Detecting Spam Tweets in Trending Topics Using Graph-Based Approach

Ramesh Paudel(✉), Prajjwal Kandel, and William Eberle

Tennessee Technological University, Cookeville, TN 38501, USA
{rpaudel42,prkandel42}@students.tntech.edu, weberle@tntech.edu

Abstract. In recent years, social media has changed the way people communicate and share information. For example, when some important and noteworthy event occurs, many people like to "tweet" (Twitter) or post information, resulting in the event trending and becoming more popular. Unfortunately, spammers can exploit trending topics to spread spam more quickly and to a wider audience. Recently, researchers have applied various machine learning techniques on accounts and messages to detect spam on Twitter. However, the features of typical tweets can be easily fabricated by the spammers. In this work, we propose a graph-based approach that leverages the relationship between the named entities present in the content of the tweet and the document referenced by the URL mentioned in the tweet for detecting possible spam. It is our hypothesis that by combining multiple, heterogeneous information together into a single graph representation, we can discover unusual patterns in the data that reveal spammer activities - structural features that are difficult for spammers to fabricate. We will demonstrate the usefulness of this approach by collecting tweets and documents referenced by the URL in the tweet related to Twitter trending topics, and running graph-based anomaly detection algorithms on a graph representation of the data, in order to effectively detect anomalies on trending tweets.

Keywords: Twitter · Spam detection · Anomaly Detection · Graph-based anomaly

1 Introduction

Twitter, a popular social media, or micro-blogging, site, allows users to post information, updates, opinions, etc. using tweets. Given its wide-spread popularity for immediately sharing thoughts and ideas, adversaries try to manipulate the micro-blogging platform and propagate off-topic content for their selfish motives [1,2]. Compounding the issue, as the popularity increases around a certain event, more people tweet about the topic, thereby increasing its "trending" rate. Spammers then exploit these popular, trending Twitter topics to spread their own agendas by posting tweets containing keywords and hash-tags of the trending topic along with their misleading content. Ideally, one would like to be

K. Arai et al. (Eds.): FTC 2019, AISC 1069, pp. 526–546, 2020.
https://doi.org/10.1007/978-3-030-32520-6_39

able to identify anomalous tweets on a trending topic that have the potential to mislead the population, or even possibly cause further harm. Currently, Twitter allows users to report spam, and after an investigation, an account can be suspended. However, suspending a spam account is not an efficient technique to deal with spam related to trending topics because the suspension process is slow, and the trending topics usually last for only a few hours or a day at most [2]. Therefore, the focus of the *anomaly detection on trending topics* in this work is on the detection of tweets containing spam, instead of detecting spam accounts.

One of the more malicious activities involves a spammer who includes a URL in the tweet, leading the reader to a completely unrelated website. It is reported that 90% of anomalous tweets contain unrelated or misleading URLs [1,3]. People use shortened URLs or links in their tweet because of the limited number of characters (280) available in the tweet. Since it is common for tweets to include shortened text, so as to fit within the character limits, spammers can conceal their unrelated/malicious links with shortened URLs. Hence, the problem with a shortened URL is that users do not know what is the actual domain until the link is clicked. The existing approaches for spam detection on Twitter use various machine learning tools on user-based features (e.g. number of followers, number of tweets, age of the user account, number of tweets posted per day or per week, etc.) and content-based features (e.g. number of hashtags, mentions, URL, likes, etc.) [1,3–9]. Though user and content-based features can be extracted efficiently, an issue is that these features can also be fabricated easily by the spammer [2,10,11]. However, being able to hide an inconsistency between the topic of a tweet and the topic of the document referred by URLs in the tweet is much harder [2,12].

In this work, we propose an unsupervised, two-step, graph-based approach to detect anomalous tweets on trending topics. First, we extract named entities (like place, person, organization, product, event, or activity) present in the tweet and add them as key elements in the graph. As tweets on a certain topic share the contextual similarity, we believe they also share same/similar named entities. These named entities representing relevant/similar topics can have a relationship (e.g., shared ontology) amongst themselves, which we believe if represented properly, will provide broader insight on the overall context of the topic. As such, graphs can be a logical choice for representing these kinds of information where a node can represent a named entity and an edge can represent the relationships between them. Using a well-known graph-based tool like GBAD [13], we then discover the normal and anomalous behavior of a trending topic. Second, we propose adding hyperlinked document information because anomalies that could not be detected from tweets alone could be detected using both the document and tweets. It is our assumption that a better understanding of patterns and anomalies associated with entities like person, place, or activity, cannot be realized through a single information source, but better insight can be realized using multiple information sources simultaneously. For instance, one can discover interesting patterns of behavior about an individual through a single social media account, but better insight into their overall behavior can be realized by examining all of their social media actions simultaneously. Analyzing multiple information sources for anomaly detection on Twitter has been explored in the past. For example, the inconsistencies between the tweet and the document referred to by a URL in the tweet using cosine similarity [12] and a language model [2] were studied for potential anomaly detection. But, the cost for [2] is high as each tweet with a link is treated as a suspect and [12] need a predefined source of reliable information for each topic which makes these approaches less flexible in real-time trending topics.

Using the above mentioned 2-step approach, we aim to detect the following types of spam/anomalies in trending tweets that are consistent with the spam scenarios listed by Twitter [14].

1. *Keyword/Hashtag Hijacking*: Using popular keywords or hashtags to promote the tweet that are not related to the topic. This is done to promote anomalous tweets to a wider audience by hijacking popular hashtags and keywords.
2. *Bogus link*: Posting a URL that has nothing to do with the content of the tweet. This is done to generate more traffic to the website. Another scenario of bogus link is link piggybacking. For example, posting an auto redirecting URL that goes to legitimate website but only after visiting an illegitimate website. Another way is to post multiple links where one link can be a legitimate link while another can be a malicious or unrelated link. The motivation behind link piggybacking is to generate traffic to the illegitimate website by concealing the link inside a legitimate website. This can also be accomplished by using a tiny URL.

To verify our approach, we collect tweets (containing URLs) related to two separate (and very different) trending topics during the summer of 2018: FIFA World Cup and NATO Summit (Sect. 3). We then construct graphs using information from the tweet text and the document referred inside the tweet (Sect. 4), followed by using a graph-based anomaly detection tool (Sect. 5). We then compare the performance of our proposed approach with several existing approaches to show the effectiveness of a graph-based approach (Sect. 6).

