











CySecAlert: An Alert Generation System for Cyber Security Events Using Open Source Intelligence Data

Thea Riebe¹ , Tristan Wirth² , Markus Bayer¹ , Philipp Kühn¹ ,
Marc-André Kauffhold¹ , Volker Knauthe² , Stefan Guthe² ,
and Christian Reuter¹ 

¹ Science and Technology for Peace and Security (PEASEC),
Department of Computer Science, Technical University of Darmstadt,
Darmstadt, Germany

`riebe@peasec.tu-darmstadt.de`

² Interactive Graphics Systems Group, Technical University of Darmstadt,
Darmstadt, Germany

`tristan.wirth@gris.informatik.tu-darmstadt.de`

Abstract. Receiving relevant information on possible cyber threats, attacks, and data breaches in a timely manner is crucial for early response. The social media platform Twitter hosts an active cyber security community. Their activities are often monitored manually by security experts, such as Computer Emergency Response Teams (CERTs). We thus propose a Twitter-based alert generation system that issues alerts to a system operator as soon as new relevant cyber security related topics emerge. Thereby, our system allows us to monitor user accounts with significantly less workload. Our system applies a supervised classifier, based on active learning, that detects tweets containing relevant information. The results indicate that uncertainty sampling can reduce the amount of manual relevance classification effort and enhance the classifier performance substantially compared to random sampling. Our approach reduces the number of accounts and tweets that are needed for the classifier training, thus making the tool easily and rapidly adaptable to the specific context while also supporting data minimization for Open Source Intelligence (OSINT). Relevant tweets are clustered by a greedy stream clustering algorithm in order to identify significant events. The proposed system is able to work near real-time within the required 15-min time frame and detects up to 93.8% of relevant events with a false alert rate of 14.81%.

Keywords: Cyber security event detection · Twitter · Active learning · CERT

1 Introduction

Social Media has become a viable source for cyber security incident prevention and response, helping to gain situational awareness for Computer Emergency

© Springer Nature Switzerland AG 2021

D. Gao et al. (Eds.): ICICS 2021, LNCS 12918, pp. 429–446, 2021.

https://doi.org/10.1007/978-3-030-86890-1_24

Response Teams (CERTs). Therefore, the trend towards processing Social Media data in real-time to support emergency management [1] continues to grow. Husák et al. [2] show how Cyber Situational Awareness (CSA) is an adaptation of situational awareness to the cyber domain and supports operators to make strategic decisions. To perform such informed, situational decision-making, CERTs have to gain CSA by gathering and processing threat data from different closed and open sources [3]. These include Open Source Intelligence (OSINT), which uses any publicly available open source to accumulate relevant intelligence [4]. Especially the micro-blogging service Twitter has proven itself as a valuable source of OSINT due to its popularity among the cyber security community [5], as well as its available content and metadata for analysis [6]. Alves et al. [7] have shown that there is a small but impactful subset of vulnerabilities being discussed on Twitter before they are included into a vulnerability database. Increasingly big amounts of data make the use of more complex models possible. While concentrating on volume might be the best variable for some use cases, focusing on near real-time and data minimizing [8] approaches have been neglected in the recent state of research. Therefore, this paper seeks to answer the following main research question: **(RQ) How can relevant cyber security related events be detected automatically in near real-time based on Twitter data?**

By answering this research question the proposed paper aims to make the following contributions (C): The first contribution (C1) deducts the concept and presents the implementation of an automated near real-time alert generation system for cyber security events based on Twitter data (Sect. 2). The second contribution (C2) covers the evaluation of the *CySecAlert* system that assists CERTs with the detection of cyber security events in order to improve CSA by automatically generating alerts on the basis of Twitter data (Sect. 3). The near real-time capability is achieved by labelling and clustering the Twitter stream within the required 15-min time frame [9]. The third contribution (C3) provides a comparison of existing tools based on the systematic of Atafeh and Khreich [10] that are suitable to detect relevant cyber security related events based on Twitter data (Sect. 4). Lastly, the results are summed up (Sect. 5). To enable further improvement of our work, we will make the source code and the labelled Twitter dataset available.¹

2 Concept

This section presents the concept of *CySecAlert*, including the data source and architecture (Sect. 2.1), data preprocessing (Sect. 2.2), and training of the relevance classifier (Sect. 2.3) which serve as input to detect novel cyber security events (Sect. 2.4). It concludes with a concise description of the concept's implementation (Sect. 2.5).

¹ <https://github.com/PEASEC/CySecAlert>.

2.1 Data Source and Architecture

Twitter offers a multitude of advantages over other Social Media platforms. Firstly, Twitter is frequently used for the early discussion and disclosure of software vulnerabilities [7]. Secondly, Twitter accommodates a broad variety of participants, that are involved in the discourse evolving around cyber security topics. Since most important cyber security news feeds (e.g., NVD, ExploitDB, CVE) are present on the platform, Twitter serves as a cyber security news feed aggregate [11] and is used by both individuals and organisations [12]. In addition, tweets can be processed fast and easily [11], due to their limited length. Hasan et al. [13] propose a general framework for *Event Detection* systems. We added a relevance classifier to the architecture that filters out irrelevant tweets. By classifying relevance per tweet, the individual relevance of each tweet was determined before the clustering process, reducing the number of tweets at an early stage. This extension was necessary because our tweet retrieval method is account-based, leveraging preexisting lists of cyber security experts' Twitter accounts (see [Appendix A](#)).

2.2 Preprocessing and Representation

In a preprocessing step, we standardized the tweet representation by converting their content to a lower case and removing any textual part that is unlikely to contain relevant information, i.e., stop words, URLs, and Social Media specific terms and constructs (e.g. “tweet”, “retweet”, user name mentions) as well as non-alphanumeric characters. Then the text was tokenized and stemmed.

