

On the Early Detection of Threats in the Real World Based on Open-Source Information on the Internet

Bouma Henri, Rajadell Olga, Worm Daniël, Versloot Corné, Wedemeijer Harry

Organization for Applied Scientific Research (TNO)
Oude Waalsdorperweg 63, 2597 AK The Hague, The Netherlands,
Tel: +31 888 66 4054, E-mail: henri.bouma@tno.nl

ABSTRACT

Many threats in the real world can be related to activities of persons on the internet. Internet surveillance aims to detect threats in an early stage and to assist in finding suspects based on information from the web. However, the amount of data on the internet rapidly increases and it is time consuming to monitor many open sources. In this paper, we present a method to find abnormal behavior on the internet and give an early warning for threats. The system was tested on Twitter data. The results show that it can successfully analyze the content of messages and recognize abnormal changes in sentiment and threatening content.

Keywords: Anomaly detection, internet surveillance, data mining, cyber security, pattern recognition.

1. INTRODUCTION

Many threats to people or infrastructures in the real world can be related to the activity of persons on social media, blogs and forums. While the exact causal relationship is yet largely unknown, internet surveillance aims to prevent possible attacks and assist in profiling suspects based on information from the cyber space. The information is gathered to improve situation awareness for the protection of citizens and critical infrastructures. However, the amount of data on the internet rapidly increases and it is time consuming to monitor the continuous flow of tweets, posts and announcements on websites.

In this paper, we present a novel method to automatically monitor trends and detect abnormal behavior on Twitter or other social media. Specifically, we propose a profiling system based on the analysis of a number of features, such as activity, sentiment, threat, demonstration, and grammatical tense. The list of features that analyze the content of messages can be easily modified and extended to meet the needs of the user. The system was tested on Twitter data. The results show that our system can successfully recognize abnormal changes in behavior based on sentiment and threatening content.

2. RELATED WORK

The disruption or destruction of certain infrastructures such as energy supply, drinking water supply, telecommunication, and various modes of transport may have serious impact on the health, safety, security or economic well-being of citizens or governments. Such infrastructures are therefore called critical infrastructures [25]. The threat of cascading failures across critical infrastructures has been identified as a key challenge for governments [18]. Security at vital infrastructures could be increased

by automated detection of deviant behaviors [9]. There are opportunities for social media (such as Facebook, Twitter and weblogs) to help the timely, comprehensive and transparent dissemination of information [21]. However, the automatic analysis of social media requires other methods than conventional text analysis. The social media provide huge amounts of visual data (photo or video), which can be analyzed [35] to use the information from multiple modalities and combine image with textual information for anomaly detection [7]. At the text retrieval conference TREC a special benchmark has appeared for the analysis of micro blogs. The overview of 2011 [29] showed that two systems performed very well: ISI [27] and FUB [4]. The ISI system is based on the full dependence variant of a Markov random field (MRF) using term, phrase and proximity-based text matching, and the FUB system is based on the Kullback-Leibler based product of information measures and size of the message, relative frequencies of terms and frequency for an additional unit of information. Furthermore, some ideas were tested to predict upcoming events in the future based on micro blog messages [42] and systems have been developed that select the most relevant information during big events and incidents (e.g., Twitcident [1], [39]). However, intuitive and flexible definition of new features for another context, and the automatic detection of relevant anomalies, remains a difficult task. In this paper, we share our findings in this area. Before we present our own method in Section 3, we give an overview of existing methods for sentiment analysis and anomaly detection in the remainder of this section.

Sentiment analysis

We use the term Twitter sentiment analysis to describe (automatic) methods for classifying the sentiment (emotions) of tweets: they can be positive, negative, or objective. In [40] a brief overview is given where three common sentiment analysis approaches are mentioned: Machine learning, lexicon-based methods and linguistic analysis. These approaches are described shortly, and after that available tools are mentioned that perform sentiment analysis.

The first type of sentiment analysis is based on machine learning. The machine-learning sentiment analysis requires a corpus: a set of tweets (or more general: texts) labeled positive, negative or objective. It can be created by hand, by labeling tweets manually. This will create a highly reliable lexicon, but also requires dozens of hours of work. Or a more clever automatic approach can be used by collecting hundreds of thousands of tweets and using happy and sad emoticons to label positive and negative messages [30]. Furthermore, newspaper accounts were used to collect objective messages. The labeled texts are then used to train an algorithm to detect features that associate related sentiment. Often-used features are *n-grams*, which is a sequence of n items (e.g., n letters or n words). So, a 3-gram can refer to the first three letters of a word, but also a word triple. Before reliable features can be extracted, often the text/tweet needs to be preprocessed. The following steps are suggested [30]: (1) filtering URL links and Twitter user names, (2) tokenizing the texts with punctuation marks and spaces, (3) removing stop words, and (4) constructing n -grams (sets of n subsequent words). After this extraction, Machine Learning methods can be used on the corpus (training data), such as SVM (support vector machines), CRF (conditional random fields) or Naïve Bayes. The Naïve Bayes appeared to work the best of these three mentioned methods [30].