2 Related Work

Anomaly detection in Twitter data can be about detecting anomalous/spam accounts or about detecting anomalous/spam tweets. Verma et al. [5] summarize spam account detection techniques in Twitter along with their analysis and comparison. Similarly, Tingmin et al. [15] present the detailed analysis, discussion and comparative studies of existing approaches on both spam accounts and spam tweets detection. Most of the approaches for detecting spam accounts use user-based and content-based features. Benevenuto et al. [1] use SVM on 62 sets of user-based and content-based features to classify spammer and non-spammer accounts. Soman and Murugappan [16] use fuzzy K-mean clustering to group the similar user profiles with the same trending topics, and an extreme learning machine (ELM) algorithm is applied to analyze the growing characteristics of spam with similar topics in Twitter from the clustering result.

Wu et al. [17] propose WordVector and deep-learning techniques to extract text-based features that are hard to fabricate by the spammer. These features are then fed into traditional classifiers. Boididou et al. [18] propose a semi-supervised approach based on bagging that uses different sets of tweet-based (TB) and user-based features (UB). Meda et al. [9] apply Random Forest by using only 5 features on the same dataset used by [1] and got comparable results to [1]. Ameen and Kaya [6] compare the performance of Naive Bayes, Random Forest, J48 and IBK classifiers to classify 1135 user accounts into spammer and non-spammer accounts. They report that Random Forest shows the best results among the four classifiers. Another approach that demonstrates the superiority of Random Forest over other machine learning algorithms (Naïve Bayesian, Support Vector Machine, and K-NN) to detect spam accounts is by McCord and Chuah [8].

Chao et al. [11] collected over 600 million public tweets, labelled around 6.5 million spam tweets, extracted 12 light weight statistical features, and conduct experiments on different machine learning algorithms simulating various scenarios. They did so to better understand the effectiveness and weakness of different algorithms for timely Twitter spam detection. Though user and content-based features can be extracted efficiently, an issue is that these features can also be fabricated easily by the spammer [2,10,11]. A more sophisticated approach by Wang [4] uses content-based along with graph-based features on a user's "follower" and "friend" relationships. These features are provided to a Bayesian classifier for classifying spam and non-spam accounts. Yang et al. [19] designed another sophisticated and robust approach by analyzing evasive techniques of a spammer. They propose the use of 10 detection features: 3 graph-based, 3 neighbor-based, 3 automation-based, and 1 timing-based feature as the input for several machine learning algorithms. Lee et al. [20] use social honeypots for harvesting deceptive spam accounts and perform statistical analysis on a spam account's properties to create spam classifiers to actively filter out existing and new spammers.

Besides spam account detection, there has been some research focusing on anomalous content. The author in [7,21] propose an anomalous tweet detection scheme by leveraging the features of embedded URL in the tweets. URL-based features show the discriminative power for classifying spam but these schemes can only detect anomalous tweets containing URLs and miss anomalous tweets containing only text or fabricated URLs [22]. Anantharam et al. [12] analyze tweet content along with documents referenced in the URL for assessing their relevance to an event/topic. Gupta et al. [23] propose the framework that takes the user-based, content-based and tweet-text features to classify the tweets using a Neural Network with accuracy up to 91.65%. Song et al. [10] use distance and connectivity between a message sender and a message receiver by constructing directed graphs based on the following and followed relations in Twitter to decide whether the message is spam or not. Martinez-Romo and Araujo [2] effectively classified trending tweets as spam or non-spam by exploiting the divergence between the statistical language models in the topic, the tweet, and the page linked from the tweet. This was based on the fact that if the tweet is spam, language models are likely to be different as the spammer usually tries to divert traffic to sites that have no semantic relation [2].

Though the use of graph-based features have better results in detecting spam accounts/tweets [4,10,19], little or no work has been done applying graph mining techniques to detect anomalous URL links in tweets. Graphs provide a powerful machinery for effectively capturing the long-range correlations among interdependent data objects/entities. Graph-based approaches have been successfully applied for anomaly detection in a wide array of applications [24]. Therefore, we plan to extend a graph-based approach for detecting spam tweets on trending topics. In this work, we will use the publicly-available graph-based anomaly detection (GBAD) tool [13] that has been used for detecting network intrusions [25,26], reporting anomalies in telecom data [27], discovering unusual elderly patient activities in a smart homes [28], detecting anomalous activity and potential fraud scenarios in medicare claim files [29], etc.

3 Data

The dataset used in this research consists of tweets and documents (primarily news stories) mentioned in the tweets. The detailed process of data collection and the description of data is presented in this section.

The data was collected using Twitter's standard search API. We collected tweets related to two trending topics, "FIFA World Cup" and "NATO Summit", during the summer of 2018. The results from Twitter's search API contains tweet text, Twitter handle name, any hashtags and URLs mentioned in the tweet, as well as all publicly available information about the user including their name. The data for tweets is a JSON dump of individual tweets.

We added a news filter onto the query to gather a sample of tweets containing links to news articles. We also added a filter for the English language so that we only get English tweets. We observed that the textual content of some tweets is the same as the title of the news (referred by the URL). Since our objective is to combine multiple sources of information together to learn new information, these particular tweets (containing the title of the news) does not provide any additional content (i.e. redundant). So, we used a python-based library called "difflib" to filter out tweets whose text content matches the title of the article referred by the URL in the tweet. To acquire the news information from the news URL, we used a python-based crawler called "Newspaper"[1]. Using Newspaper, we extracted the news title, body, summary, author name, published URL, domain name of the published website, and published date. Because of the challenges like dead URLs, URLs leading to multimedia content (videos or photos), non-English tweets, and links to non-English documents, the resulting dataset is smaller than we hoped. But, as we will show, it still demonstrates the effectiveness of our proposed approach in addressing the problem.

[1] https://github.com/codelucas/newspaper.

Table 1. Total number of tweet/news and anomalies in each topic

Trending topic	Total tweet/news	Anomalies		
		Keyword hijacking	Bogus link	Total
World Cup	1,463	2	20	22
NATO Summit	1,716	0	11	11

After data collection, we manually inspected the data for the amount of spam present in the dataset so that we can measure the performance of our approach. It should be noted that our approach is unsupervised and does not need any labelled data. We used the following criteria that are consistent with [1] to label the spam tweet.

- If the tweet has keywords related to the trending topic but the document referred by the URL does not have any.
- If the tweet has multiple links and if any of the links refer to a document not related to the trending topic.
- If the tweet has a URL that redirects to an unrelated website before redirecting to the related website. This usually occur when the tweet has a tiny URL.

Table 1 shows the number of tweets/news and anomalies in two datasets. The datasets are publicly available on https://rpaudel42.github.io/pages/dataset.html.

