure. For example, in Fig. 1, the highlighted subgraph denotes such a structure for the respective tweet, capturing rumourous context related to a user, hashtag, and linked article.

Retrieving all rumour signals from a social graph, we then reduce false positives by cross-checking between the signals, while incorporating their contexts. More precisely, we use the structural information of a social graph (i.e. relations between entities) to find a subgraph that is most rumourous. The tweets contained in this subgraph are then considered to be the actual rumour.

Rationale. Our approach is driven by the following observations:

- Identifying solely individual rumourous tweets ignores the rumour structure, i.e., it neglects that a cluster of rumourous tweets denotes a single rumour. Hence, rumour detection shall incorporate the co-occurrence of rumourous tweets as part of a rumour.
- Identifying rumours solely on the level of tweets neglects the interplay of modalities in rumour propagation. A social graph defines complex relations between entities, so that the identification of rumourous tweets, e.g., leads to the identification of rumourous users, hashtags, and links. Hence, the structure of a social graph shall be exploited to assess the propagation of rumourous information. This way, the need to detect explicit events by aggregating entities is eliminated, which is a common first step in traditional rumour detection [38].

Framework. Against this background, we design a two-step rumour detection process, illustrated in Fig. 2. In a first step, we aim to detect local anomalies in entities and relations. In a second step, these local anomalies and the relations in the graph enable the detection of rumours at the subgraph level. Below, we summarise the two steps, while their details are given in §4 and §5, respectively.

Local anomaly detection. First, we design a function that assigns an anomaly score to each entity. We argue that an anomaly scoring shall satisfy the following requirements:

(R1) *Completeness:* In order to eliminate false negatives in rumour detection, the identification of anomalies in the data shall be comprehensive. That is, complementary angles to identify deviations from expected observations should be considered.

(R2) *Uniformity:* For entities of all modalities, there shall be a uniform scoring domain (independent of the number of features), with a uniform ordering (lower value indicating more rumourousness), and a uniform distribution (scores are uniformly distributed in $[0, 1]$). The latter is important as thresholding for rumour detection is challenging for non-uniform distributions.

(R3) *Non-parametric:* We assume that features follow an unknown baseline distribution. It is estimated based on the data and serves to assess the level of anomalousness per entity.

Global anomaly detection. Second, we rely on the detected local anomalies and aim at the detection of global anomalies, which indicate rumours. This shall incorporate the following requirements:

(R4) *Cross-checking:* In order to avoid false positives, rumourousness between neighbouring entities shall be cross-checked in the social graph. As content on social platforms is dynamic and rumours may propagate very quickly, a rumourous entity is expected to affect its neighbours immediately. Hence, global anomaly detection shall consider the context of local anomalies.

(R5) *Structuredness:* Any algorithmic solution to detect global anomalies shall acknowledge the structure of rumours. The 'rumour-related' parts of a social graph, in terms of rumourous information that jointly denotes a rumour, shall be detected.

(R6) *Non-parametric:* The scoring of a global anomaly shall not assume any prior distribution of local anomaly scores. This supports multi-modality and robustness to different datasets.

4. LOCAL ANOMALY DETECTION

This section is devoted to the computation of local anomaly scores in a social graph. Guided by the above requirements (R1, R2, R3), we first show how to construct features for identifying rumours (§4.1). Then, we introduce history-based anomaly scores (§4.2) and similarity-based anomaly scores (§4.3). Based thereon, a unified anomaly score is derived for each graph element (§4.4).

4.1 Features to Identify Rumours

Feature engineering is the only domain-specific step of our approach, which we illustrate here for the case of Twitter. We distinguish history-based and similarity-based features. The former capture differences between the current and past state of an entity. The latter help to cross-check the differences between entities and relations of the same type. Specifically, we consider the following features per modality, see also Table 1:

- User: The *registration age* and *credibility score* are considered indicators for rumours, since users spreading rumours tend to create new accounts to hide their identity. Moreover, sudden changes in the *frequency* of status updates, the number of *followers*, and the number of *#friends* may be related to rumours.
- Tweet: We consider *keywords* and the *linguistic style*. Tweets that are subjective or emotional are more likely to be rumour-related as they aim to provoke strong emotions to promote sharing. Also, the number of *retweets* may indicate rumours.
- Link: Articles linked in tweets may indicate rumours, which we assess based on the *credibility score* and *linguistic style* of the linked source and article, respectively. Furthermore, the number of *mentions* over time is used as a feature.
- Hashtag: The *popularity*, as measured by a semantic ranking [10], and sudden changes in the number of *usages* of a hashtag are expected to be rumour-related.

We further consider the features of relations between entities:

- Tweet-Link: The *time*, *location*, and *event* mentioned in a tweet may be different from the respective details given in the linked article. Also, the linguistic *style* of the tweet may be different from the one of the linked article.
- User-Tweet: The linguistic *style* of a tweet may differ from the regular style of the user.
- User-Link: The *source* linked in a tweet is anomalous.
- User-Hashtag: The hashtag is *novel*, i.e., it has not been used by the user before.
- Link-Hashtag: The hashtag has been *mentioned* in the linked article very frequently.

While some of the features are static (similarity-based), others are dynamic (history-based), so that they are derived from time snapshots using streaming APIs, such as [32]. We compute the features using established methods, whose details are described in §7.2.

Table 1: Features to identify local anomalies.

	Element	Feature	Anomaly Type
Entities	User	registration age	similarity-based
		credibility score	similarity-based
		status frequency	history-based
		#followers	history-based
		#friends	history-based
		#tweets	history-based
	Tweet	keywords	similarity-based
		linguistic style	similarity-based
		#retweet	history-based
	Link	credibility score	similarity-based
		linguistic style	similarity-based
		#mentions	history-based
	Hashtag	popularity score	similarity-based
		#usages	history-based
Relations	Tweet-Link	time	history-based
		location	similarity-based
		event	similarity-based
		style	similarity-based
	User-Tweet	linguistic style	similarity-based
	User-Link	source	similarity-based
	User-Hashtag	novelty	similarity-based
	Link-Hashtag	mentioning	similarity-based

Using the above features independently may lead to false positives. For instance, although rumours usually have a specific linguistic style, the reverse is not always true as, e.g., news about tragedies also adopt an emotional style. To mitigate such effects, we consider the above diverse set of features, which addresses requirement R1.

4.2 History-based Scoring

An anomaly score may be based on the differences between the current and past values of a feature vector. To this end, we establish a baseline distribution for each attribute to represents the normal behaviour, in the absence of any rumour. Then, based on the baseline distribution and the current feature values, we estimate an empirical p-value to measure the anomalousness of a feature. Aggregating these values, we asses the anomalousness of an entity or relation.

Deriving historic data. To derive historic values of features of entities or relations, we apply a temporal window. For an entity or relation x, the historic data is denoted by $X_t = \{x_1, \ldots, x_t\}$, where all x_i are temporal snapshots of x. This way, historic data of the same length is considered for different history-based features of x, which enables the integration of features with varying temporal properties. Yet, t is not fixed across entities or relations, so that historic data of different lengths may be incorporated for different modalities. Note that collecting historic data is straight-forward for common platforms. Details on our data collection can be found in §7.2.

Anomaly score of a history-based feature. Our computation is based on the following null hypothesis: If there is no rumour and we select a random observation from the past, how likely is it that its value is greater than or equal the current one? Based on historic data, the anomaly score of a feature $j \in [1, q_i]$ of an element (entity or relation) $x \in Q_i$ at timestamp t is defined as the statistical confidence degree (i.e., the p-value, the lower the better):

$$p_T(f_{i,j}(x_t)) = \frac{|\{x_r \in X_{t-1} : f_{i,j}(x_r) \geq f_{i,j}(x_t)\}|}{|X_{t-1}|} \tag{1}$$

where $f_{i,j}(x_t)$ refers to the j-th component of the feature vector $f_i(x_t)$ of an element x at timestamp t. In other words, the p-value is computed based on the number of past values $f_{i,j}(x_r)$ that are greater than the current observation $f_{i,j}(x_t)$. This is a *non-parametric* statistical measure (addressing requirement R3), since it does not assume any prior distribution on the historic data.

Example 1. Consider a Twitter user @jacobawohl (x), who is related to rumours about the Las Vegas shooting in 2017 [4]. The number of active followers (feature 1) and the number of tweets (feature 2) of the user at three consecutive time points is $\{4.72K, 294, 7.03K\}$ and $\{102, 43, 51\}$ respectively. At the third time point, the p-values of feature 1 and feature 2 are $p(f_1(x_3)) = \frac{0}{2} = 0$ and $p(f_2(x_3)) = \frac{1}{2} = 0.5$. At the second time point, these values are $p(f_1(x_2)) = \frac{1}{1} = 1$ and $p(f_2(x_2)) = \frac{1}{1} = 1$. Moreover, at the first time point, there is no historic value and we set $p(f_1(x_1)) = p(f_2(x_1)) = 1$.