We applied a clustering-based approach to *Event Detection*. Therefore, a representation of individual tweets was necessary. To address this issue we adopted the setting of Kaufhold et al. [14], where a *Bag-of-Words* approach was applied. Clustering and classification were performed online. Therefore, the Inverse Document Frequency (IDF) regularization term would have had to be updated after every iteration, undermining the benefits of online techniques. In the context of crisis informatics, it has been suggested that the regularization via IDF does not necessarily yield a relevant benefit on classification performance [14]. Therefore, we omitted IDF regularization and represented tweets by Term Frequency (TF) vectorization only.

2.3 Relevance Classifier

To filter relevant tweets, we used an active learning approach [15], which has been found to reduce the amount of labelled data that is required to reach a certain accuracy level [16, 17]. We employed *uncertainty sampling* in order to obtain beneficial tweet samples for labeling. Therefore, we examined the suggestion of Kaufhold et al. [14] regarding rapid relevance classification. Lewis and Catlett [18] point out that it is reasonable to label the post which the current classifier instance is least confident about. Thus, the *Relevance Classification* is

performed by application of *pool-based sampling* with the *least confidence* metric. *Pool-based sampling* refers to an algorithm class that picks an optimal data point out of the set of non-labelled data points utilizing a metric that refers to the data's information content [16]. We applied the *least confidence* metric that regarded a data point as the most optimal labeling sample if the classifier was least confident about its classification [16]. Therefore, the datum with a prediction confidentiality closest to the decision boundary was selected.

Uncertainty sampling requires retraining of the classifier after every labeling process [18], which is not done in online learning. Kaufhold et al. [14] have shown, that this improvement in training time comes at the price of classifier accuracy, which can be addressed by using a fast online learning algorithm for the selection of data to be labelled, while batchwise creating a more sophisticated offline classifier with the same labelled data in parallel [18]. The combination of an incremental k Nearest Neighbor (kNN) classifier for *uncertainty sampling* and Random Forest (RF) is suggested to perform well on datasets in crisis informatics [14]. The Evaluation shows that this is true for the domain of cyber security as well (Sect. 3.2). Despite the increase of deep learning algorithms in this field, the utilization of classical machine learning algorithms suits best for this use case as the retraining can be performed automatically without the need for long training phases and specific training optimizations for every batch.

2.4 Detecting Events and Generating Alerts

Clustering based event detection approaches utilize vectorized representations of Social Media posts. In this scenario, every cluster represented a candidate event. We applied a simple greedy clustering algorithm that utilizes similarity metrics of new Social Media posts to old ones by considering them part of a new cluster if they exceeded a certain similarity threshold and otherwise adding them to the most similar preexisting cluster [19]. We performed the clustering based on nearest-neighbor search and used cosine similarity to the nearest cluster's centroids.

Alves et al. [11] propose a more sophisticated method that applies regular offline k-means clustering to improve the cluster quality. However, we chose not to do so as we put a special emphasis on near real-time applicability on our system. Furthermore, we justify the choice of relatively simple event detection techniques by the fact that the active learning approach for relevance classification in the cyber security *event detection* domain constitutes the core novelty of our contribution.

To *obtain significant events*, candidate events are filtered by their significance. Depending on the costs of alert processing and underlying costs regarding false alerts, it is reasonable to allow a system operator to configure the system's alert generation sensibility. *CySecAlert* supports the prediction of candidate events based on (1) overall post count associated with the event, (2) count of experts covering the event, and (3) the number of retweets.

The significance of candidate events based on the system operator's configuration was evaluated when a new tweet was added to the respective cluster. If

the cluster met the significance criteria and no alert had been issued based on the candidate event before, an alert was issued to the system operator. In order to assure the application's near-real-time capabilities tweets older than a certain time threshold (14 days by default) were removed from their respective cluster.

To *summarize events*, research suggests that textual clusters can be represented by display of their respective centroid [20,21]. We chose this event representation because it is cost-efficient and maintains the feeling of handling original Twitter data. We additionally allowed the display of the entirety of posts associated with an event to allow a system operator to further examine the event.

2.5 Implementation

CySecAlert was implemented in Java 11 and utilized a MongoDB database because of its high performance in handling textual documents. Figure 1 serves as an overview of the implementation's architecture.

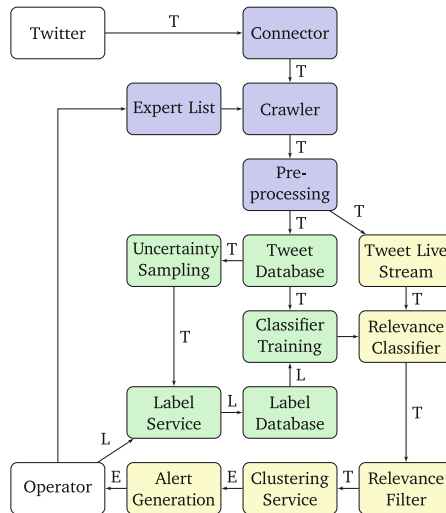


Fig. 1. Architecture of proposed Information and Communication Technology (ICT) illustrating the information flow for [T]weets, [L]abels and [E]vents. The ICT is divided into Tweet Retrieval (blue), Relevance Classifier Training (green) and Real-Time Event Detection (yellow). (Color figure online)

The Crawler module requested the most recent tweets of a list of trusted Twitter users in a regular manner. For this purpose, it used the Connector module. This functionality was implemented using Twitter4J². To train a relevance

² Twitter4J Version 4.0.7 (twitter4j.org/en/index.html on 14.08.2020).

classifier, it is necessary to manually label a set of tweets. The proposed application offers the use of active learners to reduce labeling effort. We evaluated an active batch RF, an active Naive Bayes, and an active kNN classifier. We used the classifier implementation of Weka³. A Relevance Classifier was trained based on the labelled data. The tweets to be labelled depended on the chosen sampling method. We chose an RF because its performance is well-proven in the context of Twitter Analysis, which was verified by qualitative evaluation. Our implementation utilized the Weka (See Footnote 3) implementation of an RF in its default configuration. The Relevance Classifier was used to filter out irrelevant tweets.