4 Methodology

The tweets on a certain topic should have contextual similarity. It is our assumption that this contextual similarity can be perceived by the presence of same/similar named entities in the tweet. One of the key ideas of our approach is to extract these named entities, in addition to user- and content-based features, and map their relationships using a graph. We also assume that two independent named entities are considered related if they share a common ontology. We believe these entities and their relationships (ontology) will provide valuable information about the context of the tweet. First, we will run anomaly detection on a graph using the information (named entity, user-based and content-based features) from the tweets. Second, we will run anomaly detection on a graph using information from both the tweets and the documents referred by the URL in the tweet. This will allow us to demonstrate that new anomalies can be discovered using information from multiple sources related to same entity (trending topic) which otherwise could not haven been detected using single source (tweet). The basic methodology aimed at discovering anomalies in trending Twitter topics consists of four key modules: Named Entity Extractor, Ontology Generator, Graph Parser and Graph-Based Anomaly Detection, as shown in Fig. 1. We will discuss each of these modules briefly in the following section.

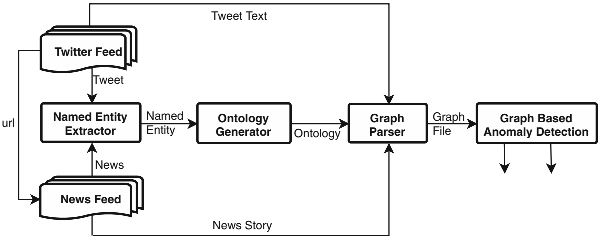


Fig. 1. Basic methodology used for experiment

4.1 Named Entity Extractor

Named entities are the real-world objects (either abstract or physical) like place, person, organization, company, date, time, etc. that can be denoted with a proper name. For our experiments, we focus particularly on three types of named entities: PERSON, ORGANIZATION and LOCATION. The python based Natural Language toolkit, NLTK [30], is used for extracting named entities from news and tweets. NLTK provides a classifier that has already been trained to recognize named entities. We collected the named entities by using the NLTK ne_chunk() function. Then, extracted entities are passed to the Ontology Generator for generating the ontology.

4.2 Ontology Generator

The primary use of an ontology in information systems is the description and structuring of shared knowledge [31]. Extracting an ontology for entities that share common structure provides added knowledge about those entities. We used DBpedia for generating the ontology. The DBpedia project has a large-scale knowledge base that extracts structured data from Wikipedia [32]. We created a python based script to query DBpedia by passing a named entity and its class (person, organization or location) to the API. The result is an ontology hierarchy that can be represented in graph form. A sample snapshot of the ontology structure for the class "Place" is given in Fig. 2. The leaf node in Fig. 2 (represented by a rectangle) are the Named Entities. For example, if the named entity is "Nashville", the result will be the hierarchy *[Place→ Populated Place→Settlement→City→Nashville]*. There can be ambiguity between entities that share common names but have no shared structure. For example, the entity "Apple" can be a technology company or a fruit. To avoid this ambiguity, we used "Apple" as the query string and "Organization" as the query class for generating an ontology for the company "Apple"; and "Apple" as the query string and "Fruit" as the query class for generating an ontology for the fruit "Apple". But if entities belong to the same ontology class (two people with same name, like John Doe the Businessman and John Doe the Athlete), the ambiguity still exists.

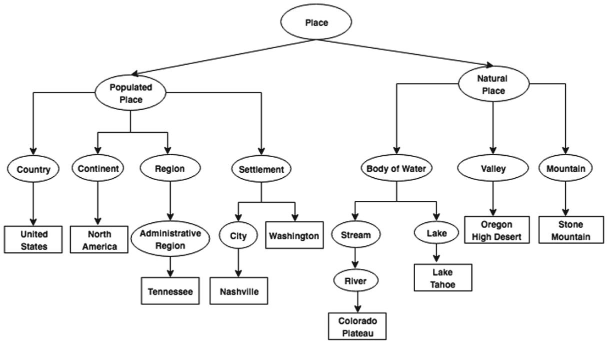


Fig. 2. Example of ontology for class place.

In this case, we have chosen to use the ontology of the first result returned by DBpedia API as that is the most likely match (i.e., most popular). The hierarchy thus generated is then passed to the Graph Parser for creating the graph used for our experiments.

4.3 Graph Parser

The desired input for our anomaly detection is a graph file. In order to create a graph input file, we created a Python-based parser that reads the data from the tweet and the news referred by the URL, and constructs the graph. The experiment is done on two graphs: a graph using information from just the tweets, and a graph using both news and tweet information. We named them *Tweet Graph* and *Mixed Graph* respectively. The basic layout of the *Mixed Graph* is shown in Fig. 3 and the process of creating the graph is described below.

In the graph, we include three content-based features (mentioned twitter handle, hashtag, and tweet posted date) and two user-based features (location and description). While we are extracting location and description as the two user-based features in the graph, we ignore other attributes like screen name, language, timezone, etc. because we cannot extract named entities (place, person, and organization) from these attributes. The Twitter account is represented by a *“handle”* node in the graph. If the Twitter account has a description, we extract the named entity (along with the ontology hierarchy) from the description and link them to the *“handle”* node with an edge *“is_about”*. Similarly, the location is linked to the *“handle”* node using an edge *“is_from”*. For example, if the handle has a location “New York” and a description “NYU Grad students, blogger, coffee lover”, the Named Entity Extractor will extract ‘NYU” and “Blogger” as the organization. This graph will therefore represent the person who is a New York University graduate student living in “New York” and loves to blog. An edge *“post”* between the Twitter handle and *“tweet”* represents that a handle posted the tweet. The entity *“hashtag”*, is linked to a tweet node with a *“has”* edge. The named entity (with their ontology hierarchy) extracted from the tweet

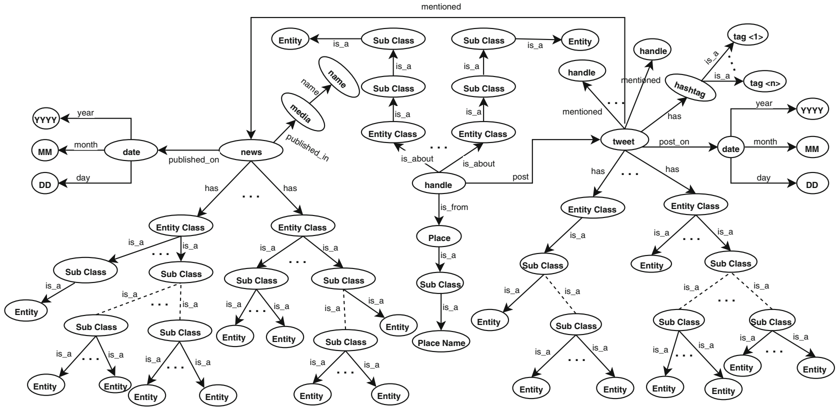


Fig. 3. Graph layout showing news-tweets features, named entities and their ontology. Dotted line represents multiple node or edge.

text are linked to the tweet node with a *"has"* edge. Tweet posted date is represented as an edge labelled *"post_on"* between *"tweet"* and *"date"* nodes. In addition, day, month, and year values are added as attributes for the date node.