The second type of sentiment analysis is based on a lexicon. The lexicon-based sentiment analysis requires a lexicon of labeled words (positive/negative/objective), possibly with weights of their intensity, and use the occurrence of these words in texts to compute the sentiment value of the text. See e.g. [36].

The third type of sentiment analysis is based on a linguistic approach. A linguistic analysis exploits the grammatical structure of the text to predict its polarity, often in conjunction with a lexicon. For instance, linguistic algorithms may attempt to identify context, negations, POS (part-of-speech, i.e. noun, verb, etc.) as part of the classification process. One potential problem with linguistic analysis for the microblog messages is the fact that most tweets are not grammatically correct – having abbreviations, incomplete sentences etc. – due to the shortness of the message. But perhaps a Twitter-specific ‘grammar’ could be constructed. Often sentiment tools combine several different approaches.

Many English sentiment tools exist on the web, including tools specifically primed for Twitter. Captico [10] shows an overview of tools that can track twitter sentiment, including: Twitter search (at twitter.com), Twitter sentiment (at appspot.com), Social Mention, Sysomos, Tweetzi, Twendz, Statsit, Twitrratr, Evri, AnalyzeWords, and MoodViews. However, it is difficult to use the English tools for Dutch messages. One possible approach would be to (automatically) translate the Dutch input data to English, and apply the English tool to the translation (e.g., by using the API of Google Translate). However, there are several tools that can directly be applied to Dutch tweets. Three are listed below.

The Pattern software [14] uses a parser together with a lexicon, thus would fall in the linguistic analysis-class. ‘Pattern’ is a web mining module for Python, that can also handle Dutch texts. It can retrieve text from the internet (from Twitter and other sites), analyze texts (using a rule-based shallow parser, containing both a Dutch and English version) and perform sentiment analysis (containing also a Dutch lexicon with a few thousand positive and negative words). A shallow parser analyzes a sentence and identifies the constituents (nouns, verbs, etc.).

Lucene [5],[19] is based on search engine technology so it does not train/model different classes as most machine learning techniques would do. A training set is indexed and new tweets are used as a query to search this index (i.e. the search engine does the actual text comparison). A form of majority voting was done on the results of this query. If the results are mainly positive the tool concludes the new tweet is also positive. Internally some smoothing and normalization is used.

The Lingpipe text classifier is an implementation of [11], [3]. The actual classifier is based on [31]. The Lingpipe implementation optimizes the (letter) *n-gram* value, which indicates the maximum allowed length of words. A dataset can be provided to train the classifier and optimize the parameters.

Violence or threat analysis

In the field of text mining, several attempts have been made to predict or classify threatening or offensive texts. In [12] methods are described for detecting offensive languages in social media, using weighted sums of the number of pre-specified weak and strong offensive words, combined with text mining techniques also used in sentiment analysis like Bag-of-Words, n-grams, Appraisal Approach. They also try to classify users as being offensive. In [22] text mining approaches for detecting cybercrime like internet predation and cyber bullying are discussed, both via purely statistical approaches and rule-based approaches.

In [28] data mining techniques have been applied to detect suspicious emails, i.e. emails alerting of upcoming terrorist events, using applied decision tree, Naïve Bayes, logistic regression and SVM algorithms, emphasizing initially on creating the feature space, and then applying different feature selection techniques. In [6] also threatening emails are being classified, in a supervised learning setting using Decision Tree, which seems to outperform methods like SVM and Naïve Bayes.

Anomaly detection for early threat detection

Anomaly detection schemes have been used in the detection of network intrusions. In this case the aim is to detect perturbations of normal behavior, that suggest the presence of attacks or faults. For such a purpose, several anomaly and outlier detection schemes have been proposed for detecting novel attacks whose nature is unknown. Often, the procedure needs to extract useful characteristic features as well as standard metrics. In this case the problem does not differ to the case of anomaly detection in Twitter data. We aim at applying techniques known for network intrusion in the field of social network data analysis by using some other more appropriate features together with standard metrics.