Then relevant tweets that covered the same topics were clustered to candidate events. This allowed an estimation of how much coverage a topic has on Twitter and helped to avoid alerts being used twice for the same topic. Therefore, we employed a greedy streaming clustering algorithm, which assigned each new tweet to the cluster with the most similar centroid according to the cosine similarity. If this similarity was smaller than a certain operator-defined threshold (*Similarity Threshold*) the tweet was designed to a new cluster.

A pre-evaluation has shown that the TF-IDF representation yielded performance benefits compared to the TF representation for the clustering task. Due to the sparsity of these vectors, we modeled them as *HashMaps*. Since classical IDF had to be updated after every added tweet, we stored the tweets in TF vectorized form and a centralized instance of IDF vector. The IDF regularization was applied on-demand if calculations required a vectorized representation. After every tweet insertion, the altered cluster was examined regarding its qualifications for an alert. Such a cluster was eligible for an alert if no alert had yet been issued for it and the count of unique tweets it contained exceeds a predefined threshold (*Alert Tweet Count Threshold*). The cosine similarity threshold and the tweet count threshold for the issuing of alerts were passed during program initialization.

3 Evaluation

This section presents the dataset (Sect. 3.1). The dataset is used to evaluate the active learning (Sect. 3.2), relevance classification (Sect. 3.2), alert generation (Sect. 3.3), system performance (Sect. 3.4), and near real-time capability (Sect. 3.5) of *CySecAlert*.

3.1 Dataset

We gathered 350,061 English tweets (151,861 tweets excl. retweets) published by 170 Twitter accounts of leading cyber security experts in the time period between 1st January 2019 and 31st July 2020. The list of accounts was derived based on a set of blog entries that provide lists of leading cyber security experts on Twitter (see [Appendix A](#), Table 4).

³ Weka v3.8.4(<https://www.cs.waikato.ac.nz/ml/weka/> on 14.08.2020).

Table 1. Class distribution over tweets of ground truth datasets.

	$S1$	$S2$
From	01/12/2019	01/05/2020
To	31/12/2019	14/05/2020
Irrelevant	5,801 (88.9%)	5,780 (85.25%)
Relevant	724 (11.10%)	1000 (14.25%)
Total	6,525	6,780
κ	0.9318	0.9377

In Relevance Classification, it is common to apply a binary classification into *relevant* and *irrelevant* tweets [11, 22, 23]. The class definitions of *relevant* and *not relevant* we applied are illustrated in a codebook (see Appendix B, Table 5) after Mayring [24].

Based on the dataset and the proposed annotation scheme, we created an annotated ground truth dataset consisting of two subsets ($S1$, $S2$) covering different time frames. The Datasets $S1$ and $S2$ were annotated by an additional researcher to estimate the inter-rater reliability of the coding scheme as shown in the codebook (Appendix B). Our ground truth shows a high level of inter-rater reliability ($\kappa > 0.90$) measured by Cohen’s kappa (κ). We used $S2$ for evaluation purposes. The class distributions of these datasets are illustrated in Table 1.

3.2 Relevance Classification

Sampling Method. We evaluated the influence of active learning and the selection of a sampling method and sampling classifier on the performance of a relevance classifier in order to choose a high-performing classifier. Therefore, we used the preprocessed and stemmed ground truth datasets $S1$ and $S2$. In this evaluation, a scenario was simulated where no labelled data is available initially. A virtual expert incrementally labelled tweets that were chosen by different sampling methods. The labels were taken from the respective ground-truth dataset. We examined a Naive Bayes classifier, a kNN classifier with $k = 50$ and an RF classifier. As *uncertainty sampling* technique we applied *least confidence* measure in a *pool-based sampling* scenario were examined.

While Naive Bayes and kNN can be implemented in an incremental manner and thus allow to add single tweets without retraining, the RF classifier did not offer this property. For this reason, kNN and Naive Bayes were updated after every new labelled tweet and the next uncertainty sampling step was performed on the updated classifier. In contrast, the RF classifier sampled a set of most uncertain tweets (rather than one) which were labelled as batches before being added to the training set. Thereafter, the classifier was retrained on the updated dataset.

An evaluation of the experiment (see Appendix C, Fig. 3) showed, that the active version of the Naive Bayes classifier performed worst, representing nearly

random classification behaviour. However, the kNN classifier was able to train a model whose AUC measure plateaus around roughly 0.75 for both datasets. This finding is similar to the results of Kaufhold et al. [14]. In contrast to them, we also considered active learning with an RF classifier. In our evaluation setting, it performed best with an AUC in the range of 0.9. Therefore, we choose a RF classifier for our system.

Classification Model. In this subsection, we analyse whether the use of a different active learning algorithm-based sampling method is useful for an RF relevance classifier. We compare (1) kNN and (2) batchwise RF uncertainty sampling with (3) random sampling and (4) batchwise Random-RF-Hybrid Sampling. This hybrid approach picks 50% of tweets per batch by RF-based uncertainty sampling and 50% tweets at random. By determining a threshold of Random Trees, which is needed to classify an instance as positive, a classifier is instantiated from the learned RF. In the context of this contribution, we chose the F_1 metric for evaluation purposes, as it is suitable for imbalanced datasets.

We evaluated the performance of the RF instances based on the F_1 measure of the classifier instance with the highest F_1 measure for every 100 labelled tweets. The evaluation was conducted by leaving out 1,000 tweets and using them as a test set. In order to mitigate performance issues, the uncertainty sampling was performed on a randomly chosen subsample of size 200 (500 for active batch RF), which changed in every iteration, rather than on the complete data pool. The results of this evaluation are illustrated in Fig. 2.