The news is represented by a *"news"* node in the graph. The information from the news mentioned by the URL can be added to a *Tweet Graph* to create the *Mixed Graph*. This can be done by adding an edge labelled *"mentioned"* between *"handle"* and *"news"* nodes. Two features, published date and the publisher media, are added to the graph by representing them as a node in *Mixed Graph*. The News published date is represented as an edge labelled *"published_on"* between *"news"* and *"date"* nodes. Again, day, month and year values are added as attributes for the date node. The relationship between *"media"* and *"news"* is represented by an edge labelled *"published_in"*, and the media name is added as an attribute to "media". Named entities and their ontology are extracted from news text and added to the graph in similar way as it is done in a tweet text. It should be noted that Fig. 3 is just a visualization, as the actual graph input files are plain ASCII text files. The number of vertices and edges on the *Tweet Graph* and the *Mixed Graph* constructed for each of the datasets is shown in Table 2.

4.4 Graph-Based Anomaly Detection

In order to lay the foundation for this effort, we hypothesize that a real-world, meaningful definition of a graph-based anomaly is an unexpected deviation to a normative pattern, which is defined as follows:

Definition 1. *A labeled graph $G = (V, E, F)$, where V is the set of vertices (or nodes), E is the set of edges (or links) between the vertices, and the function F assigns a label to each of the elements in V and E.*

Table 2. Number of vertices and edges in graphs for both dataset

Graph name	Number of vertices	Number of edges
FIFA World Cup Dataset		
Tweet Graph	23,603	22,140
Mixed Graph	88,887	87,424
NATO Summit Dataset		
Tweet Graph	28,116	26,400
Mixed Graph	90,224	88,508

Definition 2. *A subgraph SA is an anomalous in graph G if* $(0 < d(SA, S) < TD)$ *and* $(P(SA|S) < TP)$, *where* $P(SA|S)$ *is the probability of an anomalous subgraph SA given the normative pattern S in G. TD bounds the maximum distance* (d) *an anomaly SA can be from the normative pattern S, and TP bounds the maximum probability of SA.*

Definition 3. *The score of an anomalous subgraph SA based on the normative subgraph S in graph G is* $d(SA, S) * P(SA|S)$, *where the smaller the score, the more anomalous the subgraph.*

The advantage of graph-based anomaly detection is that the relationships between entities can be analyzed for structural oddities in what could be a rich set of information, as opposed to just the entities' attributes. However, graph-based approaches have been prohibitive due to computational constraints since graph-based approaches typically perform subgraph isomorphism, a known NP-complete problem. Yet, in order to use graph-based anomaly detection techniques in a real-world environment, we need to take advantage of the structural/relational aspects found in dynamic, streaming data.

In order to test our approach, we will use the publicly available GBAD test suite, as defined by [13]. Using a greedy beam search and a minimum description length (MDL) heuristic, GBAD first discovers the "best" subgraph, or normative pattern, in an input graph. The MDL approach is used to determine the best subgraph(s) as the one that minimizes the following:

$$M(S, G) = DL(G|S) + DL(S)$$

where G is the entire graph, S is the subgraph, $DL(G|S)$ is the description length of G after compressing it using S, and $DL(S)$ is the description length of the subgraph. The complexity of finding the normative subgraph is constrained to be polynomial by employing a bounded search when comparing two graphs. Previous results have shown that a quadratic bound is sufficient to accurately compare graphs in a variety of domains [13]. GBAD can discover three general categories of anomalies in a graph; insertions, modifications and deletions. Insertions would constitute the presence of an unexpected vertex or edge. Modifications would consist of an unexpected label on a vertex or edge. Deletions would constitute the unexpected absence of a vertex or edge. For more details regarding the

GBAD algorithms, the reader can refer to [13]. In summary, the key to the GBAD approach is that anomalies are discovered based upon small deviations from the norm - not outliers, which are based upon significant statistical deviations from the norm.

Table 3. Number of anomalies detected using $TD = 0.35$ based on graphs and datasets

Dataset	Anomalies	Graphs	
		Tweet Graph	Mixed Graph
World Cup	Keyword hijacking	0 (2)	2 (2)
	Bogus link	5 (20)	20 (20)
	Total	**5 (22)**	**22 (22)**
NATO Summit	Keyword hijacking	–	–
	Bogus link	5(11)	11(11)
	Total	**5 (11)**	**11 (11)**

5 Experimentation and Results

The experimentation process involves running the GBAD tool[2] on the *Tweet Graph* and *Mixed Graph* constructed from both FIFA world cup and NATO summit datasets. GBAD uses a compression technique to discover the normative patterns in the graphs; the normative patterns are then used to identify the anomalous structures. In other words, GBAD analyzes the complete dataset through the lens of the selected normative patterns in order to discover the anomalies - and in our case, the spam. We select $TD = 0.35$ as our anomaly detection threshold for GBAD. Table 3 shows the number of anomalies detected by the proposed approach using the normative patterns shown in Figs. 4 and 7 for the World Cup and NATO datasets respectively. We now discuss the results of anomaly detection on each of the trending topics.

5.1 FIFA World Cup

The normative pattern reported by GBAD on the FIFA world cup dataset for *Tweet* and *Mixed Graph* is shown in Fig. 4. The normative pattern for the *Tweet Graph* (Fig. 4(a)) indicates that the tweet was posted by a Twitter handle on July 2018 and has a hashtag of "worldcup" and mentions another twitter handle. Using this normative pattern on *Tweet Graph*, we were able to discover only 5 anomalies (out of 22 anomalies) present on this dataset. The fewer number of anomalies is reported by the *Tweet Graph* because the graph constructed using just the text of the tweet is not able to provide enough context. Therefore, we will extend the context by applying extra information from the news associated

[2] www.gbad.info.