History-based anomaly score. The non-parametric p-value of an entity or relation x specifies its anomaly score based on historic observations. We aggregate these anomaly scores as follows:

$$p_T(x_t) = \frac{|\{x_r \in X_{t-1} : p_{min}(x_r) \leq p_{min}(x_t)\}|}{|X_{t-1}|} \tag{2}$$

where $p_{min}(x_r) = min_{j=1\dots q_i} p(f_{i,j}(x_r))$. That is, at each timestamp, we compute the minimum value over all features. Then, the anomaly score $p_T(x_t)$ is the number of past minimum feature values $p_{min}(x_r)$ that are less than the current minimum feature value $p_{min}(x_t)$.

The reason for using *min* for the aggregation is to avoid false negatives, where some features are anomaly-significant, whereas others are not. Moreover, we do not consider the minimum p-value over all features at a single timestamp directly, since elements can have different numbers of features. Rather, our idea is to cross-check the scores between different timestamps across features, so that our aggregation yields *uniform* scores over all entities and relations, regardless of their modality, which addresses requirement R2.

Example 2. Taking up Example 1, we derive that $p_{min}(x_1) = \min\{p(f_1(x_1)), p(f_2(x_1))\} = 1$ as well as $p_{min}(x_2) = \min\{p(f_1(x_2)), p(f_2(x_2))\} = 1$, and $p_{min}(x_3) = \min\{p(f_1(x_3)), p(f_2(x_3))\} = 0$. The p-value of user x at the current timestamp is $p(x_3) = (0+0)/2 = 0$. With a confidence level of 99%, we say that the user is involved in some rumour, since $p(x_3) \leq 0.01$.

4.3 Similarity-based Scoring

Anomalousness can also be quantified by differences between entities and relations of the same type. For instance, the linguistic style of a tweet is a static property, that often lacks historic data, but may be a strong indicator of rumours. We therefore establish a baseline for features of static properties, as detailed below.

Anomaly score of a similarity-based feature. The null hypothesis of this case is summarised as: If there is no rumour, how likely does a randomly selected set of observations for a feature of different elements (entities or relations) of the same modality would have values greater than the considered element. We capture the null distribution of a feature of an element x of modality Q_i using the feature values of its peers ($x' \in Q_i$). Then, the p-value of a similarity-based feature $j = 1 \dots q_i$ of an element x is defined as follows:

$$p_S(f_{i,j}(x)) = \frac{|x' \in Q_i : f_{i,j}(x') \geq f_{i,j}(x)|}{|Q_i|} \tag{3}$$

That is, the p-value is computed based on the number of values $f_{i,j}(x')$ from other elements of the same modality that are greater than the value of the current element, $f_{i,j}(x)$. This p-value is also non-parametric (as defined by requirement R3), since it does not assume any prior distribution on the elements.

Example 3. Now, consider three Twitter users @prisonplanet (x), @wes_chu (y), @jacobawohl (z), who have registration ages (feature 1) of {8, 6, 1} and average credibility scores (feature 2) of {-5, -4, -3} (0 means least credible). For feature 1, we have $p(f_1(x)) = 1$, $p(f_1(y)) = 2/3$, $p(f_1(z)) = 1/3$. For feature 2, we have $p(f_2(x)) = 1$, $p(f_2(y)) = 2/3$, $p(f_2(x)) = 1/3$.

Similarity-based anomaly score. Again, based on the p-value of a similarity-based feature of an element x, the similarity-based anomaly score of x is defined as follows:

$$p_S(x \in Q_i) = \frac{|x' \in Q_i : p_{min}(x') \leq p_{min}(x)|}{|Q_i|} \tag{4}$$

where $p_{min}(x') = min_{j=1\dots q_i} p_S(f_{i,j}(x'))$. For each element, we compute the minimum value over all features. Then, the anomaly score of an element is the number of elements such that the minimum feature value of the current element is larger than their minimum feature values. As above, we choose *min* as an aggregation function to avoid outliers. We also aggregate across elements rather than features of a single element only. This yields *uniform* anomaly scores of elements from different modalities (requirement R2).

Example 4. We continue with Example 3 and derive $p_{min}(x) = \min\{p(f_1(x)), p(f_2(x))\} = 1$, $p_{min}(y) = \min\{p(f_1(y)), p(f_2(y))\} = 2/3$, and $p_{min}(z) = \min\{p(f_1(z)), p(f_2(z))\} = 1/3$. The p-value of z is $p(z) = (0+0+1)/3 = 0.33$. With a confidence level of 65%, we say that user z is involved in some rumour, since $p(z) \leq 0.35$.

4.4 Unified Scoring

As both entities and relations show history-based and similarity-based features, we combine the respective anomaly scores:

$$p(x) = \min\{p_T(x), p_S(x)\} \tag{5}$$

where $p_T(x) = 1$, if x has no history-based features, and $p_S(x) = 1$, if x has no similarity-based features. Again, *min* is used in the aggregation to avoid outliers.

We note that $p_T(.)$ and $p_S(.)$ are uniformly distributed in $[0,1]$ under the assumption that, in the absence of rumours, (i) the current observations are interchangeable with observations in the past; and (ii) the current observations of an element are interchangeable with observations from other elements. Based thereon, the probability that $f_{i,j}(x_r) \geq f_{i,j}(x)$ and $f_{i,j}(x') \geq f_{i,j}(x)$ is 0.5, which makes $p_T(f_{i,j}(x))$ and $p_S(f_{i,j}(x))$ follow a uniform distribution in $[0,1]$. Also, the minimum of p-values from different features are interchangeable with past minimum values or from other peers, so that $p_T(x)$ and $p_S(x)$ are uniformly distributed in $[0,1]$.

The *uniform* distribution of p-values is important: It enables us to handle the heterogeneity of a social graph, as different elements and modalities are mapped to the same domain of p-values. Moreover, the model facilitates the integration of multiple features for a single user, tweet, link, or hashtag, without a priori knowledge on the importance of feature for rumour detection. Finally, the overall p-value is non-parametric, since it does not assume any prior distribution, but integrates any correlation of p-values of different features.

5. GLOBAL ANOMALY DETECTION

Guided by the requirements for global anomaly detection (R4, R5, R6), we introduce the notion of an anomaly graph (§5.1), before turning to the computation of the anomalousness of a subgraph (§5.2), and the detection of a most anomalous subgraph (§5.3).

5.1 Anomaly Graph

Rumour detection using solely local information is not reliable. Local anomalies may be outliers (false positives), as features on social platforms are often noisy [32] and there are no clear-cut thresholds to filter false positives. Hence, rumour detection shall incorporate information from several elements (entities and relations) of a social graph, each providing a different view on a rumour and, thus, potentially reinforcing each other. A global view is further valuable to differentiate between anomalies that stem from the random nature of social platforms from those that originate from rumours. Finally, the propagation of rumourous information in a social graph helps to understand the rumour structure.

Formally, using the local anomaly detection, each element (entity or relation) in a social graph is associated with a p-value of being rumour-related. Given a social graph $G = (Q, V, E, f)$, this yields an anomaly graph $A = (Q, V, E, p)$, where $p : Q \rightarrow [0, 1]$ is a mapping that assign anomaly scores to entities or relations. This anomaly graph is the starting point for the identification of global anomalies, which materialise as subgraphs of the anomaly graph.

5.2 Anomalousness of a Subgraph

Rumour structure. Given an anomaly graph $A = (Q, V, E, p)$, a rumour structure is a subgraph of A that is *induced* and *connected*, which are standard graph properties [18]. Connectedness is required to cross-check anomaly scores between different elements. The subgraph shall be induced as we shall consider all relations between connected entities as a whole to eliminate false positives.

The anomalousness of a rumor structure is assessed based on:

- *Direct connections*, i.e., the relations (edges) of the graph. While both entities and relations are assigned anomaly scores, we need to conclude on the anomalousness of entities only (e.g., a tweet may be rumourous, while it is not meaningful to consider a tweet-link relation as rumourous). Hence, anomaly scores of a relation and its endpoints need to be unified.
- *Indirect connections* hold between entities that are connected by a path (of length larger than one) in the graph. The longer the path, the smaller the effect of the entities on each other, though.