Data mining based intrusion detection techniques are used for misuse detection and anomaly detection. In general, most anomaly detection approaches, build models of normal data and detect deviations from the normal model in observed data. A statistical description of the normal training data is created and for new data hypothesis testing is applied [15]. In this problem, given a set of normal training data and a new piece of test data, the goal of the anomaly detection algorithm is to determine whether the test data belong to “normal” or to an anomalous behavior [16], [20]. This problem is referred to as supervised anomaly detection, since the models are built only according to the normal behavior on the network. There are several different non-parametric test statistics and distances capable of detecting change points in the data distribution, like the CUSUM and the Kolmogorov-Smirnov test [8], and the Kullback-Leibler divergence. However, the application of such methods for multidimensional data is not straight forward. On the contrary, classifiers are pattern recognition tools that have proven to be successful in describing data in multidimensional spaces. Here, we consider the problem as a classification or outlier search problem where the data is described with features, like this it can be tackled from different angles and the wide range of machine learning techniques can be used [23].

3. METHOD

Overview of the method

The complete system consists of the components listed in Figure 1, which is similar to the approach of [7]. Each of the main components is explained in more detail.

Data collection

To develop and validate on relevant data, a tool was made that collects both the historic messages and real-time streams of Twitter-data. The historic data allows the analysis of an incident after it occurred and helps to see in hindsight whether it could have been predicted. This data is important for research and development. The live streams have less restrictions are needed for early threat detection to give an alarm before the incident occurs. The messages are cleaned before enrichment, training and classification, by applying a preprocessing step that consists of the following:

- All rt, Rt, RT (retweets) are removed.
- All urls are removed.
- All ‘@username’ are removed.

- Smileys are replaced by Dutch words: ‘blij’(=happy) or ‘boos’ (=angry). This was replaced because punctuation may be removed or ignored in the subsequent processing steps.
- The keyword ‘doodsbedreiging’ (= death threat) is removed from all tweets because it was used to create the database with threats.

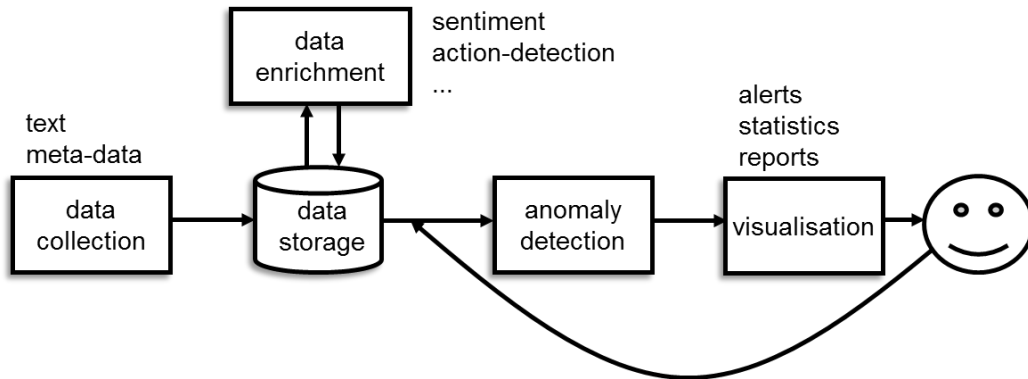


Figure 1. Overview of system architecture.

Data enrichment

For each minute, we computed the following features: activity (number of tweets), positive sentiment, negative sentiment, neutral sentiment, threat, non-threat, demonstration, non-demonstration, past tense, present tense, future tense. For sentiment, threat and demonstration features we computed both the absolute values (which correlate with activity) and the normalized values (which are invariant to activity). Furthermore, we compute the change of sentiment, threat and demonstration by computing the gradient. Each feature was filtered with a causal filtering method to suppress noise and empty minutes were filled based on previous tweets. In this subsection, we describe in more detail the features: sentiment, threat, demonstration, and tense.

3.1.1. Sentiment feature

The sentiment feature distinguishes between positive, neutral and negative sentiment. For sentiment analysis we selected three tools: Pattern [14], Lucene [5] and Lingpipe [3]. For Lingpipe, we used a maximum n-gram value of 5 letters. For Lucene and Lingpipe, we used the dataset of Tromp [41], and Pattern used its own lexicon for sentiment analysis.

3.1.2. Threat and demonstration feature

The threat feature and the demonstration feature are both related to violence. The threat feature distinguishes between (death) threats and non-threats and the demonstration feature distinguishes between protests/demonstrations and non-demonstrations. For the threat and demonstration features, we used the same methodology (Lingpipe) as for the sentiment feature, but different datasets. The neutral (non-threat / non-demonstration) dataset was created by selecting equal portions of positive, neutral and negative sentiment messages from the Tromp-dataset [41]. The threat dataset was scraped

from the website www.doodsbedreiging.nl, where thousands of Dutch tweets containing death threats – posted in the last few years – have been collected. The demonstration dataset was created by scraping tweets with related keywords.

