Information Systems 95 (2021) 101586

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/is

Processing tweets for cybersecurity threat awareness

Fernando Alves *, Aurélien Bettini, Pedro M. Ferreira, Alysson Bessani

LASIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal

ARTICLE INFO

Article history:
Received 20 April 2020
Received in revised form 24 June 2020
Accepted 27 June 2020
Available online 4 July 2020
Recommended by Dennis Shasha

Keywords:
Threat intelligence
Threat discovery
OSINT
Twitter
Machine learning
Stream clustering

ABSTRACT

Receiving timely and relevant security information is crucial for maintaining a high-security level on an IT infrastructure. This information can be extracted from Open Source Intelligence published daily by users, security organisations, and researchers. In particular, Twitter has become an information hub for obtaining cutting-edge information about many subjects, including cybersecurity. This work proposes SYNAPSE, a Twitter-based streaming threat monitor that generates a continuously updated summary of the threat landscape related to a monitored infrastructure. SYNAPSE is designed to accurately select any kind of cybersecurity events and summarise them for the convenience of security analysts. Its tweet-processing pipeline is composed of filtering, feature extraction, binary classification, an innovative clustering strategy, and generation of Indicators of Compromise (IoCs). A quantitative evaluation considering over 195.000 tweets from 80 accounts over more than 8 months, shows that our approach successfully finds the majority of security-related tweets concerning an example IT infrastructure (true positive rate above 90%), incorrectly selects a small number of tweets as relevant (false positive rate under 10%), and summarises the results in few IoCs per day. A qualitative evaluation of the IoCs generated by SYNAPSE demonstrates their relevance, and timeliness. Finally, we provide some highlights of a real-world integration of SYNAPSE with the Security Operation Center of a nation-wide electric utility.

1. Introduction

A security analyst must be aware of the latest developments regarding updates, patches, mitigation measures, vulnerabilities, attacks, and exploits to adequately protect an IT infrastructure. Security Operations Centers (SOC) improve their awareness through Security Information and Event Management (SIEM) software, thereby allowing the correlation of the latest cybersecurity developments with internal infrastructure events.

There are two primary ways of obtaining cybersecurity news. One is to purchase a curated feed from a specialised company such as SenseCy [1] or SurfWatch [2]. Another, is to collect Open Source Intelligence (OSINT) [3] available from various sources on the internet (*e.g.*, Threatpost [4]). Integrating OSINT in SOCs became a simple and attractive approach to complement cybersecurity events with the latest intelligence discovered by the practitioners. In particular, Twitter earned the spotlight as an information source due to its natural aggregation capabilities [5] and attractive short messages. The research community has taken a general interest in exploiting Twitter's capabilities and use them in many different contexts, such as detecting emerging topics [6], detecting earthquakes [7], recommending new contents to follow [8,9], or predicting stock changes [10].

Previous research shows that Twitter is a relevant source of cybersecurity intelligence [11], in addition to being timely [12–15]. In fact, the most important cybersecurity news feeds are present in Twitter (*e.g.*, NVD, ExploitDB, CVE, Security Focus), making it a hub for all these sources. Although short, tweets provide enough elements to categorise their content, as well as links for more detailed material.

There are two requirements for efficient OSINT (including Twitter) usage [16,17]: adequate data selection, and post-processing in the form of data aggregation and deduplication. There are numerous threat intelligence tools (*e.g.*, SpiderFoot [18], IntelMQ [19]) that can collect cybersecurity-related OSINT (including tweets). However, these focus on collecting data from a wide variety of sources or on capabilities such as ordering. They use simple keyword-based filters to narrow the big volume of collected information, and do not employ sophisticated methodologies to address the aforementioned requirements.

In the literature, one can find many different proposals for selecting cybersecurity-related tweets [15,20–32]. However, to the best of our knowledge, (with the exceptions of Le et al. [32] and Dionísio et al.'s [24] approaches, which share similarities with our data collection methodology) all approaches are either: dependent on a secondary cybsersecurity data source

* Corresponding author at: LASIGE, Faculdade de Ciências, Universidade de Lisboa, Portugal.
E-mail addresses: fbalves@fc.ul.pt (F. Alves), pmferreira@fc.ul.pt (P.M. Ferreira), anbessani@fc.ul.pt (A. Bessani).

https://doi.org/10.1016/j.is.2020.101586
0306-4379/

to validate the tweets [23,27,29,33]; or restricted to collecting tweets discussing a fixed number of cybersecurity topics [20–22,25,26,28,30,31] or to specific events such as exploits [15] or possible zero day vulnerabilities [23]. Both approaches are not practical for a real use case scenario, as any sort of restrictions will lose timeliness [12–14] or important information. Moreover, the research community largely overlooked the post-processing of the collected cybersecurity data, which is simply not discussed in the majority of papers. Organising tweets by taxonomies or topics [15,20–23,25,26,28,30,31] is too coarse grained to be used in practice (more details in Section 2). Finally, as an extension to the previous challenge, a system considering a production environment has to process a *stream* of tweets, something overlooked in the literature.