Anomaly Hypergraph. To incorporate the above aspects, we propose to transform the anomaly graph to an anomaly hypergraph. The idea is to replace every two entities and the relation between them by a hypernode, which represents the collective information on the entities and the relation, while also providing an aggregated view on their anomaly scores. The hypernode inherits all further relations of the two original entities, i.e., it is connected to all entities to which the original entities had been connected. Formally, given two entities $v_1, v_2 \in V$ and a relation $e = \{v_1, v_2\} \in E$ of an anomaly graph $A = (Q, V, E, p)$, we define the respective hypernode as $v_H = \{v_1, v_2, e\}$ with an anomaly score:

$$p_H(v_H) = \max\{p(v_1), p(v_2), p(e)\} \tag{6}$$

Since $p(.)$ is uniformly distributed in $[0, 1]$, $p_H(.)$ also follows a uniform distribution in $[0, 1]$. Here, using *max* for aggregation reduces the chance of false positives, following requirement R4.

Processing all pairs of entities that are connected by a relation in the anomaly graph $A = (Q, V, E, p)$ as detailed above yields an anomaly hypergraph $H = (Q_H, V_H, E_H, p_H)$, with $Q_H \subset [Q]^2$ being a set of modalities, V_H being a set of hypernodes, $E_H \subseteq [V_H]^2$ being a set of edges, and p_H being a mapping function that assigns a anomaly score to each hypernode. Fig. 3 illustrates this construction.

Anomalousness measurement. Using the hypergraph H, we strive for a *connected* subgraph S that shows the highest level of anomaly. Since the hypernodes already include the original relations, it is straightforward to revert a subset of connected hypernodes to an induced connected subgraph of the original anomaly graph.

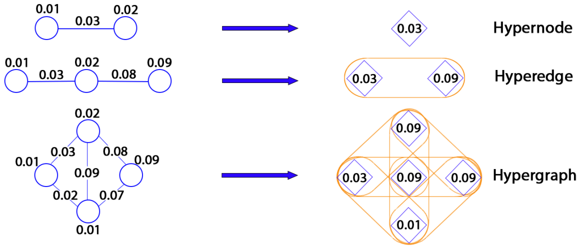


Figure 3: Hypergraph construction

To this end, we first measure the anomalousness of a subgraph, acknowledging the structure of rumours, see requirement R5. We employ the idea of scan statistics [26], which computes the statistical significance of a subgraph S being anomalous without assuming any prior distribution of the subgraph:

$$P(S) = \max_{0<\alpha\leq\alpha_{max}} \phi(\alpha, |V_\alpha(S)|, |V(S)|) \tag{7}$$

where α_{max} is the maximum statistical significance level ($\alpha_{max} = 0.05$ indicates that the value is *at least* 95% statistical significant), $V(S)$ is the node set of S, $V_\alpha(S) = \{v \in V(S) : p_H(v) \leq \alpha\}$ is the set of nodes in S with anomaly scores that are significant at the confidence level $\alpha > 0$.

To maximize the detection coefficient (see §3.2), function $\phi(.)$ shall favour the propagation of rumours, meaning that 'insignificant' nodes ($V(S) \setminus V_\alpha(S)$) are also accepted as long as they are connected with enough 'significant' entities ($V_\alpha(S)$). This is motivated by the dynamic nature of a rumour: Anomaly scores of rumourous entities vary over time and may not be significant at the same time. Moreover, function $\phi(.)$ shall be *non-parametric* (requirement R6), i.e., a function that compares the observed number of α-significant p-values $|V_\alpha(S)|$ to the expected number of α-significant p-values $\mathbb{E}[|V_\alpha(S)|$. Since our p-values are uniformly distributed in $[0, 1]$, we have $\mathbb{E}[|V_\alpha(S)|] = \alpha|V(S)|$. Therefore, we can directly compare $|V(S)|$ and $|V_\alpha(S)|$ as follows [11]:

$$\phi(\alpha, |V_\alpha(S)|, |V(S)|) = |V(S)| \times KL(\frac{|V_\alpha(S)|}{|V(S)|}, \alpha) \tag{8}$$

where KL is the Kullback-Leibler divergence defined as $KL(x, y) = x\log(x/y) + (1-x)\log(\frac{(1-x)}{(1-y)})$. Since $KL(x, y) \geq 0$, it follows that $P(S) \geq 0$ (the higher, the more anomalous). Based thereon, our goal is to detect subgraphs as large as possible (via $|V(S)|$), that have a high confidence level of anomalousness (via $|V_\alpha(S)|/|V(S)|$).

Example 5. Consider a subgraph S with nodes $V(S) = \{v_1 = 0.02, v_2 = 0.03\}$ and $\alpha_{max} = 0.05$. We have $|V(S)| = 2$. With $\alpha = 0.05$, we have $|V_{0.05}(S)| = 2$ and $\phi(0.05, 2, 2) = 2 \times (1\log(1/0.05) + 0\log(0/0.95)) = 2.6$. With $\alpha = 0.02$, we have $\phi(0.02, 1, 2) = 1.1$. With $\alpha = 0.03$, $\phi(0.03, 2, 2) = 3.0$. Therefore, we say that with *at least* 95% statistical significance ($\alpha_{max} = 0.05$), we are confident that the anomalousness of $S4$ is $P(S) = \max\{2.6, 1.1, 3.0\} = 3.0$.

5.3 Detection of a Most Anomalous Subgraph

Detecting a rumour structure in an anomaly graph $A = (Q, V, E, p)$ is equivalent to finding a connected subgraph with maximal anomalousness in the anomaly hypergraph $H = (Q_H, V_H, E_H, p_H)$:

$$\arg\max_{S\in\mathcal{S}(H)} P(S) \tag{9}$$

where $\mathcal{S}(H)$ contains all possible connected subgraphs of H.

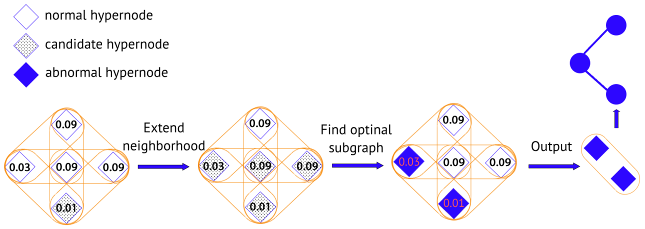


Figure 4: Illustration of Alg. 1

Proposition 1 *Solving Eq. 9 is NP-hard.*

PROOF. With a given α, we can construct a weight function on the node set as $w(v) = 1$ if $p(v) \leq \alpha$ and $w(v) = 0$ otherwise [16]. It is known that $\phi(.)$ is monotonically increasing w.r.t. $|V_\alpha(.)|$ [11]. Thus, $\phi(.)$ is monotonically increasing w.r.t. $\sum_{v \in S} w(v)$. Solving Eq. 9 is now equivalent to finding a solution to the maximum weighted subgraph problem, which is known to be NP-hard [9]. □

As the above problem is computationally expensive, we develop an approximation solution that scales to real-world social graphs. In the context of online social platforms, we argue that such a detection algorithm needs to satisfy two additional requirements:

- *Extensibility.* In practice, multiple rumours may occur at the same time. Hence, we consider a threshold as a relaxation parameter. We then aim at detecting all subgraphs in the anomaly graph that have an anomalousness value above this threshold. Such a threshold may be set based on rumours detected and verified in the past.
- *Incremental processing:* To cope with continuous data generated by social platforms, detection shall be incremental, incorporating new data as it arrives.

An Extensible and Incremental Algorithm. Due to the inherent complexity of Eq. 9, we present an approach to approximate a solution, see Alg. 1. It takes as input an anomaly graph and a detection threshold, and returns a sorted list of the most anomalous subgraphs that satisfy the threshold. The solution to Eq. 9 is simply the top-1 in the list. Moreover, in the light of the rumour detection problem (§3.2), only the tweet nodes of the output graph may be considered. Since multiple rumours may spread simultaneously on social platforms, however, we include a coverage level K as an input parameter, to cover rumours with smaller anomalousness values.

Our algorithm first expands the subgraphs from a seed node to their neighbours, before greedily optimising the anomaly score for the subgraphs. Specifically, we construct a hypergraph H (line 1), in which each hypernode has an anomaly score, as detailed above. We sort the hypernodes by these scores as this later improves the runtime of the scan statistics subproblem. We then select a root node (line 6), determine its neighbourhood (line 8), and find the subgraph in this neighbourhood with the highest anomaly score (line 9) using Alg. 2 (extended from [35]). The latter greedily retains nodes in the increasing order of p-values (the smaller, the better). Then, we continue to expand the subgraph until our root node set is equal to the most anomalous node set (line 10), i.e., it cannot be expanded further to increase the anomaly score. This guarantees that the subgraph is connected and its anomaly score is maximal.

Fig. 4 illustrates the core step of extending the neighbourhood of a root node and finding the optimal subgraph in Alg. 1 (line 6- 10).