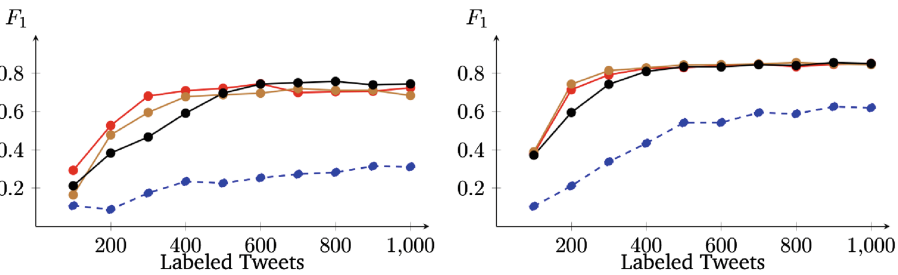


Fig. 2. Performance comparison of RF Classifier trained on dataset S1 (left) and S2 (right) with uncertainty sampling by different classifiers: Random (blue), RF Classifier (red), 50% RF and 50% Random (brown) and by kNN classifier with $k = 50$ (black). Average over 5 Executions using a 1,000 tweet holdout set measured in F_1 . (Color figure online)

The experimental results show that every examined type of uncertainty sampling leads to classifier out-performance compared to random sampling. For every experiment, the classifier instance that used a randomly sampled dataset was not able to achieve the performance of uncertainty sampled classifier with 300 or more labelled tweets, even if it was trained based on 1,000 randomly sampled

tweets. Furthermore, the results indicate that there are no significant performance differences between the tested uncertainty sampling classifiers.

Due to the fact that there are no substantial classification quality implications, we opted for the kNN based uncertainty sampling because it can be executed in an online manner. Additionally, the results suggest that the overall classification performance suffered for datasets with higher class imbalance. Nevertheless, the results indicate that after around 600 labelled tweets the classifier achieved its best classification quality and therefore did not show significant improvements for a bigger training dataset. This constituted a reduction of manual tweet annotation of up to 90% compared to a randomly sampled approach, which makes it necessary to label the whole dataset (roughly 6,000 tweets each).

3.3 Alert Generation

In this section, we jointly evaluate the clustering algorithm and the alert generation process. Therefore, we executed the combination of these modules using different parameters for *Similarity Threshold* and *Alert Tweet Count Threshold*. Even though there are multiple configurations for alert generation thresholds, the evaluation was performed based on the relevant tweet count per cluster metric only. Thereby, we received a list of clusters that represent a list of relevant events and their associated tweets. By comparing this list to the ground truth dataset (Sect. 3.1), the quality of the alert generation process could be estimated.

Therefore, clusters that were found by the clustering algorithm and flagged as alerts are classified as *topic related*, *mixed* or *duplicate*. A cluster was regarded as *topic related* if more than half of its tweets belong to the same topic of the ground truth topic list. If a *topic related* cluster that discussed this topic had been found before, the cluster was marked as *duplicate*. If there was no major topic in the cluster, it was defined as *mixed*. *Topic related* clusters were marked as positive, while *mixed* and *duplicate* clusters were marked as negative. Combining this information we derived a calculation for precision and recalled measures as follows:

$$Precision = \frac{\#truepositives}{\#truepositives + \#falsepositives} = \frac{\#topicrelated}{\#clusters} \quad (1)$$

$$Recall = \frac{\#truepositives}{\#truepositives + \#falsenegatives} = \frac{\#topicrelated}{\#topics} \quad (2)$$

In order to decouple the evaluation of clustering and alert generation from the performance of the relevance classifier, we tested the clustering-based alert generation algorithm on the set of *relevant* and *potentially relevant* tweets from our ground truth datasets $S1$ and $S2$. We used TF-IDF as tweet vectorization in order to avoid the formation of big clusters based on frequently used common words. The results show that an increase in the value of the used similarity threshold (in the observed range) decreases the recall (see Appendix D). Intuitively, this can be explained by the creation of more clusters due to similarity

failing the threshold. Therefore, clusters are smaller on average and stay under the alert generation threshold, which leads to suppression of alert generation for relevant topics. In contrast, the influence of similarity threshold on cluster precision (which is the invert of the wrongful alert quote) is lower. This is the reason why operators should be advised to prefer lower values for the Cosine Similarity Threshold. Even though this configuration increases the wrongful alert rate, it increases the recall. Nevertheless, if the similarity threshold is chosen too low, this does not hold. For example, a similarity threshold of 0 led to every tweet being part of one giant cluster. This led to a low recall as well. The alert generation instance with the best performance regarding the F1 score resulted in a precision of 96.08% and a recall of 96.23%.

Our experiment shows that the value of the *Cosine Similarity Threshold* leading to an optimal F1-measure depends on the *Alert Tweet Count Threshold*. Furthermore, the results indicate that minor changes in *Alert Tweet Count Threshold* have no significant effect on the Alert Generation System's performance. Comparing the best performing configurations for every examined *Alert Tweet Count Threshold* (similarity threshold of 0.3 for 3, similarity threshold of 0.25 for 5) shows that the performance differences are lower than 5%. Therefore, the system operator is advised to choose the *Alert Tweet Count Threshold* based on an alert frequency, that s/he is willing to process.