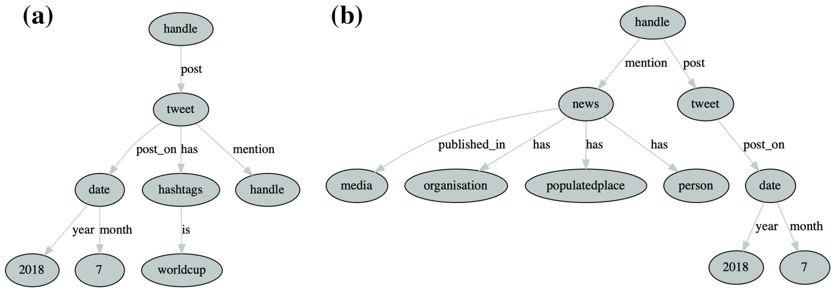


Fig. 4. Normative pattern on World Cup Dataset (a) Tweet Graph (b) Mixed Graph

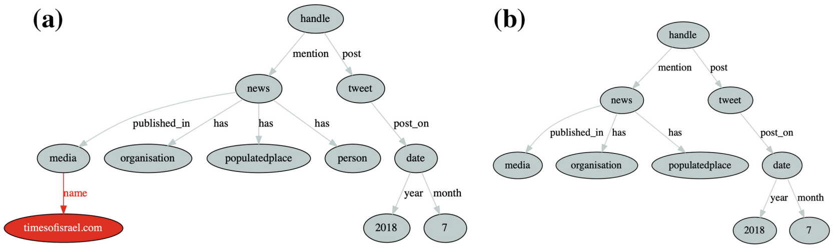
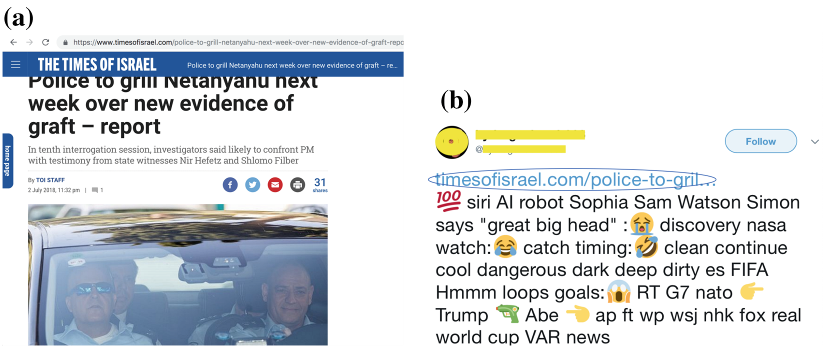


Fig. 5. Anomalous patterns reported in mixed graph (a) addition of node (b) deletion of node *person* in FIFA world cup dataset


(a)
https://www.timesofisrael.com/police-to-grill-netanyahu-next-week-over-new-evidence-of-graft-repo
THE TIMES OF ISRAEL
Police to grill Netanyahu next week over new evidence of graft – re...
home page
31 shares
(b)
Follow

Police to grill Netanyahu next week over new evidence of graft – report

In tenth interrogation session, investigators said likely to confront PM with testimony from state witnesses Nir Hefetz and Shlomo Filber

By TOI STAFF
2 July 2018, 11:32 pm | 1

[redacted]
@[redacted]

timesofisrael.com/police-to-gril...
💯 siri AI robot Sophia Sam Watson Simon says "great big head" :😭 discovery nasa watch:😂 catch timing:🤣 clean continue cool dangerous dark deep dirty es FIFA Hmmm loops goals:😱 RT G7 nato 👉 Trump 🔫 Abe 👈 ap ft wp wsj nhk fox real world cup VAR news

Fig. 6. Example screen-shot of (a) news and (b) tweet showing keyword hijacking anomaly in world cup dataset

with the tweet in *Mixed Graph*. This will allow us to identify more informative normative patterns and anomalies.

The normative pattern for the *Mixed Graph* (Fig. 4(b)) indicates that a tweet posted by a Twitter handle in July of 2018 mentions news published in some media and has a person, an organization, and a populated place. This normative pattern provides extra information about the news besides tweet information. Using this normative pattern on the *Mixed Graph* we discovered several types of anomalies. For example, the anomaly represented by Fig. 5(a) has an extra node *"timesofisrael.com"* (shown as a red node) linked to a *"media"* node by an edge *"name"*. In this particular case, GBAD reported the anomaly because the tweet has the keywords "FIFA" and "WorldCup" but the link posted has news not related to the world cup - instead it was political news from Israel. This is an example of a keyword hijacking anomaly that we introduced earlier. The screen shot of the tweet and news related to this anomaly is shown in Fig. 6. Similarly, another instance of an anomaly (as shown in Fig. 5(b)) has a *"person"* node missing from the normative patterns. Upon inspection, we find that the tweet had two URLs; one was a legitimate URL that has real world cup news, while the other was a link to a website that was not related to the World Cup.

5.2 NATO Summit

Figure 7 shows the normative pattern discovered by GBAD on the *Tweet* and *Mixed Graph* for the NATO dataset. The normative pattern for the *Tweet Graph* (as shown in Fig. 7(a)) indicates that a twitter handle posted a tweet in July of 2018 that has a person, an organization, and another twitter handle mentioned. Using it for anomaly detection on the *Tweet Graph*, we are able to discover only 5 anomalies (out of a total of 11 anomalies) on the NATO dataset. The sub-graph representing these 5 anomalies present in *Tweet Graph* is shown in Fig. 8(a) where the label of the node *"person"* in terms of the normative pattern is modified into a label *"hashtags"* (represented by a red node).

However, the normative pattern discovered by GBAD on the *Mixed Graph* on the NATO dataset (as shown in Fig. 7(b)) provides extra information about news besides the information from the tweets. This normative pattern indicates that a tweet posted by a Twitter handle in July of 2018 mentions news published in some media that has a person "Donald Trump". Using this normative pattern for anomaly detection, we discover several anomalies. The anomalies represented by the anomalous sub-graph in Fig. 8(b) has nodes *"person"* and *"donald trump"* in the normative pattern modified to *"organization"* and *"company"* respectively (marked by the red nodes). Further inspecting the instances of the anomaly represented by this sub-graph, we discover that the tweet mentions two URLs. The second URL is the legitimate URL that refers to the news about the NATO summit, while the first URL links to a website called *"robinhood.com"* which talks about investing in the stock market. This is not related to the NATO summit. Hence, it is a bogus link (piggybacking) anomaly. The screen-shot of the tweet and the website referred by the URL is shown in Fig. 9.