Proposition 2 *The output of Alg. 1 is a sorted list of subgraphs in the decreasing order of anomaly level.*

Algorithm 1: Anomalous Subgraphs Detection

```
input  : An anomaly graph $A = (Q, V, E, p)$,
         a retain threshold $\tau$ (for streaming version),
         a coverage level of anomaly $K$ (default = 5),
         a specified number of hops $Z$ (default = $log(|V|)$)
output : A sorted list of subgraphs $\mathbb{S}$
1  Construct anomaly hypergraph $H = (Q_H, V_H, E_H, p_H)$ from $A$;
2  Sort the nodes in $H$ by anomaly score;
3  $\alpha_{max} = 0.05, \mathbb{S} = \mathbb{C} = \emptyset$;
4  for $q \in [1, \ldots, |Q_H|]$ do
5      for $k \in [1, \ldots, K]$ do
6          $R = \{v_k\}$, $v_k$ is the $k$-th most anomalous node in $V_H$ of modality $q$ ;
7          for $z \in \{1, ..., Z\}$ do
8              $H' = \{v \in V_H \setminus R : \exists v' \in R, \{v, v'\} \in E_H\}$;
9              $\langle S, P(S) \rangle = bestNeighbourhood(H', R, \alpha_{max})$ ;
10             if $S \setminus R \neq \emptyset$ then $R = S$ ;
11             else break;
12         $\mathbb{S} = \mathbb{S} \cup \{R\}$;
13 for $S \in \mathbb{S}$ do
14     if $P(S) \geq \tau$ then $\mathbb{C} = \mathbb{C} \cup \{S\}$ ;   // candidate rumours
15 return $\mathbb{S}$;
```

Algorithm 2: Optimal subgraph in the neighbourhood

```
input  : An anomaly hypergraph $H$, a root set $R$, a threshold $\alpha_{max}$
output : The most anomalous subset $S^*$ and its score $P(S^*)$
1  $W = \{p(v) : v \in S\} \cup \{\alpha_{max}\}$;
2  $S^* = \emptyset; P(S^*) = 0$;
3  for $\alpha \in W$ do
4      $S = \emptyset; S^*_\alpha = \emptyset; P(S^*_\alpha) = 0$;
5      for $v \in sorted(V(H) \cup R)$ do
6          $S = S \cup \{v\}$;
7          $P(S) = \phi(\alpha, |V_\alpha(S)|, |V(S)|)$;
8          if $P(S) > P(S^*_\alpha)$ and $R \subseteq S$ then
9              $S^*_\alpha = S$;
10             $P(S^*_\alpha) = P(S)$;
11     if $P(S^*_\alpha) > P(S^*)$ then
12         $S^* = S^*_\alpha$;
13         $P(S^*) = P(S^*_\alpha)$;
14 return $\langle S^*, P(S^*) \rangle$ ;
```

PROOF. Alg. 1 processes the nodes in increasing order of p-values (line 6). Since $\phi(.)$ is monotonically increasing w.r.t. $|V_\alpha(.)|$ and monotonically decreasing w.r.t. α and $|V(.)|$ [11], a detected subgraph has smaller anomalousness value than its predecessor. □

6. THE STREAMING SETTING

We now lift our approach to a streaming setting. We first discuss how local anomaly scores of a social graph can be computed incrementally (§6.1), before turning to the incremental computation of anomalous subgraphs detection (§6.2).

6.1 Incremental Anomaly Computation

Recall that computing local anomaly scores is based on historical data. However, in a streaming setting only a window w of data is available, and current observations continuously become historic observations; i.e. $X_{t+|w|} \leftarrow X_t \cup w$. To avoid continuous re-computation of anomaly scores, we propose a heuristic that estimates the score, but works incrementally. Below, we discuss this heuristic for history-based anomaly scores. However, the same approach can also be followed for similarity-based scores.

Intuitively, our approach avoids evaluating Eq. 1 and Eq. 2 whenever new data arrives. To this end, we approximate Eq. 1 with an incremental approach, as long as the respective feature is expected to have no effect on the anomaly score computation. In addition, we discuss how Eq. 2 can be evaluated efficiently.

Feature-level. To approximate Eq. 1, we assume that the historical data of a feature of an element x (entity or relation), i.e. $f_{i,j}(X_{T-1}) = \{f_{i,j}(x_t) : x_t \in X_{T-1}\}$ where T is the current timestamp, follows a normal distribution. Note that we consider this assumption solely in the streaming setting, as it yields runtime improvements by not using historic data. In practice, the anomaly scores can be justified by periodic updates from historic data. This distribution, denoted by $N_{j,x}(\mu,\sigma)$, is induced by the empirical mean μ and standard deviation σ computed from historic data. The empirical mean μ and standard deviation σ are updated incrementally as new data arrives:

$$\mu_{t+1} = \frac{\mu_t \times t + x_{t+1}}{t+1}; \quad \mu'_{t+1} = \frac{\mu_t \times t + x_{t+1}^2}{t+1}; \quad \sigma_{t+1} = \sqrt{\mu'_{t+1} - \mu_{t+1}^2}$$

as derived from $\mu = \mathbb{E}[X]$ and $\sigma = \sqrt{\mathbb{E}[X^2] - \mathbb{E}^2[X]}$.

Under the above assumption, Eq. 1 is approximated using μ and σ. Eq. 1 essentially counts the number of past values $f_{i,j}(X_{T-1})$ that are greater than the current observation $f_{i,j}(x_T)$. Given a new observation $f_{i,j}(x_{T+1})$ and the historical data captured by $N_{j,x}(\mu,\sigma)$, we derive the percentile of $f_{i,j}(x)$. This percentile is an approximation of how many past observations are greater than the current one. To compute the percentile, we convert $f_{i,j}(x)$ to a z-critical value:

$$z_{i,j}(x) = \frac{f_{i,j}(x) - \mu}{\sigma}$$

Based thereon, the percentile is computed as follows:

$$P(Z \geq z) = \int_{z}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}} dx$$

The percentile value provides us with approximation of the p-value of a specific feature: $p_j(x = z) = P(Z \geq z)$.

The above approximation is used to determine when Eq. 1 shall be evaluated from scratch. To this end, we exploit that $p_j(x)$ is used to calculate $p_{min}(x_t) = \min_{j=1...q_i} p(f_{i,j}(x_t))$, while $p_{min}(x_{T+1})$ is compared with other $\{p_{min}(x_t)\}_{t=1...T}$ in Eq. 2. Thus, $p_j(x)$ has an effect on the anomaly score of entity x only if it is smaller than the smallest value $p_{min}(x_t)$. That is, if $\hat{p}_j(x) < \min\{p_{min}(x_t)\}_{t=1...T}$, we do not need to re-evaluate Eq. 1. We later demonstrate experimentally that this heuristic helps to reduce the runtime significantly. However, the heuristic requires us to maintain $\min\{p_{min}(x_t)\}_{t=1...T}$, which is done as part of the computation on the entity-level.

Entity-level. When new data arrives, many terms of Eq. 2 remain unchanged, such as the anomaly score of a feature of an element in the past, $p_{min}(x_t)$. The only term that needs re-computation is the anomaly score of features at the current timestamp, $p_{min}(x_{T+1})$. Therefore, to evaluate Eq. 2 efficiently, we maintain all values $p_{min}(x_t)$. Given the requirement of maintaining $\min\{p_{min}(x_t)\}_{t=1...T}$, these values are kept in a sorted list. Evaluating Eq. 2 then becomes counting the number of values stored in the list before $p_{min}(x_{T+1})$.

6.2 Incremental Subgraph Detection

To handle streaming data in the computation of anomalous subgraphs, we realise the following idea: Upon the arrival of new data, the anomaly hypergraph will contain new nodes. For these nodes, we identify whether they are rumour-related due to being connected to existing anomalous subgraphs or inducing a new such subgraph. To this end, we associate nodes which belong to an anomalous subgraph with an identifier of the root node used for expansion (nodes may have several such identifiers). This way, upon adding a node, we immediately identify the subgraphs that it may be related to. These subgraphs can be rumour-related ($\mathbb{S}$ in Alg. 1) or potentially-anomalous ($\mathbb{C}$ in Alg. 1), which we distinguish as follows:

In case the new node connects to a rumour-related subgraph, the node is assessed based on a property of Alg. 2. Recall that in Alg. 1, we detect anomalous connected subgraphs by expanding subgraphs from root nodes using their neighbours. For each candidate set, we strive for the maximal connected subgraph (Alg. 2). The algorithm relies on a list of nodes, sorted by their p-values. When a new node arrives, we identify the related anomalous subgraphs (if any) and add the new node to the sorted list. If the p-value of the new node is higher than the value of any other node in the subgraph, the new node is rumour-related and added to the subgraph. If a node can be added to several rumour-related subgraphs, the subgraph with the highest anomalousness value is chosen.