To address these gaps, this paper proposes SYNAPSE, a Twitter-based streaming threat monitor that generates a continuously updated summary of the threat landscape concerning a monitored infrastructure. SYNAPSE's design addresses two main challenges [16]: cybersecurity-related content selection, and the aggregation of related tweets. Tweets are collected from carefully selected accounts who discuss all kinds of cybersecurity-related intelligence. Then, a supervised machine learning model selects those relevant for the protection of an IT infrastructure, avoiding the presentation of repeated information by employing a novel stream clustering method adapted to the context of cybersecurity. We collect tweets with no topic restrictions from a set of cybersecurity-related accounts. This way, the collected data is more likely related to any kinds of cybersecurity subjects, including for example patches—a type of security data overlooked in the literature. To comply with the challenge of post-processing a tweet *stream*, our *k*-means application strategy aims at organising the tweets such that each cluster discusses the same subject, *i.e.*, the same news-item regarding the same product. Through this method, security analysts can observe only one element from each cluster. Finally, to enable SYNAPSE's integration with SIEMs (*e.g.*, IBM QRadar [34]) and threat intelligence/sharing tools (*e.g.*, MISP [35]), SYNAPSE creates IoCs from the obtained clusters by featuring some specific keywords from the tweets.

We experimentally validated that SYNAPSE is capable of selecting cybersecurity-related tweets. A quantitative evaluation considering over 195.000 tweets from 80 accounts over more than 8 months, shows that SYNAPSE finds the majority of security-related tweets concerning an example IT infrastructure (true positive rate above 90%), and incorrectly selects a small number of tweets as relevant (false positive rate under 10%). We also perform a quantitative evaluation of the clustering strategy and show that it accurately aggregates tweets by specific issues. When compared to a naive text-filtering approach (as employed by most threat intelligence systems used in practice), it decreases the number of tweets presented by approximately 80%, with the number of summarised IoCs being only 21% of the tweets classified as relevant. This volume of data can either be inspected manually or processed by a SIEM as OSINT-generated events. Further, a qualitative analysis of the largest 65 clusters generated by SYNAPSE revealed two paramount findings. Firstly, 43% of the IoCs describe high-impact security alerts, and for half of these, the tweet publication preceded the vulnerability publication on the National Vulnerability Database (NVD) by eight days (on average). Secondly, 70% of the analysed clusters provided serviceable intelligence, including exploits whose vulnerabilities were not matched to NVD entries.

Finally, SYNAPSE was integrated with the Security Operation Center of a nation-wide electric utility. Together with SOC operators, we were able to design solutions that integrate tweet-based IoCs in the SOC's daily operation. The resulting rules enriched internal events with external data, and increased the SOC awareness to critical cybersecurity events. We provide access to a anonymised version of a SYNAPSE dashboard (otherwise equal to the one provided to the SOC), to show its data selection and aggregation qualities [36].

In summary, our contributions are:

1. An end-to-end streaming threat monitor architecture for collecting, classifying, and clustering tweets related to a specified infrastructure (Section 3);
2. A novel application strategy and adaptation of well-known clustering techniques to the context of cybersecurity threat awareness (Section 4);
3. A detailed system evaluation using three real-world datasets and a qualitative analysis of the security alerts generated thereof (Section 6);
4. Methods for generating MISP-compatible IoCs from tweets that enable the integration of SYNAPSE into SOC operation (Sections 7 and 3.6);
5. Highlights of the integration and real-world validation in a SOC of the techniques proposed (Section 7.4).

2. Related work

In the following, we briefly review the previous work related to SYNAPSE: processing tweets for cybersecurity, threat intelligence tools, and stream clustering algorithms.