3.4 System Performance

This section examines the performance of the overall system combining Uncertainty Sampling, Relevance Classification, and Alert Generation. The evaluation is conducted based on the datasets *S1* and *S2*. After data preprocessing, an RF classifier was trained based on 600 tweets that were chosen by Uncertainty Sampling using a kNN classifier. Every tweet in the dataset that the resulting classifier deemed relevant was passed to the Alert Generation System which is configured according to the findings in Sect. 3.3: *Alert Tweet Count Threshold* = 5, *Cosine Similarity Threshold* = 0.25. The evaluation of the clusters was performed analogous to the procedure in Sect. 3.3 with *irrelevant* clusters as additional cluster class. A cluster was thereby considered *irrelevant* if it contained at least 50% tweets that are labelled as irrelevant. The experimental results (Table 2) suggest that the system is capable of detecting 90% of the events occurring in the ground truth data while 15% of reported alerts were not part of the ground truth data (false alert rate).

3.5 (Near-)Real-Time Capability

The run-time tests were performed on a computer with an *AMD Phenom II X6* CPU and 12 GB DDR3 RAM running Windows 10. We divided the alert generation system into two stages and measured their execution time separately: (TU1) the Relevance Classifier and (TU2) combining the clustering process with the alert generation process. We conducted the experiments using dataset *S1*. Since individual tweet frequency is highly volatile, we conducted our simulation

Table 2. Combined performance of relevance classifier, clustering algorithm and alert generation for datasets *S1* and *S2*.

Dataset	<i>S1</i>	<i>S2</i>
Precision	95%	85.19%
Recall	90.48%	93.88%
F_1	92.68%	89.32%

assuming the following worst-case scenario: Every user sends twice his/her average daily tweet count in the same one our frame: 2.5 Tweets per user per 15 min time-frame.

Sabottke et al. [9] suggest that the cyber security community on Twitter consists of about 32,000 accounts. Assuming that the system is used to issue alerts based on the tweets of 25% of these accounts, 20,000 have to be processed in a 15-min time frame in order to allow near real-time execution. Our experiments show that the execution of (TU1) takes 17.5s for 20,000 Tweets. Based on the class distribution, we determined in Sect. 3.1, $\approx 2,000$ of these tweets are going to be labelled as positive. Assuming that tweets that are older than 14 days are discarded, the clusters of the clustering service contain about 112,000 tweets at any time in this scenario. Extrapolation of the experiment on the execution time for the proposed clustering algorithm suggests that the clustering of 500 tweets takes about 210s in this case. That corresponds to around 840s (or 14min) for the given 2,000 tweets. Adding the execution times of (TU1) and (TU2) up shows that an execution in the given 15-min time frame is possible. An execution in a timely manner for more accounts or accounts that are more active is possible using a more powerful machine.

4 Related Work and Discussion

To use Twitter as an OSINT source for CERTs, we conducted a comparative analysis of existing tools and approaches which are suitable to complete this task (Sect. 4.1). Based on our contributions (Sect. 4.2), we identified limitations and potentials for future work (Sect. 4.3).

4.1 Cyber Security Event and Hot Topic Detection

Previous work has examined the possibilities of Twitter as an information source for cyber security event detection (overview in Table 3). As the techniques for event detection using Twitter differ, Atafeh and Khreich [10] offer a systematic approach that allows a comparison based on the of the necessary parts. Most previous work [12, 21–23, 25] examines the detection of generic cyber security threats. The majority of these publications [12, 21, 23] employs some kind of clustering algorithm on a Term Frequency-Inverse Document Frequency (TF-IDF) representation of single tweets compared by the cosine similarity distance.

Even though the publications’ core approach is related, they differ in details concerning the preprocessing of tweets and usage of the detected clusters. On closer inspection, most methodologies use human-generated input that serves as a filter for user-generated content and automatically expands these filters configuration by utilizing Twitter data [26]. These filters are either represented by lists of relevant keywords [26] or a set of credible experts [27]. To our knowledge, the scientific literature has not discussed the advantages and disadvantages of either approach extensively. This is especially true for the performance of machine learning algorithms on the respective databases. While a keyword-based retrieval approach is less prone to miss relevant tweets regarding a certain objective, it may attract a lot of tweets that contain a relevant keyword in a different semantic. Account-based approaches reduce the number of tweets that have to be processed and therefore reduce performance requirements for the underlying hardware. However, these accounts have to be known beforehand.

Table 3. An overview of event detection techniques with application to the cyber security domain, categorized by Retrieval Method (RM, [A]ccount-based or [K]eyword-based (* is filtering)), Detection Method (DM, [S]upervised or [U]nsupervised), as well as Pivot Technique (PT, [D]ocument- or [F]eature-based) and Detection Technique (DT) and Model, based on Atefeh and Khreich [10].

Work	RM		DM		PT		Application	DT	Model
	A	K	S	U	D	F			
[11]	✓	*	✓		✓		Summarization	CluStream, SVM, NN	TF-IDF
[23]		✓		✓		✓	Threats	DBSCAN	TF-IDF
[21]			✓	✓			Novel malware	Counting, K-Means	#, TF-IDF
[22]	✓	*	✓	✓	✓		Threats	NER by NN	Word Emb.
[28]	✓	*	✓	✓	✓		Threats	NER by MTL	Word Emb.
[29]		✓	✓		✓		Threat events	MTL	Word Emb.
[30]		✓	✓		✓		Cur. incidents	Prob. learning	TF
[31]		✓		✓		✓	Attacks	Clustering	Exp. queries
[27]	✓			✓	✓		Topics	Clustering	TF, Corr.
[26]		✓		✓	✓		Classification	Clustering	TF-IDF
[32]		✓		✓		✓	IT-Sec. alerts	Rule-based reason.	Graph(VKG)
[20]		✓	✓		✓		IT-Sec. events	Expect. Reg.	Diff. feat.
[25]	✓			✓	✓		Ident. Attacks	Term Filtering	TF
[33]		✓	✓		✓		Threat indicators	CNN-GRU	Random Emb.
[12]		✓		✓		✓	0-day exploits	K-Means	Documents
CySecAlert									
	✓	✓	✓		✓		IT-Sec Events	Rel. Filter, Clustering	TF-IDF