In the case that the new node connects to a potentially-anomalous subgraph, Alg. 2 is re-run to identify whether the addition of the node yields a new anomalous subgraph.

7. EMPIRICAL EVALUATION

We evaluated our approach with a large real-world dataset obtained from Twitter. Below, we introduce our experimental setting (§7.1), data collection methodology (§7.2), and report characteristics of our data (§7.3). We show that our approach outperforms baseline methods for rumour detection in terms of effectiveness (§7.4) and explore the design choices of our model (§7.5). Next, we evaluate the scalability of our methods, including their use in a streaming setting (§7.6). Finally, we present an illustrative case study (§7.7).

7.1 Experimental Setting

Metrics. We use the following evaluation metrics:

- The detection *coefficient*, first proposed in [41], can be seen as a combination of precision and recall applied to a graph setting. R^* is defined as the set of rumour-related entities, whereas R is the set of entities labelled by a rumour detection technique. Then, the measure is defined as:

$$\textit{Coefficient} = \frac{|R^* \cap R|}{|R^* \cup R|}$$

- The *run-time* of processing a set of tweets.
- The *lag time to detection*, which is the time difference the first occurrence of a rumour (i.e., the first rumour-related entity) and its detection (i.e., a first entity is labelled accordingly).

Baselines. State-of-the-art rumour detection [57] is not applicable in our context, as it aims at learning a classification model based on a collection of entities that have been labelled with rumours. Such a collection is typically extracted by a pre-processing step that crawls the data related to a particular event, thereby assuming that the extracted elements can be labelled accordingly. As a result, the performance of these approaches strongly depends on the accuracy of such pre-processing [14, 29]. In our work, we progressively detect rumour-related entities by scanning abnormal signals (entities with high anomaly scores) in the social graph.

This fundamental difference in the taken approach is also reflected in the employed evaluation measures. Existing rumour detection techniques are evaluated using machine learning metrics, applied per rumour. This is not possible for our approach, so that we rely on the detection coefficient, applied per graph entity. In a broad sense, most rumour detection techniques focus on maximizing accuracy, instead of striving for a balance of accuracy and completeness.

Against this background, we consider several baseline methods. We implemented these methods based on the respective papers.

- *Decision* [13]: A decision tree classifier that is based on the Twitter information credibility model. The decision tree is constructed based on several hand-crafted features.

- *Nonlinear* [50]: An SVM-based approach that uses a set of hand-crafted features, selected for the tweets to classify.
- *Rank* [55]: A rank-based classifier that aims to identify rumours based on enquiry tweets.

In addition, we also compare our approach with methods based on homogeneous graphs that contain only a single modality. For instance, a tweet graph contains only tweets, while edges between tweets represent that tweets stem from the same user, have retweet relations, or share a keyword. We constructed four such homogeneous graphs, for users, tweets, links, and hashtags, respectively.

Parameters. We set the statistical significance level $\alpha_{max} = 0.05$ (i.e. the result is guaranteed to be *at least* 95% confidence). The coverage level K in Alg. 1 has been varied, so that we can detect multiple rumours at the same time.

For the static version of our approach, our rumour detection algorithm is executed multiple times by gradually extending the historical data $X_t = \{x_1, \ldots, x_t\}$ from the first day ($t = 1$) to the last day of each dataset. At each extension, all tweets in detected rumours will be removed to avoid that some rumours in the future will have smaller anomaly scores than the past (and thus the p-values might not be high enough with 95% confidence threshold).

For the incremental version, we set the window size $|w|$ to 12 hours; i.e. the historical data is defined by $X_t = \{x_{t-|w|}, \ldots, x_t\}$. Again, all tweets in detected rumours are removed. Note that, however, we cannot remove other types of entities (users, hashtags) since they potentially participate in different rumours. The threshold τ to retain the candidate rumours is set by the 20-quantiles of the anomalousness values of returned subgraphs.

Experimental environment. All results have been obtained on an Intel Core i7 system (2.8 Ghz, 32GB RAM).

7.2 Data Collection

Rumour collection. Snopes is a world-leading rumour-debunking service. Unlike other organizations such as Politifact and Urbanlegends, it is considered to be objective when evaluating the veracity of rumours [3, 46]. Snopes editors investigate each rumour along different dimensions and provide an argumentative report as shown in Table 2. For example, the claim describes the rumour succinctly and the rating represents its truth value according to the fact-checker.

Table 2: Information about a rumour.

Attribute	Example
id	trump-aid-puerto-rico
date	10/2/2017
genesis tweet	[..] President Trump has dispatched 140 helicopters [..]
sources of veracity	press reports, local officials, organizations
rating	MIXTURE [5]

Multi-model social graph construction. Twitter is a large social platform with tweets covering various domains such as politics and crime. It is frequently used by users to express their opinions in a timely manner, e.g., by retweeting others, which provides insights into how rumours propagate. These characteristics make Twitter data particularly suitable for evaluating rumour detection methods.

We followed the dataset construction process described in [28]. For each rumour, we identify its fingerprint, which is a set of keywords. Then, we use these keywords to search for tweets that are related to this rumour using Spinn3r [1]. We take the ID of a Snopes article as the starting point to create the fingerprint of a rumour. If the ID is not unique or too general, keywords are manually selected from the rumour's claim and the respective Snopes article. Applying modifications to these keywords provided us with a set of search queries to identify rumourous tweets. Since the queries may not identify all tweets that are rumour-related, we also considered retweets. To obtain negative samples, we collected further tweets from the timelines of users that authored rumourous tweets and of other users identified by retweets of regular tweets.

At this point, the social graph contains two entity types (tweets and users) and one relation type (user-tweet). The remaining entity and relation types are constructed as follows. For each tweet, we extract the links using regular expressions and crawl the corresponding articles, which results in a tweet-link relation. The link-hashtag relation is created by connecting an article to any hashtag it mentions. The user-hashtag relation is created by connecting a user to a hashtag they used in their tweets. The user-link relation stems from connections of a user to an article they mentioned in their tweets.

Feature engineering. Features of each individual entity are engineered as follows. Static features (similarity-based) have been extracted directly from the Twitter REST API [27], including user features such as registration age. To assess the credibility of a user, we relied on Tweetcred framework [21], which is an aggregation of 45 characteristics such as #retweets, #favorites, #replies, and presence of swear words into Likert Scale (score 1-5). The credibility feature of linked articles was assessed using the Alexa ranking (higher ranking, higher credibility). Popularity of hashtags was quantified using semantic ranking [10]. The linguistic style of tweets and linked articles was evaluated using OpenIE framework [34]. Each linguistic feature is measured as the fraction of English words in a tweet that reflect the writing style of the user. Six linguistic features are used: discrepancy words (e.g., could, would), tentative words (e.g., perhaps), filter words (e.g., I mean), punctuations, swear words (e.g., damn), and exclusion words (e.g., but).

Dynamic features (history-based) are extracted using the Twitter Streaming API [32]. For instance, the number of retweets of a tweet is collected over time by monitoring the respective tweet. Similarly process is used for status frequencies, numbers of followers and friends of a user. Numbers of tweets as well as mentions of hashtags and links were obtained using this way.

Similarly, data is collected for features of relations. For example, the difference between the time mentioned in a tweet and given in a linked article is assessed for all tweets in a specific time window. Then, upon receiving a tweet that links to an article, the respective time difference can be compared to those observed for historic data. Location and event features, in turn, are binary and capture whether the tweet and link originate from the same location or event.

Datasets. The collected data comprises 4 million tweets, 3 million users, 28893 hashtags, and 305115 linked articles, revolving around 1022 rumours from 01/05/2017 to 01/11/2017. This period was chosen as it contains several rumours, e.g., related to the Las Vegas shooting and information published by the US administration. Our data spans over 20 different domains, available at [8]. Here, we report results for the most popular ones:

- *Politics*: rumours related to all political issues.
- *Fraud & Scam*: rumours related to online hoax/scam entreating users to share posts and photographs under the false premise of a greater good.
- *Fauxtography:* rumours related to images or videos circulating on the Web.
- *Crime*: rumours related to criminology and incidents, such as the Las Vegas shooting.
- *Science & Technology:* rumours related to scientific myths and exaggerated technological inventions.

Each of the datasets is a full view of the social graph. The modelled entity types, relation types, and features are summarised in Table 1.