Table 3 shows the numbers and types of anomalies detected by our proposed approach on both datasets. Running GBAD on just the *Tweet Graph*, we were able to detect 5 out of 22 anomalies, while all known 22 anomalies were discovered successfully using the *Mixed Graph* on the FIFA world cup dataset. Similarly, in the NATO summit dataset, we were able to detect 5 anomalies using the *Tweet Graph*, while all 11 anomalies were discovered using the *Mixed Graph*. More anomalies are discovered using *Mixed Graph* in both cases because we are able to extend the context by applying extra information from the news

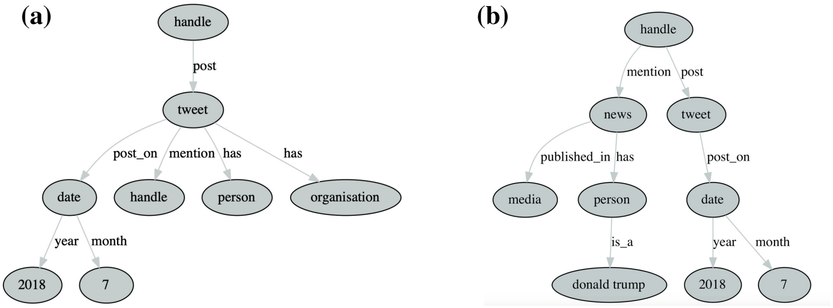
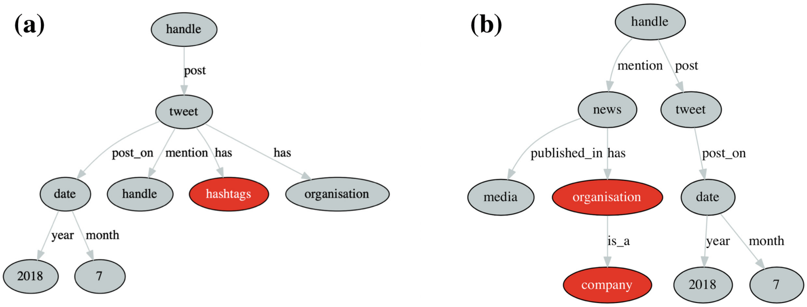
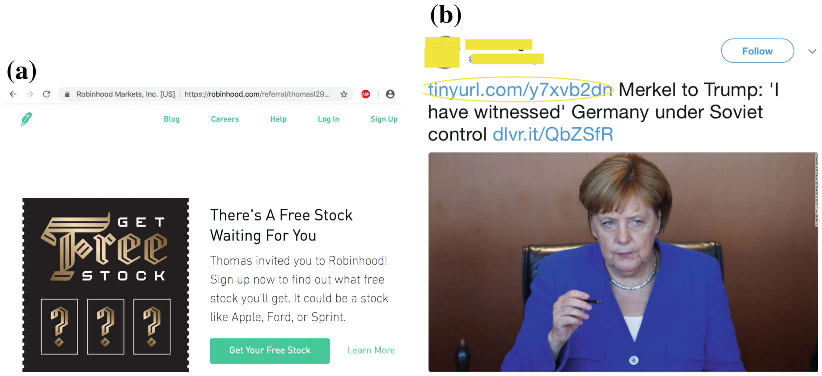


Fig. 7. Normative pattern on NATO Summit Dataset (a) Tweet Graph (b) Mixed Graph.




(a)
Robinhood Markets, Inc. [US] | https://robinhood.com/referral/thomasl29...
Blog
Careers
Help
Log In
Sign Up
GET
Free
STOCK
?
?
?

There's A Free Stock Waiting For You

Thomas invited you to Robinhood! Sign up now to find out what free stock you'll get. It could be a stock like Apple, Ford, or Sprint.

[redacted]

tinyurl.com/y7xvb2dn Merkel to Trump: 'I have witnessed' Germany under Soviet control dlvr.it/QbZSfR

Fig. 9. Example screen-shot of (a) news and (b) tweet showing bogus link anomaly in NATO summit dataset.

Fig. 8. Anomalous patterns on (a) Tweet Graph (b) Mixed Graph in NATO dataset.

associated with a tweet. The fact that we were able to detect both types of spam (keyword/hashtag hijacking and bogus link) using our approach supports our first hypothesis: "*Normal and anomalous behavior of a trending topic can be* *discovered using a graph-based tool on a graph representing named entities and their relationship*". Though we were able to detect a few anomalies in the *Tweet Graph* for both datasets, we are clearly more successful using the *Mixed Graph*, which represents using multiple sources. This is because the graph constructed using just the text of the tweet was not able to provide enough context. There are cases where the content of the tweet (represented by *Tweet graph*) looks normal, but after adding the information from the news, we can tell that the tweet was, in fact, an anomaly. For example, the tweet shown in Fig. 9 looks legitimate (couldn't detect it as an anomaly in *Tweet Graph*), but after adding the information referred by the tiny URL we are able to identify that the tweet was spam (i.e. is detected as an anomaly in the *Mixed graph*). This demonstrates our second hypothesis: "*Adding information from the document referred by the URL together with information from the tweet into the graph, anomalies that could not be detected using just the information from tweet can be detected*". Also, it is hard for the spammer to fabricate information in both the tweet and the document referred to by the URL in the tweet. The proposed graph-based approach exploits this type of dissimilarity in the content of the tweet and the document for anomaly detection.

6 Evaluation

In this section, we evaluate the performance of our graph-based approach. The dataset is highly imbalanced because spam tweets are very few in comparison to normal tweets. This challenges the interpretation of the standard evaluation metrics like accuracy or error rate [33–35]. For example, the FIFA world cup dataset has a class ratio of almost 99:1 and 99% accuracy is achievable by always predicting the majority class (i.e., normal tweets). Therefore, accuracy or error rate in this case does not provide adequate information on a classifier's functionality [33,35]. Other evaluation metrics that are frequently adopted in the research community to provide comprehensive assessments of imbalanced learning problems are precision, recall, and F1-score [33,35]. The focus of learning algorithms should be towards improving the recall, without sacrificing the precision. The F1-score incorporates both precision and recall, and the "goodness" of a learning algorithm for the minority class can be measured by the F1-score [35–37].

6.1 Baseline Approaches

In order to evaluate our approach, we compare our results against some well-known approaches that deal with the detection of spam tweets.

Benevenuto et al. [1] proposes an approach to detect spam tweets as well as the spam accounts using an SVM-based classifier. We compare our results with their spam tweet detection approach. The SVM classifier for spam tweet detection uses the following attributes for each tweet: number of words from a list of spam words, number of hashtags per words, number of URLs per words, number of words, number of characters in the text, number of characters that are numbers, number of URLs, number of hashtags, number of mentions, number of times the tweet has been replied, and if the tweet was posted as a reply.

Chen et al. [11] extract 12 user and content-based features and conduct experiments on different machine learning algorithms simulating various scenarios. They conclude random forest was the best approach. Account age, number of followers, followings, favorite users, list joined, and tweets posted are the user-based features. Similarly, the number of digits, characters, URLS, hashtags, mentions, and retweets on the tweet text are the content-based features. We compare the result of our approach against the performance of the random forest.