- Demonstration: haat, haten, vermoord, uitroeien, honden, protesteert, protesteren, protestatie, protest, protesterende, protesteert, demonstratie, demonstreren, wij gaan ook, schreeuw, schreeuwen, kut, fok, shit, vlag, vlaggen, wie gaan zometeen, tegen, neuk, kkveel, gevecht, kanker, stinken, stoet, marcheert, ruzie, roept, dood, rellen, rellende, politie, plisie, politieauto, no go, gedoe, loopt uit de hand, woedende, overnemen, ruzie, aangevallen, aanvallen, vechtpartij, vechten, rennen, uitvechten, uitgevochten, geslagen, gewond, gewonden, vernieling, vernielingen, klimmen, schade, fik, steken, slaags, aanhoudingen, gaan aanvallen, tegen, zielig, alarm, dader, tuig, onbeschoft, criminaliteit, anti, heling, agent, mishandeling, opsporen, geweld, dreiging, bedreiging, doodsbedreiging, dood, moord, vermoord, fraude, politieman, brand, incident, criminaliteit, groeiend verzet, sharia4belgium, achterlijk, komtallen, verhefjestem, evenement, aanslagen, negativiteit, irritant, oorlog, overprikkeld.

3.1.3. Grammatical tense feature

The grammatical tense features distinguishes between past, present and future. The tense is important for early threat detection because it is a predication of violence in the (near) future and not a reflection on activity in the past. For the grammatical tense features, we used Lingpipe – as for the sentiment feature – but with a different database. The training database is created by a scraping tweets with related keywords. We perform a scrape to expand our short list of query keywords to make the tense feature more reliable.

- Future (before event): gaan, gaat, morgen, overmorgen, kom, wil, willen, wordt, word, worden, zal, zul, zullen, proberen, zo, zodadelijk, zometeen, oproepen.
- Present (during event): vandaag, nu, ondertussen, is, hier, zie, hoor, staan, staat, loopt, lopen, zitten, proberen, zijn, moeten.
- Past (after event): gisteren, gister, eergisteren, hebben, had, heeft, gehad, waren, was, geprobeerd, gingen, gelopen, stonden, liep, liepen, zat, zaten, kwam, kwamen, zijn, moesten.

Anomaly detection

From the known pattern recognition techniques available, we have selected one-class classifiers (OCC) [38]. These classifiers model one single class, called the target class, and discern between what is similar to it and what deviates from it, called outliers. Because we face a problem where we know what is normal but not what it is abnormal, this type of tools fit the definition of our problem. The normal class is our target class and we aim to detect all outliers, that is, anomalies. Consequently, we do not need to model or define the anomalies.

Among the OCC known, we used a Gaussian based one-class classifier (*gauss_dd*) from PRTools [17] for anomaly detection. This classifier computes the mean and standard deviation for each feature in the training data to describe the distribution of the training data. Then the classifier computes the Mahalanobis distance (considering all features) from all samples within the training to the mean of the distribution. The maximum of all those distances is chosen as threshold. Then when a new sample is tested, the Mahalanobis distance from the features of the new sample to the distribution of the training

data is calculated. If such a distance is greater than the threshold, the new sample is an anomaly (outlier), otherwise it is normal.

The only thing that needs to be defined is the concept of normal, which means selecting what the training data is for the classifier. New data that does not lie within the same distribution of the normal training data, will be detected as anomaly. Nevertheless, the features are context dependent, what is normal in one context (trend or topic) may be abnormal in another and vice versa. Therefore, the classifier should always be trained with a normal portion in each context in order to detect changes in trends and anomalies in that specific context. Furthermore, not every deviation from the normal pattern may be interesting for the user. For example, a positive remark may be irrelevant while increasing threat is important, depending on the needs of the user.

Notice that the scheme is targeted at monitoring trends. So analyzing novel upcoming data requires data from the past. That historic data is used for training (normal) and the following processing steps raise an alarm on upcoming data. The classification scheme uses some features to classify anomalies and some other as post-processing to filter the output of the classification.

- Pre-processing: The inactive periods (minutes with 0 tweets) are removed from the training set. We focus on the classification of the content of messages and ignore inactive periods. Content-based features cannot reliably be measured in empty minutes, due to the lack of content itself. Besides, to assure analysis of the content of messages and independency to activity, the activity features are not considered in the anomaly classification process. For applications where changes in activity are relevant, they could be included.
- Anomaly detection: The detector uses one-class classification based on the following content-based features: the sentiment, threat and demonstration. The detector combines an OCC on the absolute values and an OCC on the normalized values of these features with a union (OR) operation.
- Post-processing: Since not every anomaly is relevant, several post-processing conditions are applied. For early threat detection we require the tense to be future or present, the past is excluded. We also focus on the increase of negative sentiment, threat and demonstration (positive gradient). And the indication of a threat must be probable (probability > 0.5).