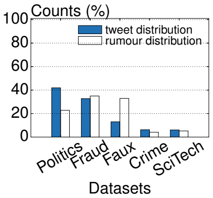


Figure 5: Data distributions

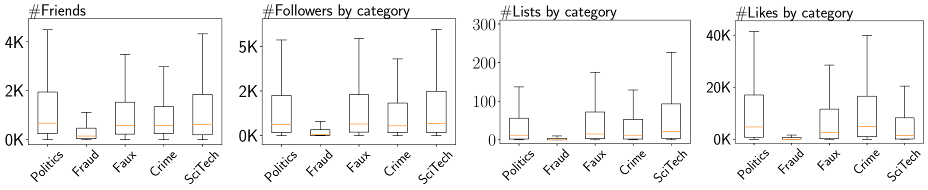


Figure 6: Relations between user features and rumours

7.3 Understanding Rumour Characteristics

What are the rumours about? In order to understand the diffusion of rumours on social platforms, we plot the distribution of rumours with their respective tweets in Fig. 5. The top-3 domains with the most number of rumours and tweets are *Politics*, *Fraud & Scam*, and *Fauxtography*. In total, they comprise over 80% of number of rumours and tweets. This implies that rumours are easily spread in the domains where being right or wrong is rather subjective.

We also observe discrepancies between the number of rumours and the number of tweets in each domain. Although the majority of tweets is in the Politics domain, the number of rumours belonging to this domain is only the third highest. As political rumours are controversial, they tend to attract more interactions, leading to a high number of tweets [47]. On the other hand, although more than 30% of rumours are Fauxtography, only 10% of the tweets belong to this category. An explanation may be that false pictures are easy to create, but may not deceive people easily.

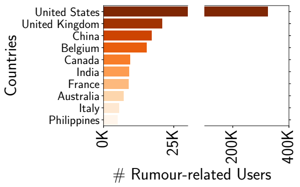


Figure 7: Users by countries

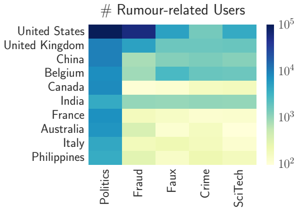


Figure 8: Users by domains

Who post rumours? To investigate the features of rumour-related users, Fig. 6 displays boxplots of the relations between the number of friends, followers, lists [25] (groups on Twitter that a user can subscribe to) and likes of a user and the domain of rumours to which they contributed. Interestingly, users who post fraud & scam tweets have lower numbers of features on average in comparison with other domains. Moreover, there seems to be no correlation between the number of friends and followers and the domain of rumours.

Where are the rumours from? Fig. 7 shows the number of users who tweet about rumours by country. Here, the most prominent countries are English-speaking (US, UK) or populous (China, India). The majority of users in our dataset, however, resides in the US, with nearly 0.4M users. Fig. 8 analyses whether there is an indication that the location of the users affects the domain of their tweets. The top popular domains for most countries are *Politics*, *Fraud*, and *Faux*, which is similar to the top domains in overall. This fits with the data collection period after the 2016 US presidential election.

In Fig. 9, we show a histogram of the numbers of users who post tweets related to different rumours. The histogram follows a long-tail distribution in which most users tweet about 1-2 rumours. There are users who tweet about more than 100 rumours. However, their number is extremely small. Analysing these users, we identify several interesting characteristics. The accounts who post about most rumours are extremely similar. We suspect that they are bots or part of a network. Given our focus on rumour detection, however, we refer to [47] for an in-depth analysis of user accounts.

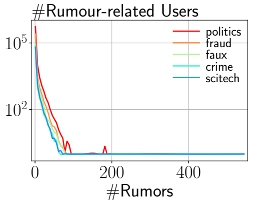


Figure 9: Users who tweet-/retweet rumours

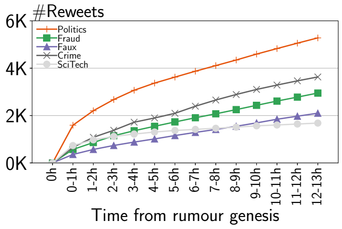
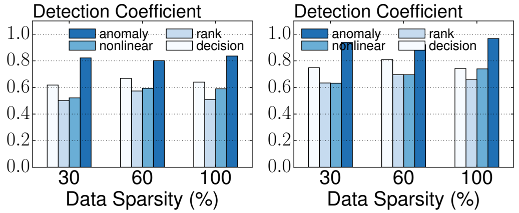


Figure 10: Propagation of rumours



(a) Politics

(b) Crime

Figure 11: Rumour Detection Coefficient across datasets

How do rumours propagate? To illustrate the propagation of rumours, we collect the number of retweets per tweet, which is a measure of its influence. Fig. 10 shows the number of retweets per rumor per domain in the first 13 hours.

We observe that political rumours are extremely bursty. In the first hour, the average number of retweets of these rumours is over 1000, which indicates that these rumours can spread in a short amount of time. After the first hour, these rumours keep propagating extremely fast, following a linear trend. Therefore, it is important that rumours belonging to this domain are detected early. On the other hand, rumours in other domains follow a log-scale increase after the first hour. In addition, rumours in these domains are not as bursty. The number of retweets after the first hour is moderate as most of them have less than 500 retweets in the first hour.

7.4 Effectiveness of Rumour Detection

Detecting rumourous tweets. We evaluate the detection coefficient of our approach versus the baseline methods in Fig. 11 for the domains *Politics* and *Crime* (the same trends emerge for the other domains). We vary the amount of rumours contained in the dataset, i.e. data sparsity, by randomly removing some rumours, so that the remaining rumours cover 30%, 60%, 100% of the original count.

In general, our approach outperforms the baseline methods in the detection of rumour-related tweets. For instance, taking the results of the *Politics* dataset, when considering 30% of the rumours, our approach achieves a coefficient of 0.82, whereas the best baseline method achieves solely a coefficient of 0.62.

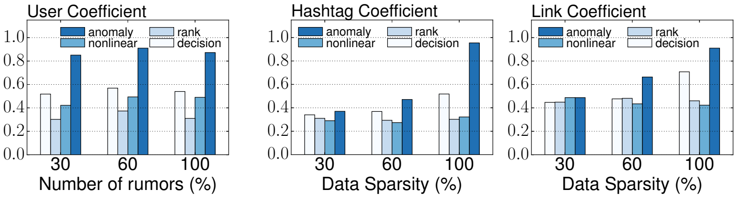
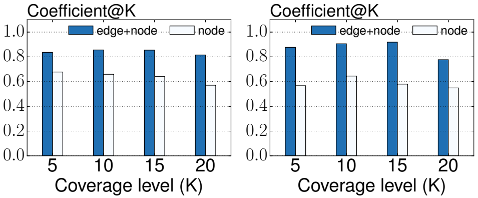
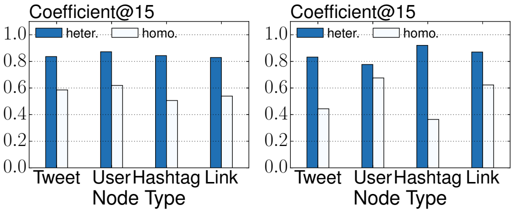


Figure 12: Coefficients for different modalities



(a) Politics (b) Crime

Figure 13: With vs. without relations



(a) Politics (b) Crime

Figure 14: Heterogeneous vs. homogeneous graphs

Going beyond the detection of tweets. Our multi-modal approach enables not only the detection of rumour-related tweets, but also rumour-related users, hashtags and links. We therefore evaluated the effectiveness of rumour detection for these modalities, in comparison with the baseline methods. As the baseline methods detect solely rumour-related tweets,we used these tweets to determine rumour-related users, hashtags, and links that are their direct neighbours in the social graph. We assessed the performance of our approach and the baseline methods in terms of the achieved coefficient, when varying the amount of rumours contained in the dataset.

Fig. 12 shows the results obtained for users, links and hashtags on *Politics* (results for other datasets are similar). Our approach still outperforms in the detection of rumour-related users, links, and hashtags. This is expected as our approach incorporates multiple modalities explicitly, which yields a synergistic effect when trying to detect rumour-related entities of different types.

7.5 Model Design Choices

Effects of Relations. We analyse the effect of considering relations of the social graph when detecting rumours. To this end, we detected anomalies using only entities (*node*) and compare the results to our actual approach (*edge+node*). We varied the coverage level in Alg. 1 to obtain multiple anomalous subgraphs.

The results in Fig. 13 show that using solely entities yields worse coefficients, e.g., a value of 0.64 instead of 0.85, when considering $K = 15$ in the *Politics* dataset (again, trends are consistent over all domains). This highlights that relations constitute an important source of information for rumour detection.