Anantharam et al. [12] proposes an approach to spot topically anomalous tweets in twitter streams by analyzing the content of the document pointed to by the URLs in the tweets in preference to their textual content. They manually identify reliable sources of information and compute the average cosine similarity Sim_{avg} amongst them. For every incoming tweet, they compute the cosine similarity between the document pointed to by the URL in the tweet and each of the trusted documents, then calculate the maximum cosine similarity Sim_{max}. If this maximum cosine similarity is less than the average cosine similarity among trusted documents (i.e. $Sim_{\text{max}} < Sim_{\text{avg}}$), the tweet is flagged as an anomaly.

Boididou et al. [18] proposes a semi-supervised framework that relies on two independent classification models built on the training data using tweet-based (TB) and user-based features (UB). Both models are built using a bagging technique. At prediction time, an agreement-based retraining strategy is employed (fusion) to combine the outputs of these two models (in a semi-supervised manner) which increases the generalization capabilities of the framework given tweets from a new unknown event. A corpus of labeled posts is necessary to build these classification models.

6.2 Discussion

We now present the performance evaluation of the proposed approach and the baseline approaches in terms of precision (P), recall (R), and F1-score. The performance score is shown in Table 4.

Using an anomaly detection threshold, $TD = 0.35$, without any other parameters, we are able to get a recall of 100% on both datasets. This indicates that the proposed graph-based approach can successfully detect spam tweets in trending topics. The higher recall is good for any anomaly detection problem, however, one also wants to achieve high precision or F1-score. The trade-off between precision and recall can be decided according to the need of our system. With some tradeoff on recall we are able to improve precision and F1-score by tuning GBAD

Table 4. Performance evaluation on both dataset

Approach	P	R	F1-score
World Cup Dataset			
Graph-based (non-parametric)	0.156	**1.0**	0.270
Graph-based (parametric)	0.516	**0.727**	**0.603**
Benevenuto et al. [1]	**0.799**	0.181	0.296
Chen et al. [11]	0.783	0.459	0.575
Anantharam et al. [12]	0.235	0.364	0.286
Boididou et al. [18]	0.692	0.409	0.514
NATO Summit Dataset			
Graph-based (non-parametric)	0.136	**1.0**	0.239
Graph-based (parametric)	**0.539**	**0.636**	**0.583**
Benevenuto et al. [1][a]	0.333	0.022	0.042
Chen et al. [11]	0.519	0.30	0.361
Anantharam et al. [12]	0.235	0.364	0.286
Boididou et al. [18]	0.375	0.272	0.315

[a] This approach was unable to classify spam on the NATO dataset, so we up-sampled spam tweets to make the spam to no-spam ratio ≈1:50

parameters[3]. Although the result of our parametric approach demonstrates the modest F1-score, the result can be easily modified based on our need (whether we need higher recall or higher precision). We present the result of both parametric as well as non-parametric approach in Table 4.

Our parametric approach has better performance in terms of recall and F1-score against all four baseline approaches on the FIFA World Cup dataset. Benevenuto et al. Chen et al. and Boididou et al. have better precision than our approach. However, the low recall and F1-score by these approaches indicates that our proposed graph-based approach has superior performance. Similarly, on the NATO summit dataset, our graph-based (parametric) approach has a better performance in terms of all metrics when comparing with all four baseline methods. Benevenuto et al. and Chen et al. use a machine learning approach that is more likely to be affected by the imbalanced nature of the data. Benevenuto et al. use a 1:3 ratio of spam to non-spam data in their original work while Chen et al. use 1:19. The dataset used in our experiment has a spam to non-spam ratio of ≈1:100. The low performance of these two approaches is because of their inflexibility in a highly imbalanced dataset. Also, Benevenuto et al. Chen et al. and Boididou et al. require a labeled dataset for training their classifier. However, our proposed graph-based approach is unsupervised and is not affected by the class imbalance problem. In fact, the performance of the proposed approach

[3] We use $maxAnomalousScore = 24$ for FIFA World Cup and $maxAnomalous Score = 52$ for NATO Summit datasets.

gets better in the data where the spam is rare (see definition of an anomaly in Sect. 4.4) which is usually the case in a Twitter trending topic. The approach by Anantharam et al. needs a predefined reliable information source for each topic, making it less flexible in real-time Twitter trends where new topics are evolving quickly. Also, the context generated (named entities and their relationships) from both the tweet and the document referred to by the URL in the tweet as used in *Mixed Graph* is hard to fabricate by the spammer. Furthermore, the better recall and F1-score in comparison to the existing approaches make our proposed unsupervised graph-based approach a solid approach for spam detection in a trending topic.

7 Conclusion and Future Work

In this work, we propose an unsupervised graph-based approach that leverages the relationship between the named entities present in the content of the tweet and the document referenced by the URL mentioned in the tweet for detecting possible spam. Our graph-based approach has superior performance in terms of recall and F1-score to that of existing approaches. Graphs provide a powerful machinery for effectively capturing the long-range correlations among interdependent data objects/entities. This interdependent nature of the data can be represented as an edge between each entity represented as nodes. We further claimed that a better understanding of patterns and anomalies associated with an entity like person, place, or activity, cannot be realized through a single information source, but better insight can be realized using multiple information sources simultaneously. We were able to discover new and unknown anomalies in the *Mixed Graph* which were not discovered in a *Tweet Graph*. We demonstrated this by collecting tweets relating to two different trending topics. Our approach focuses on the detection of spam tweets instead of the spam account. And the detection of the spam tweet can be useful for filtering spam in near real-time, while the detection of a spam account is about identifying such accounts retrospectively and blocking them so that they cannot spread future spam.

Although our research showed some good results, we were limited to static graphs. However, since the Twitter social graph can be extremely huge, graph construction and analysis can be time and resource consuming. If we think of a near real-time anomaly detection tool in social media or the web, it should be able to handle data that comes in streams. So, in the future we would like to focus on analyzing a near real-time feed by converting data streams into graph streams. Our experimentation was done on data sets of two trending topics, but we would like to extend it to more topics and test the robustness of the proposed approach. While our approach is not designed to discover anomalies in tweets that do not have URLs, discovering anomalies in tweets with URLs is key because it is reported that 90% of anomalous tweets contain unrelated or misleading URLs [1,3]. We would also like to use this approach for proactive criminal/terrorist activity detection by fusing multiple social media feeds. This is because we believe this anomaly detection technique will be able to track the change in behavior of an individual, particularly in terms of an individual's communications and transactions that may be represented in a graph.