So, a minute is anomalous when its sentiment, threat and demonstration behavior do not fit the normal pattern and it is intended in the present or future. Furthermore, the negative sentiment, threat and demonstration confidence should be increasing and the threat over 0.5.

4. EXPERIMENTS AND RESULTS

Data sets

The Tromp-dataset [41] (approx. 1500 negative, 2100 neutral, 1500 positive) was used for sentiment analysis and additional data was gathered for threat (approx. 3300 threats, 2900 non threats), demonstration (3300 demonstration, 2900 non demonstration) and tense analysis (3000 past, 3000 present, 3000 future). The details of these datasets that were used for training and testing are described in Section 3. We applied our method on two datasets, related to two events in the Netherlands.

- Turkish-Kurdish data: This dataset describes the violent clash between Turkish and Kurdish people in Amsterdam in October 2011. Typical for this data is that every night, around midnight, sick (but hardly threatening) racial jokes are made about Turkish and Kurdish people. Thursday night (Oct. 20/21) the first call for a demonstration starts. Friday afternoon the demonstration occurs. Saturday afternoon there is another call for demonstration. Sunday afternoon is the biggest violent demonstration. On Monday and Tuesday, there were several calls for new aggressive activity on Wednesday.
- 4daagse data: This data describes the peaceful international four-day walking event in Nijmegen in July 2012. In this year, more than 38,000 participants walked four times 30, 40 or 50 kilometer. Every evening there are entertaining activities in the city center.

Results of sentiment and threat features

Both the sentiment and the threat classifier have been evaluated in the same way. The classifiers were 10 times trained on samples that were taken with 90% probability and tested on the remaining samples. The accuracy was used as a performance criterion. In general it is “#correct / #all”, e.g. for two classes:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN).$$

The average accuracy after 10 iterations with Lingpipe is 0.77 for sentiment classification and 0.96 for the threat classification. On sentiment classification, Lucene had similar performance and Pattern appeared to be 39% worse. On threat classification, Lucene appeared to be 10% worse than Lingpipe. Therefore, we reused Lingpipe for the demonstration and tense features. Below we show the confusion matrices of the Lingpipe classifier.

Table 1. Threat confusion matrix for Lingpipe

Threat confusion matrix	Lingpipe = 1	Lingpipe = -1
Ground truth = 1	307	16
Ground truth = -1	5	265

Table 2. Sentiment confusion matrix for Lingpipe

Sentiment confusion matrix	Lingpipe = 1	Lingpipe = 0	Lingpipe = -1
Ground truth = 1	86	17	27
Ground truth = 0	21	209	8
Ground truth = -1	37	13	101

Note that the positive sentiment is most often confused with negative and vice versa, and the neutral response has a much lower confusion. This may be caused by sarcasm. If we apply the Lingpipe and Pattern sentiment classifiers to the Turkish-Kurdish data, the following result is obtained.

Table 3. Sentiment of Lingpipe vs. Pattern

Sentiment	Lingpipe = 1	Lingpipe = 0	Lingpipe = -1
Pattern = 1	6.3 %	3.0 %	1.5%
Pattern = 0	9.0%	66.3 %	2.8%
Pattern = -1	5.3%	4.1 %	1.8 %

The tables also show that the two classifiers actually give opposite sentiments regularly, while they often seem to agree about the neutral messages.

Results of early anomalous threat detection

Our anomaly detection method is applied to the Turkish-Kurdish data and the 4daagse data. The initial 24 hours (not included in the plots) are used as training to learn the normal pattern in each context and the rest of the upcoming time is monitored to detect anomalies.

The results are shown in Figure 2, Figure 3 and Figure 4. Each of these plots show activity, the threat level and the demonstration confidence. In Figure 2 and Figure 3, all anomaly candidates (before post processing) are indicated in blue for both datasets. In Figure 4 the automatically detected threatening anomalies are shown (after post processing) for the Turkish-Kurdish data in yellow (present) and red (future). A similar plot for the 4daagse data was eliminated because threatening anomalies are not present in that dataset. An important observation is that the treat and demonstration features give a much higher response in the Turkish-Kurdish data than in the 4daagse data. The 4daagse data was expected to contains less messages related to threats, protests and demonstrations. Thus, it clearly indicates that the features respond correctly to these datasets.