4.2 Contributions

For the **CySecAlert concept (C1)**, we opted for an account-based retrieval approach, that retrieves tweets based on a list of credible cyber security experts’ accounts. Active learning using uncertainty sampling has shown to be beneficial for training supervised classifiers with limited data in other domains [14, 16, 17, 34]. Literature of crisis informatics in combination with our evaluation suggests that an

incremental kNN classifier outperforms a Naive Bayes classifier and an active batch sampling version of an RF classifier if they are used as uncertainty sampling classifier for a batch RF classifier. Therefore, they allow high-quality classifiers with a smaller training set. This is valuable for the privacy by design principle of data minimization [8]. This means that fewer accounts and tweets are needed. In detail, our **evaluations (C2)** show that a training set containing only 600 tweets gathered by Uncertainty Sampling (10% of ground truth database) is suited to build a sufficient classifier. A classifier based on a training set consisting of 1,000 randomly sampled tweets is outperformed by a set of 200 uncertainty sampled tweets. The evaluation shows that *CySecAlert* scores a maximal F_1 measure of 92.68% (Precision: 95%, Recall: 90.48%) (Sect. 3.4). In **comparison to other approaches (C3)**, this exceeds the performance of Bose et al. [23] with an F_1 measure of 78.26% (Precision: 81.82%, Recall: 75%) and is comparable to the results of Dionísio et al. [28] with an F_1 measure of 95.1%, who have examined a related task. Although these papers are most comparable as they conduct similar experiments, a direct comparison of the evaluation results is nevertheless impractical because they refer to datasets of different time periods gathered from different sets of accounts. Regarding the real-time capability to our knowledge, only Le Sceller et al. [26] included a simple evaluation in their experiments. We extend the research in this direction as we perform a more in-depth analysis also incorporating the usage behavior. The near real-time of the system is not only supported by its capability to analyse the real-time Twitter stream [21, 25, 26], it also performs almost as fast as the SONAR system [26] (17.5 s for 20,000 tweets compared to 12 s).

4.3 Limitations and Future Work

As the *CySecAlert* system is designed to support CERTs, further improvements and evaluations as part of larger-scale incident monitoring are planned, such as the deployment on other social media platforms and longitudinal testing with larger datasets. The tests will include further studies regarding the security of the system against hacked or fake accounts as well as the risk of model poisoning. Further, controlled experiments will be conducted to exclude the impact of the dataset. Additionally, in recent times more sophisticated clustering algorithms have been proposed. For instance, Alves et al. [11] extends a greedy clustering approach by offline re-clustering if the cluster affiliation of a new tweet is unclear. This approach may be suited to avoid *duplicate* clusters in our clustering algorithm but may have a negative impact on the real-time properties. Furthermore, re-clustering, in general, interferes with the used online event selection process by changing cluster affiliation of past tweets. Future work should examine streaming clustering algorithms that are suited to enhance the proposed system’s overall performance without strongly influencing the capability of processing tweets of many users in a timely manner and the need for re-clustering.

Following the proposed system by Kaufhold et al. [14], we used the bag-of-word approach to represent text. However, recent contributions suggest that *Word Embeddings* can have relevant performance advantages over a multitude of other textual representation methods, including the bag of word approach

applied in this contribution [35]. Future research should examine if the application of Word Embeddings is suited to further improve the proposed alert generation system’s performance without the negative influence of the system’s timing constraints. Furthermore, NNs in general and in the domain of cyber security related event detection enjoy increasing popularity and show high performance in relevance classification tasks [22]. While the current state of the system with its real-time, low-resource, and robust applicability is only suited for classical machine learning algorithms, future work should examine the influence of different uncertainty sampling classifiers on the performance of NNs as relevance classifiers.

5 Conclusion

This work proposes a framework for timely detection of novel and relevant cyber security related events based on data from the social media platform Twitter (*CySecAlert*). *CySecAlert* is capable of collecting tweets based on a list of trusted user accounts, filtering them by relevance, dividing them into clusters by topic similarity, and issuing alerts if one such topic surpasses a predefined significance threshold. The system further aims to support data minimization for OSINT by focussing on a network of expert accounts. Further, it is easy for an expert community, such as CERTs, to adopt as well as quick to train with little labelling and runs in near real-time. Our study based on manually labelled ground truth data shows that the amount of labelled data to train a classifier can be substantially reduced by the application of uncertainty sampling for training set generation in contrast to random sampling. The proposed classifier achieves a precision of 87.18% and a recall of 84.12%, while the cluster-based alert generation subsystem achieves a false alert rate of 3.77% and detects 96.08% of relevant events in the ground truth dataset. An evaluation of the overall system shows that it is able to detect up to 93.88% of relevant events in a ground truth dataset with a false alert rate of 14.81%.

Acknowledgements. This work was supported by the German Federal Ministry for Education and Research (BMBF) in the projects CYWARN (13N15407) and KontiKat (13N14351), as well as by the BMBF and the Hessian Ministry of Higher Education, Research, Science and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE. We would like to thank the anonymous reviewers for their valuable and constructive comments.

Appendix A Dataset

Table 4 provides the websites and blogs we used to retrieve 170 accounts of the leading cyber security experts on Twitter, from which we gathered the dataset of 350,061 English tweets (see Sect. 3.1).