References

1. Benevenuto, F., Magno, G., Rodrigues, T., Almeida, V.: Detecting spammers on twitter. In: Collaboration, Electronic Messaging, Anti-abuse and Spam Conference (CEAS), vol. 6, p. 12 (2010)
2. Martinez-Romo, J., Araujo, L.: Detecting malicious tweets in trending topics using a statistical analysis of language. Expert Syst. Appl. **40**(8), 2992–3000 (2013)
3. Gayo Avello, D., Brenes Martínez, D.J.: Overcoming spammers in twitter–a tale of five algorithms. In: Spanish Conference on Information Retrieval. CERI (2010)
4. Wang, A.H.: Don't follow me: spam detection in twitter. In: Proceedings of the 2010 International Conference on Security and Cryptography (SECRYPT), pp. 1–10. IEEE (2010)
5. Verma, M., Sofat, S.: Techniques to detect spammers in twitter-a survey. Int. J. Comput. Appl. **85**(10), 27–32 (2014)
6. Ameen, A.K., Kaya, B.: Detecting spammers in twitter network. Int. J. Appl. Math. Electron. Comput. **5**(4), 71–75 (2017)
7. Thomas, K., Grier, C., Ma, J., Paxson, V., Song, D.: Design and evaluation of a real-time URL spam filtering service. In: 2011 IEEE Symposium on Security and Privacy (SP), pp. 447–462. IEEE (2011)
8. Mccord, M., Chuah, M.: Spam detection on twitter using traditional classifiers. In: International Conference on Autonomic and Trusted Computing, pp. 175–186. Springer (2011)
9. Meda, C., Bisio, F., Gastaldo, P., Zunino, R.: A machine learning approach for twitter spammers detection. In: 2014 International Carnahan Conference on Security Technology (ICCST), pp. 1–6. IEEE (2014)
10. Song, J., Lee, S., Kim, J.: Spam filtering in twitter using sender-receiver relationship. In: International Workshop on Recent Advances in Intrusion Detection, pp. 301–317. Springer (2011)
11. Chen, C., Zhang, J., Chen, X., Xiang, Y., Zhou, W.: 6 million spam tweets: a large ground truth for timely twitter spam detection. In: 2015 IEEE International Conference on Communications (ICC), pp. 7065–7070. IEEE (2015)
12. Anantharam, P., Thirunarayan, K., Sheth, A.: Topical anomaly detection from twitter stream. In: Proceedings of the 4th Annual ACM Web Science Conference, pp. 11–14. ACM (2012)
13. Eberle, W., Holder, L.: Anomaly detection in data represented as graphs. Intell. Data Anal. **11**(6), 663–689 (2007)
14. Twitter: Report Spam on Twitter. https://help.twitter.com/en/safety-and-security/report-spam. Accessed 9 Oct 2018
15. Wu, T., Wen, S., Xiang, Y., Zhou, W.: Twitter spam detection: survey of new approaches and comparative study. Comput. Secur. **76**, 265–284 (2018)
16. Soman, S.J., Murugappan, S.: Detecting malicious tweets in trending topics using clustering and classification. In: 2014 International Conference on Recent Trends in Information Technology, pp. 1–6. IEEE (2014)

17. Wu, T., Liu, S., Zhang, J., Xiang, Y.: Twitter spam detection based on deep learning. In: Proceedings of the Australasian Computer Science Week Multiconference, p. 3. ACM (2017)
18. Boididou, C., Papadopoulos, S., Apostolidis, L., Kompatsiaris, Y.: Learning to detect misleading content on twitter. In: Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval, pp. 278–286. ACM (2017)
19. Yang, C., Harkreader, R., Gu, G.: Empirical evaluation and new design for fighting evolving twitter spammers. IEEE Trans. Inf. Forensics Secur. **8**(8), 1280–1293 (2013)
20. Lee, K., Caverlee, J., Webb, S.: Uncovering social spammers: social honeypots+machine learning. In: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 435–442. ACM (2010)
21. Lee, S., Kim, J.: WarningBird: a near real-time detection system for suspicious URLs in twitter stream. IEEE Trans. Dependable Secure Comput. **10**(3), 183–195 (2013)
22. Egele, M., Stringhini, G., Kruegel, C., Vigna, G.: COMPA: detecting compromised accounts on social networks. In: NDSS (2013)
23. Gupta, H., Jamal, M.S., Madisetty, S., Desarkar, M.S.: A framework for real-time spam detection in twitter. In: 2018 10th International Conference on Communication Systems & Networks (COMSNETS), pp. 380–383. IEEE (2018)
24. Akoglu, L., Tong, H., Koutra, D.: Graph based anomaly detection and description: a survey. Data Min. Knowl. Disc. **29**(3), 626–688 (2015)
25. Noble, C.C., Cook, D.J.: Graph-based anomaly detection. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 631–636. ACM (2003)
26. Paudel, R., Harlan, P., Eberle, W.: Detecting the onset of a network layer dos attack with a graph-based approach. In: FLAIRS Conference, pp. 38–43 (2019)
27. Chaparro, C., Eberle, W.: Detecting anomalies in mobile telecommunication networks using a graph based approach. In: FLAIRS Conference, pp. 410–415 (2015)
28. Paudel, R., Eberle, W., Holder, L.B.: Anomaly detection of elderly patient activities in smart homes using a graph-based approach. In: Proceedings of the 2018 International Conference on Data Science, pp. 163–169. CSREA (2018)
29. Paudel, R., Eberle, W., Talbert, D.: Detection of anomalous activity in diabetic patients using graph-based approach. In: FLAIRS Conference, pp. 423–428 (2017)
30. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. O'Reilly Media Inc., Sebastopol (2009)
31. Jurisica, I., Mylopoulos, J., Yu, E.: Ontologies for knowledge management: an information systems perspective. Knowl. Inf. Syst. **6**(4), 380–401 (2004)
32. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., et al.: DBpedia-a large-scale, multilingual knowledge base extracted from wikipedia. Semant. Web **6**(2), 167–195 (2015)
33. Sun, Y., Wong, A.K., Kamel, M.S.: Classification of imbalanced data: a review. Int. J. Pattern Recogn. Artif. Intell. **23**(04), 687–719 (2009)
34. Krawczyk, B.: Learning from imbalanced data: open challenges and future directions. Prog. Artif. Intell. **5**(4), 221–232 (2016)
35. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. **21**(9), 1263–1284 (2008)

36. Chawla, N.V., Lazarevic, A., Hall, L.O., Bowyer, K.W.: SMOTEBoost: improving prediction of the minority class in boosting. In: European Conference on Principles of Data Mining and Knowledge Discovery, pp. 107–119. Springer (2003)
37. Guo, H., Viktor, H.L.: Learning from imbalanced data sets with boosting and data generation: the DataBoost-IM approach. ACM SIGKDD Explor. Newslett. **6**(1), 30–39 (2004)