The anomalies detected on the Turkish-Kurdish data match the information gathered on the news. We detect a future threat during Friday early morning that corresponds to tweets calling for demonstration. Then the demonstration itself took place during the afternoon of the same day which is detected as a present threat. Later there was a calling for demonstration again during Saturday (future threat) and the main burst of violence on Sunday (present threat). On Monday and Tuesday, the media pointed out the possibility of more quarrels the next days which is detected as a future threat by our scheme.

For the 4daagse case, no threats should be detected. Results show that anomalies because of deviation of the data are detected but they do not fulfill the treat definition in the post processing.

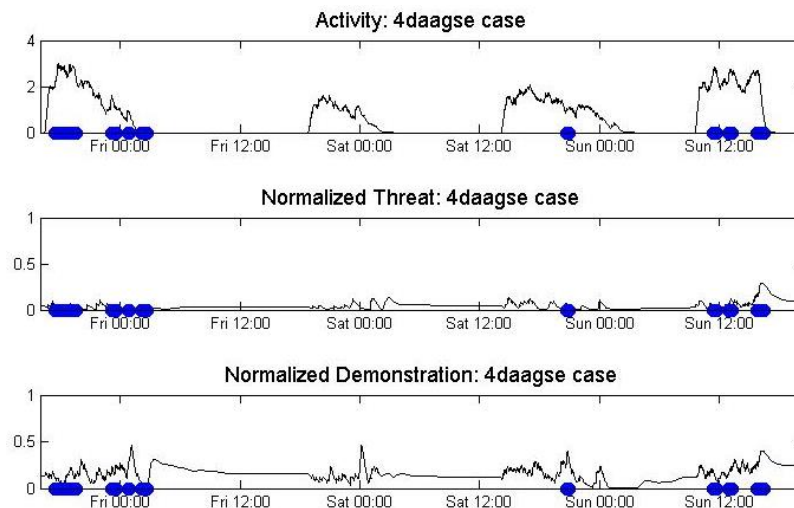


Figure 2. Anomaly candidates on 4daagse data.

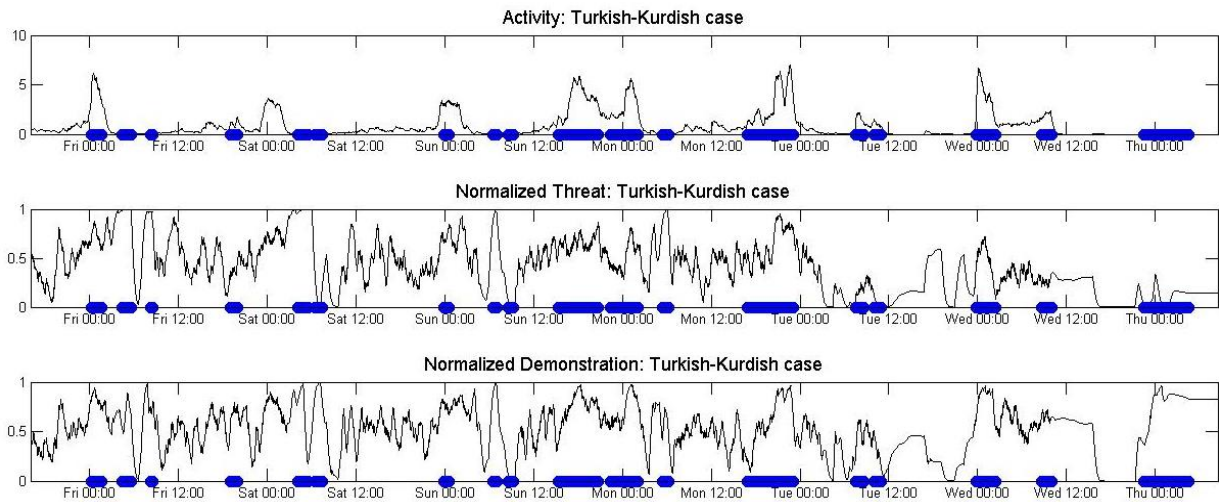


Figure 3. Anomaly candidates on Turkish-Kurdish data.