Table 4. Sources for cyber security experts on Twitter

List of security expert sources
The top 25 infosec leaders to follow on Twitter ^a
Top 15 security experts to follow on Twitter in 2018 ^b
Best cyber security Twitter profiles to follow 2018 ^c
100 security experts you could follow on Twitter ^d
10 cybersecurity Twitter profiles to watch ^e
21 cyber security Twitter accounts you should be following ^f
^a techbeacon.com/security/top-25-infosec-leaders-follow-twitter/ , accessed 2021-07-08
^b resources.whitesourcesoftware.com/blog-whitesource/top-15-security-experts-to-follow-on-twitter-in-2018/ , accessed 08.07.2021
^c cyberdb.co/best-cyber-security-twitter-profiles-follow-2018/ , accessed 08.07.2021
^d bridewellconsulting.com/100-security-experts-follow-twitter/ , accessed 08.07.2021
^e darkreading.com/vulnerabilities-threats/10-cybersecurity-twitter-profiles-to-watch/d/d-id/1325031/ , accessed 08.07.2021
^f sentinelone.com/blog/21-cybersecurity-twitter-accounts-you-should-follow/ , accessed 08.07.2021

Appendix B Codebook

In Table 5 the codebook [24] for the annotation of tweets is presented, which is applied to the coding of the dataset (see Sect. 3.1). Table 5 gives an overview of the codes' definitions.

Table 5. Codebook for tweet relevance classification.

Code	Definition	Example
Relevant (2)	Information on existence, properties, assessment, real-world application or warning of (1) vulnerabilities in software, (2) vulnerabilities in hardware, (3) malware, or (4) attack vectors, that are (a) currently in use, (b) may be (ab-used) or (c) in theory	“Zeppelin, a new #ransomware variant of Vega family, is targeting #technology and health companies across Europe, the US and Canada.” ^a , “Frankfurt City IT Network Taken Offline to Stop #Emotet #Botnet Infection” ^b , “Citrix Vulnerability Puts 80K Companies at Risk” ^c
Irrelevant (1)	None of the above	

^a Twitter (twitter.com/unix_root/status/1204813126371295238)

^b Twitter (twitter.com/neirajones/status/120881702295068672)

^c Twitter (twitter.com/InfosecurityMag/status/1209175732695523330)

Appendix C Classifier Comparison

Figure 3 depicts the results of active classifier comparison. Experiment details are discussed in Sect. 3.2.

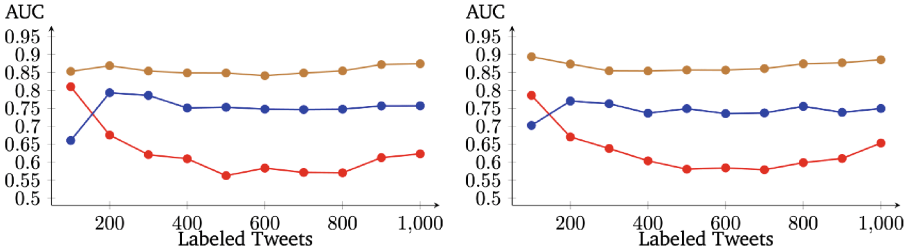


Fig. 3. Performance comparison of Naive Bayes (red), kNN with $k = 50$ (blue) and Random Forest (brown) classifier with uncertainty sampling based on their respective model on dataset $S1$ (left) and $S2$ (right). Average over 5 executions using Cross-Validation. (Color figure online)

Appendix D Alert Generation by Similarity Threshold

Table 6 depicts how recall and alert generation is impacted by the similarity threshold of the greedy clustering (see Sect. 3.3).

Table 6. Performance measures of greedy clustering-based generated alerts for different similarity thresholds and for alert count thresholds 3 and 5 for the datasets $S1$ and $S2$, respectively.

Alert count thresh.	3 ($S1$)				5 ($S2$)		
	0.25	0.3	0.4	0.5	0.2	0.25	0.3
Precision	81.54%	96.08%	90.63%	94.11%	75%	95.24%	86.67%
Recall	100%	96.23%	60.41%	30.18%	100%	95.24%	61.9%
F_1	89.83%	96.15%	72.5%	45.7%	86%	95.24%	72.22%

References

1. Reuter, C., Kaufhold, M.A.: Fifteen years of social media in emergencies: a retrospective review and future directions for crisis informatics. *J. Contingencies Crisis Manage.* **26**(1), 41–57 (2018)
2. Husák, M., Jirsík, T., Yang, S.J.: SoK: contemporary issues and challenges to enable cyber situational awareness for network security. In: *Proceedings of the 15th International Conference on Availability, Reliability and Security. ARES 2020.* Association for Computing Machinery, New York, NY, USA (2020)