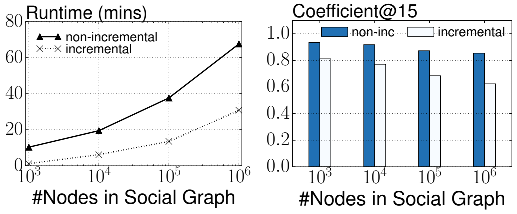


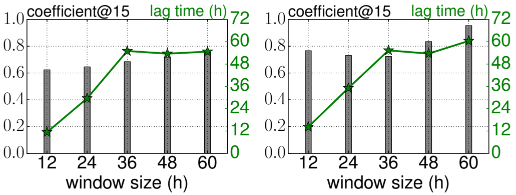
Figure 15: Incremental vs. non-incremental

Effects of Multi-Modality. We further evaluated the impact of multi-modal information, by comparing our approach with rumour detection based on homogeneous graphs, built of a single modality. The respective modality is then taken as the target for rumour detection, e.g., the user graph is used to detect rumour-related users. We measure the detection coefficient, while considering the best coverage level $K = 15$ from the previous experiment.

As illustrated in Fig. 14 for two domains, the multi-modal social graph yields a better coefficients. This underlines the importance of a rich model, with multiple modalities, for rumour detection.

7.6 Scalability and Streaming Settings

Effects of data size. This experiment compared the non-incremental and incremental versions of our approach. We constructed sub-datasets to vary the number of nodes in the social graph of the *Politics* dataset from 10^3 to 10^6 and compare the observed coefficient and run-time. Fig. 15 shows that the incremental computation indeed improves the run-time of our approach, halving the time needed to process a graph of size 10^6. Moreover, the error introduced by incremental computation stays within reasonable bounds.



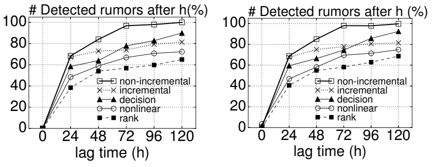
(a) Politics (b) Crime

Figure 16: Streaming setting: effects of window size

Effects of window size. We varied the window size, from 12 to 60 hours, while considering the coverage level $K = 15$. The results in terms of coefficients and lag time to detection are shown in Fig. 16. With larger windows, the coefficient increases, since rumour detection exploits more information. The lag time to detection also increases, until reaching a plateau. Again, this is due to the amount of available information. Initially, some rumours cannot be detected and thus do not affect the lag time. With larger windows, these rumours are detected and increase the lag time.

Distribution of lag time. Further, we studied the relation between lag time and detection accuracy. For our incremental approach, we computed the lag time for each rumour and aggregate them into several bins. For all other methods, we constructed datasets with varying detection deadlines θ, controlling that for each rumour, only tweets from the start of the rumour (θ_0) until $\theta_0 + \theta$ are kept. We then report the percentage of detected rumours for each such deadline. According to Fig. 17, our approaches outperform the baseline methods, especially for small lag times. For instance, in the *Politics* dataset, with a lag time of 48 hours, our non-incremental approach detects 84% of rumours, whereas the best baseline achieves 64%.

Average delay analysis. We provide a fine-grained view of the lag time by computing the difference between the timestamps at which the same rumour was *first* detected (i.e. any tweet related to that rumour is flagged) by different methods. Table 3 presents the analysis within 1 day after the genesis of rumours. Our approach detect rumours earlier than the baselines a few hours in average.



(a) Politics (b) Crime

Figure 17: Timeliness of rumour detection

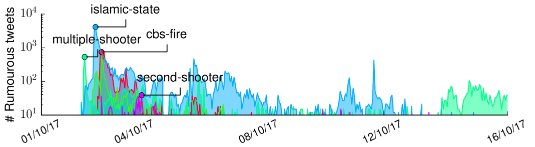
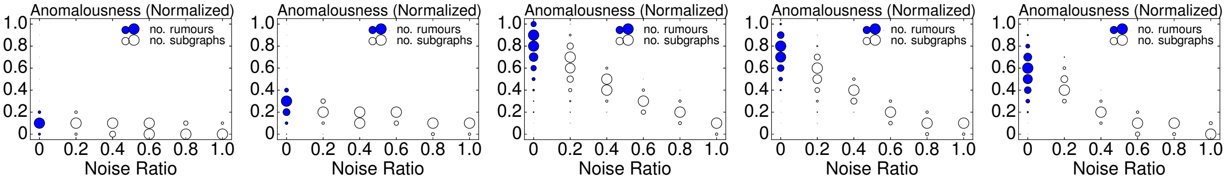


Figure 18: Timeline of rumours about the Las Vegas shooting in October 2017

Table 3: Delay analysis (within 1 day)

Baseline	Rumours detected	Average delay
our	68.29%	+0.0h
decision	59.46%	+1.7h
nonlinear	47.73%	+2.3h
rank	38.23%	+ 3.1h



(a) 2 days before genesis (b) 1 day before genesis (c) Genesis time (d) 1 day after genesis (e) 2 days after genesis

Figure 19: Correctness of anomaly scores

7.7 Case Study

Effects of timeline. Fig. 18 highlights some detected rumours from the *Crime* dataset along a timeline. Most of the rumours are related to the Las Vegas shooting, one of the biggest events of the year that attracts many hoaxes, fake news, and viral misinformation [7]. It plots the hourly numbers of tweets for each rumour. Here, most rumours occurred around on October 2, the date of the incident. Most rumour-related tweets are about the shooting being caused by a member of ISIS. Also, two days after the incident, there was rumour that the shooter had an accomplice (second-shooter rumour).

Correctness of anomaly scores. Fig. 19 depicts the correctness of our anomalousness measure on subgraphs. When a rumour happens (genesis), we compute its anomalousness score, do the same at ± 1 day and ± 2 days, and then normalize by the maximum values among all rumours. These scores are compared with those of other subgraphs, which are constructed by randomly adding regular tweets into the rumours (this noise ratio is varied from 0.0 to 1.0). Finally, we do a histogram by counting the number of rumours and subgraphs with scores that fall into 0.1-bins.

At the genesis, the scores of positive samples (i.e. rumours) turn out to be significantly higher than noisy samples (i.e. other subgraphs), supporting effective detection. Before the genesis, anomaly scores are small and nearly uniform, as historic data is not anomalous. After the genesis, the scores decrease. Yet, they are still relatively high, since anomalies are still present, which captures the temporal movement of rumours.

8. RELATED WORK

Rumour detection. While there is a large body of work on rumour detection on social platforms, surveyed in [56], little has been done to exploit multiple modalities to detect rumours. Most work leverages only textual data such as tweets [13, 55, 21]; whereas others consider different data entities such as users and hashtags but still treat them as additional features or textual data only [30, 19]. Techniques based on hand-crafted features [13, 55, 50] are grounded in an ad-hoc definition of features, which are expected to be strong indicators of rumours. Recently, deep features based on temporal dependencies of the posts have been proposed [30]. While this approach achieves high detection accuracy, it first requires the detection of an explicit event and thus depends on the accuracy of this event detection step. There are further approaches [31, 48] that take into account how rumours propagate. However, these techniques require large collections of tweets to conduct the respective analysis. As such, they cannot be expected to yield small lag times in the detection of rumours and are not well-suited for a streaming setting. Our approach is the first to leverage not only the textual data, but also other modalities in both offline and online settings.

Anomaly detection. Anomaly detection can be classified into point or group-based techniques [53]. Point-based anomaly detection aims to detect individuals, for which the behaviour is different from the general population [39, 24, 22]. Group-based anomaly detection, in turn, strives for groups of individuals that collectively behave differently compared to some population [15, 17, 16, 52, 33, 49]. However, none of the above techniques has been applied to rumour detection. While [16] addresses a similar use case, it neglects the anomalies related to feature differences between entities. Our technique is the first one for group-based anomaly detection that simultaneously identify anomalies in all features, entities, and relations. Most of the work on anomaly detection in general and rumour detection in particular focuses on accuracy. Here, we define the detection coefficient to capture the balance between accuracy and completeness, which is optimised by our approach.

Information networks. There exists various graph-based models for data of social platforms, referred to as information networks [40]. Some models capture real-world entities, such as users and posts [43], while others represent derived data elements, such as topics [44]. Existing work on anomaly detection in information networks focuses on modelling the propagation patterns of known phenomena [20, 57] or classifies known events [55]. This setting is orthogonal to our work, since we strive for the detection of phenomena that emerge on social networks, but are not known a priori.

9. CONCLUSION

This paper proposed an approach for rumour detection that is grounded in the anomalies of a social graph. Unlike traditional approaches that focus only on accuracy, we optimised the detection coefficient, which represents the trade-off between accuracy and completeness. We presented a two-step detection approach that detects anomalies at the local and global level. While the former increases the completeness of detection by reducing false negatives, the latter optimises the detection accuracy by reducing false positives. Our experiments showed that our method is effective and efficient, detecting rumours early and accurately. It outperformed several baselines in both static and streaming settings.