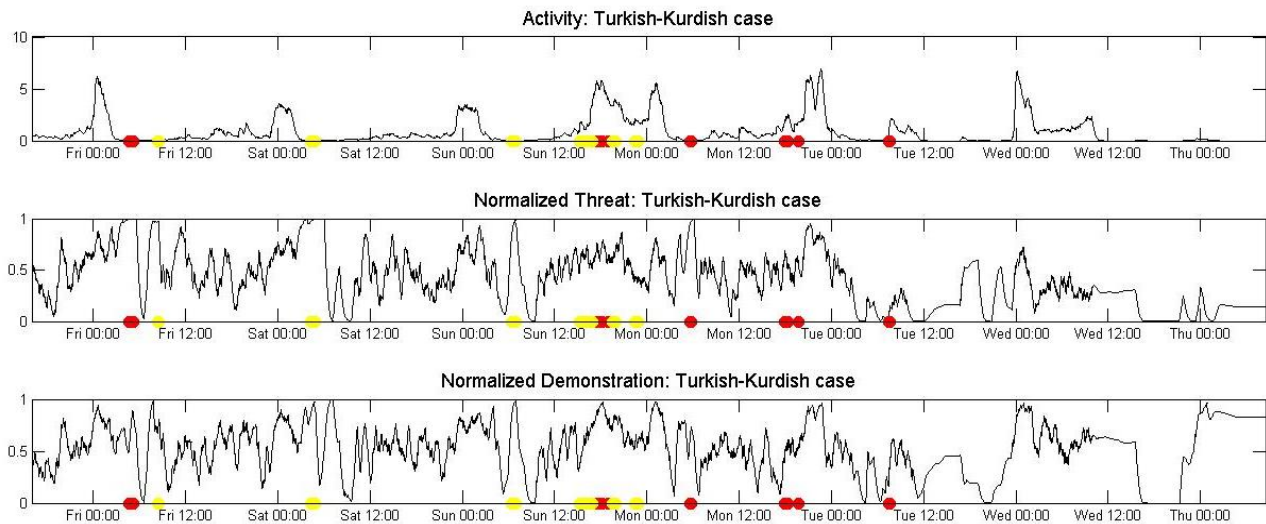


Figure 4. Automatically detected threatening anomalies on the Turkish-Kurdish data with future (red) and present (yellow).

5. CONCLUSION

In this paper, we presented a method that can automatically detect threatening abnormal activities in the real world based on information from the internet. We showed a way to define multiple features that analyze the content of messages, such as sentiment, threat, demonstration and tense. These features are trained on messages that were selected with a short list of query keywords. This list can easily be modified and extended to refine the existing features or to define new categories for another domain.

The anomaly detection separates normal behavior from deviant behavior to ignore a large portion of the data and draw attention to the abnormal changes. In combination with our post-processing steps, we are able to report threats and demonstration activities that are planned in the future, which allows the early detection of threats before they occur.

Future work will focus on the development of an interactive demonstrator to study the effects on overall performance for the user, to assess the strong and weak points in this system, and to get feedback from our stakeholders.

6. ACKNOWLEDGEMENT

The work for this paper was supported by the Dutch Ministry of Security and Justice in the program for safety and security research: “Veilige maatschappij” (project “Slimmer omgaan met informatie in de veiligheidsketen”). The authors thank the National Police Intelligence Service (IPOL) in The Netherlands for their help and advice.

REFERENCES

1. Abel F., Hauff C., Houben G., Stronkman R., Tao K. “Semantics + Filtering + Search = Twitcident. Exploring Information in Social Web Streams”, Int. Conf. Hypertext and Social Media, (2012).
2. Ahmed T., Coates M., Lakhina A. “Multivariate online anomaly detection using kernel recursive least squares”, IEEE Int. Conf. Computer Comm. INFOCOM, 625-633 (2007).
3. Alias-i “Lingpipe, toolkit for processing text using computational linguistics”, <http://alias-i.com/lingpipe>, Last accessed (2012).
4. Amati G., Amodeo G., Bianchi M. e.a. “FUB, IASI-CNR, UNIVAQ at TREC 2011 Microblog track”, TREC, (2011).
5. Apache “Lucene, indexing and search technology”, <http://lucene.apache.org>, Acc. (2012).
6. Appavu alias Balamurugan S. “Learning to classify threatening e-mail”, IEEE Int. Conf. Modeling and Simulation AICMS, 522-527 (2008).
7. Bouma H., Raaijmakers S., Halma A., Wedemeijer H. “Anomaly detection for internet surveillance”, Proc. SPIE vol. 8408, (2012).
8. Brodsky B., Darkhovsky B. “Non-parametric methods in change-point problems”, Kluwer Academic Publishers, (1993).
9. Burghouts G.J., Hollander R., Schutte e.a. “Increasing the security at vital infrastructures: automated detection of deviant behaviors”, Proc. SPIE 8019, (2011).
10. Captico “Tools for tracking twitter sentiment”, <http://captico.com/12-tools-for-tracking-twitter-sentiment/2010/08>, Last accessed: (2012).
11. Carpenter B. “LingPipe for 99.99% recall of gene mentions”, BioCreative workshop, (2007).
12. Chen Y., Zhu S., Zhou Y., Xu H. “Detecting offensive language in social media to protect adolescent online safety”, ASE/IEEE Int. Conf. Social Computing SocialCom, (2012).
13. Cheng H., Tan P., Potter C., Klooster S. “Detection and characterization of anomalies in multivariate time series”, SIAM Int. Conf. Data Mining, (2009).
14. Clips “Pattern, webmining module”, <http://www.clips.ua.ac.be/pages/pattern>, Acc. (2012).
15. Csorgo M., Horváth L. “Limit theorems in change-point analysis”, Wiley, (1997).
16. Denning D. “An intrusion-detection model”, IEEE T. Softw. Eng. 13(2), 222-232 (1987).
17. Duin R.P.W., Juszczak P., Paclik P., Pekalska E., de Ridder D., Tax D.M.J. “PRTools, A Matlab toolbox for pattern recognition”, <http://prtools.org>, TU Delft, (2004).
18. Eeten M. van, Nieuwenhuijs A., Luijff E., Klaver M., Cruz E. “The state and the threat of cascading failure across critical infrastructures: The implications of empirical evidence from media incident reports”, Public Administration 89(2), 381-400 (2011).
19. Hatcher E., Gospodnetic O. “Lucene in action”, Manning Publ., (2004).