3. Yang, W., Lam, K.Y.: Automated cyber threat intelligence reports classification for early warning of cyber attacks in next generation SOC. In: International Conference on Information and Communication Systems (ICICS), pp. 145–164 (2020)
4. Mittal, S., Das, P.K., Mulwad, V., Joshi, A., Finin, T.: CyberTwitter: using Twitter to generate alerts for cybersecurity threats and vulnerabilities. In: 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 860–867. IEEE (2016)
5. Behzadan, V., Aguirre, C., Bose, A., Hsu, W.: Corpus and deep learning classifier for collection of cyber threat indicators in Twitter stream. In: 2018 IEEE International Conference on Big Data (Big Data), pp. 5002–5007. IEEE (2018)
6. Tundis, A., Ruppert, S., Mühlhäuser, M.: On the automated assessment of open-source cyber threat intelligence sources. In: Krzhizhanovskaya, V.V., et al. (eds.) ICCS 2020. LNCS, vol. 12138, pp. 453–467. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-50417-5_34
7. Alves, F., Andongabo, A., Gashi, I., Ferreira, P.M., Bessani, A.: Follow the blue bird: a study on threat data published on Twitter. In: Chen, L., Li, N., Liang, K., Schneider, S. (eds.) ESORICS 2020. LNCS, vol. 12308, pp. 217–236. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58951-6_11
8. Koops, B.J., Hoepman, J.H., Leenes, R.: Open-source intelligence and privacy by design. *Comput. Law Secur. Rev.* **29**(6), 676–688 (2013)
9. Sabottke, C., Suci, O., Dumitras, T.: Vulnerability disclosure in the age of social media: exploiting Twitter for predicting real-world exploits. In: 24th USENIX Security Symposium USENIX Security 15, pp. 1041–1056 (2015)
10. Atefeh, F., Khreich, W.: A survey of techniques for event detection in Twitter. *Comput. Intell.* **31**(1), 132–164 (2015)
11. Alves, F., Bettini, A., Ferreira, P.M., Bessani, A.: Processing tweets for cybersecurity threat awareness. arXiv preprint [arXiv:1904.02072](https://arxiv.org/abs/1904.02072) (2019)
12. Trabelsi, S., et al.: Mining social networks for software vulnerabilities monitoring. In: 2015 7th International Conference on New Technologies, Mobility and Security (NTMS), pp. 1–7. IEEE (2015)
13. Hasan, M., Orgun, M.A., Schwiter, R.: A survey on real-time event detection from the Twitter data stream. *J. Inf. Sci.* **44**(4), 443–463 (2018)
14. Kaufhold, M.A., Bayer, M., Reuter, C.: Rapid relevance classification of social media posts in disasters and emergencies: A system and evaluation featuring active, incremental and online learning. *Inf. Process. Manage.* **57**(1), 102132 (2020)
15. Habdank, M., Rodehutsors, N., Koch, R.: Relevancy assessment of tweets using supervised learning techniques: mining emergency related tweets for automated relevancy classification. In: 2017 4th International Conference on Information and Communication Technologies for Disaster Management (ICT-DM), pp. 1–8. IEEE (2017)
16. Settles, B.: Active learning literature survey. University of Wisconsin (2010)
17. Imran, M., Mitra, P., Srivastava, J.: Enabling rapid classification of social media communications during crises. *Int. J. Inf. Syst. Crisis Response Manage. (IJIS-CRAM)* **8**(3), 1–17 (2016)
18. Lewis, D.D., Catlett, J.: Heterogeneous uncertainty sampling for supervised learning. In: Machine Learning Proceedings 1994, pp. 148–156. Elsevier (1994)
19. Allan, J., Lavrenko, V., Jin, H.: First story detection in TDT is hard. In: Proceedings of the Ninth International Conference on Information and Knowledge Management, pp. 374–381 (2000)

20. Ritter, A., Wright, E., Casey, W., Mitchell, T.: Weakly supervised extraction of computer security events from Twitter. In: Proceedings of the 24th International Conference on World Wide Web, pp. 896–905 (2015)
21. Concone, F., De Paola, A., Re, G.L., Morana, M.: Twitter analysis for real-time malware discovery. In: 2017 AEIT International Annual Conference, pp. 1–6. IEEE (2017)
22. Dionisio, N., Alves, F., Ferreira, P.M., Bessani, A.: Cyberthreat detection from twitter using deep neural networks. In: 2019 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2019)
23. Bose, A., Behzadan, V., Aguirre, C., Hsu, W.H.: A novel approach for detection and ranking of trendy and emerging cyber threat events in Twitter streams. In: Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 871–878 (2019)
24. Mayring, P.: Qualitative content analysis. *Companion Qual. Res.* **1**(2004), 159–176 (2004)
25. Sapienza, A., Ernala, S.K., Bessi, A., Lerman, K., Ferrara, E.: Discover: mining online chatter for emerging cyber threats. In: Companion Proceedings of the The Web Conference 2018, pp. 983–990 (2018)
26. Le Sceller, Q., Karbab, E.B., Debbabi, M., Iqbal, F.: Sonar: automatic detection of cyber security events over the Twitter stream. In: Proceedings of the 12th International Conference on Availability, Reliability and Security (ARES), pp. 1–11 (2017)
27. Lee, K.C., Hsieh, C.H., Wei, L.J., Mao, C.H., Dai, J.H., Kuang, Y.T.: Sec-buzzer: cyber security emerging topic mining with open threat intelligence retrieval and timeline event annotation. *Soft. Comput.* **21**(11), 2883–2896 (2017)
28. Dionísio, N., Alves, F., Ferreira, P.M., Bessani, A.: Towards end-to-end cyberthreat detection from twitter using multi-task learning. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2020)
29. Fang, Y., Gao, J., Liu, Z., Huang, C.: Detecting cyber threat event from twitter using IDCNN and BiLSTM. *Appl. Sci.* **10**(17), 5922 (2020)
30. Ji, T., Zhang, X., Self, N., Fu, K., Lu, C.T., Ramakrishnan, N.: Feature driven learning framework for cybersecurity event detection. In: Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, pp. 196–203 (2019)
31. Khandpur, R.P., Ji, T., Jan, S., Wang, G., Lu, C.T., Ramakrishnan, N.: Crowdsourcing cybersecurity: Cyber attack detection using social media. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 1049–1057 (2017)
32. Mittal, S., Joshi, A., Finin, T.: Cyber-all-intel: an AI for security related threat intelligence. arXiv preprint [arXiv:1905.02895](https://arxiv.org/abs/1905.02895) (2019)
33. Simran, K., Balakrishna, P., Vinayakumar, R., Soman, K.P.: Deep learning approach for enhanced cyber threat indicators in Twitter stream. In: Thampi, S.M., Martinez Perez, G., Ko, R., Rawat, D.B. (eds.) SSCC 2019. CCIS, vol. 1208, pp. 135–145. Springer, Singapore (2020). https://doi.org/10.1007/978-981-15-4825-3_11
34. Bernard, J., Zeppelzauer, M., Lehmann, M., Müller, M., Sedlmair, M.: Towards user-centered active learning algorithms. In: Computer Graphics Forum, vol. 37, pp. 121–132. Wiley Online Library (2018)
35. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint [arXiv:1301.3781](https://arxiv.org/abs/1301.3781) (2013)