10. REFERENCES

[1] http://docs.spinn3r.com/.

[2] https://www.engadget.com/2018/08/21/facebook-rates-user-trustworthiness/.

[3] https://www.networkworld.com/article/2235277/data-center/data-center-fact-checking-the-fact-checkers-snopes-com-gets-an-a.html.

[4] https://www.snopes.com/fact-check/las-vegas-shooting-rumors-hoaxes-and-conspiracy-theories/.

[5] https://www.snopes.com/fact-check/trump-aid-puerto-rico/.

[6] https://www.theverge.com/2018/8/21/17763886/facebook-trust-ratings-fake-news-reporting-score.

[7] http://tiny.cc/las-vegas-shooting.

[8] http://tiny.cc/p1s2qy.

[9] E. Álvarez-Miranda, I. Ljubić, and P. Mutzel. The maximum weight connected subgraph problem. In *Facets of Combinatorial Optimization*, pages 245–270. 2013.

[10] P. Bansal, S. Jain, and V. Varma. Towards semantic retrieval of hashtags in microblogs. In *WWW*, pages 7–8, 2015.

[11] R. H. Berk and D. H. Jones. Goodness-of-fit test statistics that dominate the kolmogorov statistics. *Probability theory and related fields*, pages 47–59, 1979.

[12] M. Buckland and F. Gey. The relationship between recall and precision. *Journal of the American society for information science*, 45(1):12–19, 1994.

[13] C. Castillo, M. Mendoza, and B. Poblete. Information credibility on twitter. In *WWW*, pages 675–684, 2011.

[14] S. Cazalens, J. Leblay, P. Lamarre, I. Manolescu, and X. Tannier. Computational fact checking: a content management perspective. *PVLDB*, 11(12):2110–2113, 2018.

[15] V. Chandola, A. Banerjee, and V. Kumar. Outlier detection: A survey. *ACM Computing Surveys*, 2007.

[16] F. Chen and D. B. Neill. Non-parametric scan statistics for event detection and forecasting in heterogeneous social media graphs. In *KDD*, pages 1166–1175, 2014.

[17] K. Das, J. Schneider, and D. B. Neill. *Detecting anomalous groups in categorical datasets*. Carnegie Mellon University, 2009.

[18] R. Diestel. *Graph theory*. Springer Publishing Company, Incorporated, 2018.

[19] C. T. Duong, Q. V. H. Nguyen, S. Wang, and B. Stantic. Provenance-based rumor detection. In *ADC*, pages 125–137, 2017.

[20] A. Friggeri, L. A. Adamic, D. Eckles, and J. Cheng. Rumor cascades. In *ICWSM*, 2014.

[21] A. Gupta, P. Kumaraguru, C. Castillo, and P. Meier. Tweetcred: Real-time credibility assessment of content on twitter. In *SocInfo*, pages 228–243, 2014.

[22] A. Ihler, J. Hutchins, and P. Smyth. Adaptive event detection with time-varying poisson processes. In *KDD*, pages 207–216, 2006.

[23] X. Jin, C. X. Lin, J. Luo, and J. Han. Socialspamguard: A data mining-based spam detection system for social media networks. *PVLDB*, 4(12):1458–1461, 2011.

[24] W.-H. Ju and Y. Vardi. A hybrid high-order markov chain model for computer intrusion detection. *Journal of Computational and Graphical Statistics*, 10(2):277–295, 2001.

[25] D. Kim, Y. Jo, I.-C. Moon, and A. Oh. Analysis of twitter lists as a potential source for discovering latent characteristics of users. In *ACM CHI workshop on microblogging*, page 4, 2010.

[26] M. Kulldorff. A spatial scan statistic. *Communications in Statistics-Theory and methods*, pages 1481–1496, 1997.

[27] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *WWW*, pages 591–600, 2010.

[28] S. Kwon, M. Cha, and K. Jung. Rumor detection over varying time windows. *PloS one*, 12(1):e0168344, 2017.

[29] J. Leblay, I. Manolescu, and X. Tannier. Computational fact-checking: Problems, state of the art, and perspectives. In *The Web Conference*, 2018.

[30] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, and M. Cha. Detecting rumors from microblogs with recurrent neural networks. In *IJCAI*, pages 3818–3824, 2016.

[31] J. Ma, W. Gao, and K.-F. Wong. Detect rumors in microblog posts using propagation structure via kernel learning. In *ACL*, pages 708–717, 2017.

[32] F. Morstatter, J. Pfeffer, H. Liu, and K. M. Carley. Is the sample good enough? comparing data from twitter's streaming api with twitter's firehose. In *ICWSM*, 2013.

[33] K. Muandet and B. Schölkopf. One-class support measure machines for group anomaly detection. *arXiv preprint arXiv:1303.0309*, 2013.

[34] S. Mukherjee, G. Weikum, and C. Danescu-Niculescu-Mizil. People on drugs: credibility of user statements in health communities. In *KDD*, pages 65–74, 2014.

[35] D. B. Neill. Fast subset scan for spatial pattern detection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(2):337–360, 2012.

[36] A. Olteanu, C. Castillo, N. Diakopoulos, and K. Aberer. Comparing events coverage in online news and social media: The case of climate change. In *ICWSM*, pages 288–297, 2015.

[37] A. Olteanu, C. Castillo, F. Diaz, and S. Vieweg. Crisislex: A lexicon for collecting and filtering microblogged communications in crises. In *ICWSM*, pages 376–385, 2014.

[38] V. Pham, T. Bluche, C. Kermorvant, and J. Louradour. Dropout improves recurrent neural networks for handwriting recognition. In *ICFHR*, pages 285–290, 2014.

[39] M. Schonlau, W. DuMouchel, W.-H. Ju, A. F. Karr, M. Theus, and Y. Vardi. Computer intrusion: Detecting masquerades. *Statistical science*, pages 58–74, 2001.

[40] C. Shi, Y. Li, J. Zhang, Y. Sun, and S. Y. Philip. A survey of heterogeneous information network analysis. *TKDE*, 29(1):17–37, 2017.

[41] S. Speakman, Y. Zhang, and D. B. Neill. Dynamic pattern detection with temporal consistency and connectivity constraints. In *ICDM*, pages 697–706, 2013.

[42] Y. Sun, C. C. Aggarwal, and J. Han. Relation strength-aware clustering of heterogeneous information networks with incomplete attributes. *PVLDB*, 5(5):394–405, 2012.

[43] Y. Sun and J. Han. Mining heterogeneous information networks: principles and methodologies. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 3(2):1–159, 2012.

[44] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In *WWW*, pages 1067–1077, 2015.

[45] I. Taxidou and P. Fischer. Realtime analysis of information diffusion in social media. *PVLDB*, 6(12):1416–1421, 2013.

[46] N. Thanh Tam, M. Weidlich, H. Yin, B. Zheng, N. Quoc Viet Hung, and B. Stantic. User guidance for efficient fact checking. *PVLDB*, 12(8):850–863, 2019.

[47] S. Vosoughi, D. Roy, and S. Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

[48] K. Wu, S. Yang, and K. Q. Zhu. False rumors detection on sina weibo by propagation structures. In *ICDE*, pages 651–662, 2015.

[49] L. Xiong, B. Póczos, J. G. Schneider, A. Connolly, and J. VanderPlas. Hierarchical probabilistic models for group anomaly detection. In *AISTATS*, pages 789–797, 2011.

[50] F. Yang, Y. Liu, X. Yu, and M. Yang. Automatic detection of rumor on sina weibo. In *KDD*, page 13, 2012.

[51] J. Ye, S. Kumar, and L. Akoglu. Temporal opinion spam detection by multivariate indicative signals. In *ICWSM*, pages 743–746, 2016.

[52] R. Yu, X. He, and Y. Liu. Glad: group anomaly detection in social media analysis. *TKDD*, 10(2):18, 2015.

[53] R. Yu, H. Qiu, Z. Wen, C. Lin, and Y. Liu. A survey on social media anomaly detection. *ACM SIGKDD Explorations Newsletter*, 18(1):1–14, 2016.

[54] F. Zhang, W. Zhang, Y. Zhang, L. Qin, and X. Lin. Olak: an efficient algorithm to prevent unraveling in social networks. *PVLDB*, 10(6):649–660, 2017.

[55] Z. Zhao, P. Resnick, and Q. Mei. Enquiring minds: Early detection of rumors in social media from enquiry posts. In *WWW*, pages 1395–1405, 2015.

[56] A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, and R. Procter. Detection and resolution of rumours in social media: A survey. *arXiv preprint arXiv:1704.00656*, 2017.

[57] A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, and R. Procter. Detection and resolution of rumours in social media: A survey. *CSUR*, 51(2):32, 2018.