20. Javitz H., Valdes A. "The NIDES statistical component: description and justification", Tech. Report SRI Intern., (1993).
21. Kleij R., Vries A., Faber W. "Opportunities for social media in the comprehensive approach", NATO RTO-MP-HFM-201, (2012).
22. Kontostathis A., Edwards L., Leatherman A. "Text mining and cybercrime", Chapter 8 in Text mining – Applications and Theory, (2010).
23. Lane T., Brodley C. "Detecting the abnormal: machine learning in computer security", Tech. Report Purdue Univ., (1997).
24. Li W., Eickhoff C., Vries A. "DMIR on microblog track 2011", TREC, (2011).
25. Luijff E. "Understanding cyber threats and vulnerabilities", Critical Infrastructure Protection LNCS 7130, 52-67 (2012).
26. Meij E., Weerkamp W., Rijke M. "Adding semantics to microblog posts", Dutch-Belgian Information Retrieval Workshop DIR, (2012).
27. Metzler D., Cai C. "USC/ISI at TREC 2011: Microblog Track", TREC, (2011).
28. Nizamani S., Memon N., Wiil U.K., Karampelas P. "Modeling suspicious email detection using enhanced feature selection", Int. J. Modeling and Optimization 2 (4), (2012).
29. Ounis I., Macdonald C., Lin J., Soboroff I. "Overview of the TREC-2011 microblog track", TREC, (2012).
30. Pak A., Paroubek P. "Twitter as a corpus for sentiment analysis and opinion mining", Proc. Conf. Language Resources and Evaluation LREC, (2010).
31. Pang B., Lee L. "A sentimental education: Sentiment analysis using subjectivity", Proc. ACL, 271-278 (2004).
32. Qiu P., Hawkins D. "A nonparametric multivariate cumulative sum procedure for detecting shifts in all directions", J. Royal Statistical Society: Series D - The Statistician, (2003).
33. Raaijmakers S., Versloot C., Wit J. "A cocktail approach to the VideoCLEF 09 linking task", CLEF LNCS 6242, 401-408 (2010).
34. Rijke M., Jijkoun V.B. e.a. "Generating, refining and using sentiment lexicons", Essential Speech and Language Techn. for Dutch, Springer, (2012).
35. Schavemaker J, Eendebak P., Staalduinen M., Kraaij W. "Notebook paper: TNO instance search submission 2011", Proc. TRECVID, (2011).
36. Taboada M., Brooke J., Tofiloski M., Voll K., Stede M. "Lexicon-based methods for sentiment analysis", Computational Linguistics, (2011).
37. Tao K., Abel F., Hauff C. "WISTUD at TREC 2011 microblog track", TREC, (2011).
38. Tax D. "One-class classification", PhD Thesis TU Delft, (2001).
39. Terpstra T., Vries A., Stronkman R., Paradies G. "Towards a realtime Twitter analysis during crises for operational crisis management", Proc. Int. Conf. Information Systems for Crisis Response and Management ISCRAM, (2012).
40. Thelwall M., Buckley K., Paltoglou G. "Sentiment in Twitter events", J. American Society for Information Science and Technology, (2011).
41. Tromp E. "Multilingual sentiment analysis on social media", M.Sc. Thesis TU Eindhoven The Netherlands, (2011).
42. Weerkamp W., Rijke M. "Activity prediction: a twitter-based exploration", SIGIR Workshop on Time-aware Information Access, (2012).