2.1. Twitter for cybersecurity

Several previous works aim to find cybersecurity OSINT about a given IT infrastructure. Okutan et al. [30] integrate tweets with posts from the GDELT news service and Hackmageddon to detect new threats related to one of three topics: Defacement, Denial of Service, and Malicious Email/URL. Khandpur et al. [31] use semantic trees to validate if tweets mention one of three topics: distributed denial of service attacks, data breaches, and account hijacking. Liu et al. [37] use semantic trees to group cybersecurity tweets by their semantic. Sabottke et al. [15] show that information about exploits are published on Twitter two days before they are included in NVD (on average). These entities are used to detect complex events and categorise them into one of seven topics. Behzadan et al. [25] use two convolutional neural networks, one to assess the relevance of a tweet for cybersecurity, the other to assign it to one of seven topics. Ji et al. [38] use a multitask neural network where each task is classifying if tweets are related to seven topics. Bose et al. [39] extract keyword-based and social-based features to cluster them and detect trending events concerning 11 different topics. Niakanlahiji et al. [40] use regex expressions to extract IoCs from tweets and form discussion threads by chaining tweet replies. Mittal et al. [20] use a knowledge base created from security concepts to evaluate if a tweet is relevant for cybersecurity. Le Sceller et al. [21] designed a framework that collects tweets on a keyword basis and is capable of extending the keyword set automatically, focused on six topics. Ritter et al. [22] search Twitter for occurrences of three specific topics: DoS attacks, data breaches, and account hijacking. Yagcioglu et al. [26] fuse various machine learning techniques to select tweets relevant for cybersecurity by following a cybersecurity taxonomy. Sapienza et al. [27] validate tweets mentioning new threats using darkweb sources. The authors published an extension of this work that also uses technical blogs to increase the quality of the approach [33]. Lee et al. [29] complement Twitter with blogs to form a timeline of cybersecurity events. Le et al. [32] use CVE descriptions (therefore, only positive samples) to train a classifier and infer if a tweet is relevant for cybersecurity. Trabelsi et al. [23] cluster tweets by subject. Threats not referred by NVD are considered novel and handled like zero-day vulnerabilities.

Table 1
Summary of the related work and comparison of the techniques used.

Authors	Number of topics	No external validation	Aggregation capabilities	Stream processing
Okutan et al. [30]	3	✗	✗	✗
Khandpur et al. [31]	3	✓	✗	✗
Liu et al. [37]	5	✓	✓	✓
Sabottke et al. [15]	7	✗	✗	✗
Behzadan et al. [25]	7	✓	✗	✗
Ji et al. [38]	7	✓	✗	✗
Bose et al. [39]	11	✓	✓	✗
Niakanlahiji et al. [40]	13	✓	✗	✓
Mittal et al. [20]	Taxonomy	✗	✗	✗
Le Sceller et al. [21]	Taxonomy	✗	✓	✗
Ritter et al. [22]	Taxonomy	✓	✗	✗
Yagcioglu et al. [26]	Taxonomy	✓	✗	✗
Sapienza et al. [27]	Unrestricted	✗	✗	✗
Sapienza et al. [33]	Unrestricted	✗	✗	✗
Lee et al. [29]	Unrestricted	✗	✗	✗
Le et al. [32]	Unrestricted	✓	✗	✗
Trabelsi et al. [23]	Unrestricted	✗	✓	✗
Dionísio et al. [24]	Unrestricted	✓	✗	✗
SYNAPSE (This paper)	Unrestricted	✓	✓	✓

Dionísio et al. [24] used deep learning techniques to detect and extract security-related information from tweets.

Table 1 presents a summary of these works according to four features: topic or taxonomy-restricted data collection, tweet validation through external knowledge, data aggregation, and stream processing capabilities. With the exceptions of Le et al. [32] and Dionísio et al.'s [24] approaches – which share similarities with our data collection methodology – all approaches are either: dependent on a secondary cybsersecurity data source to validate the tweets [23,27,29,33]; or restricted to collecting tweets discussing a fixed number of cybersecurity topics [20–22,25,26,28,30,31] or to specific events such as exploits [15] or possible zero day vulnerabilities [23]. Both approaches are not practical for a real use case scenario. The first solution is likely to lose the Twitter timeliness advantage [12–14]. The latter is too restrictive, as companies cannot afford to be protected only against some attacks. Furthermore, by setting a number of topics to focus on, one of two things will happen when a tweet discussing an unpredicted topic appears: either the tweet is discarded because it does not fit the predefined model (and possibly important information is lost), or it is placed in a topic it does not belong to. In both cases the user is likely to lose confidence that the system is performing correctly.

As can be seen in Table 1, the research community largely overlooked the post-processing of the collected cybersecurity data, which is simply not discussed in the majority of papers. Among the previously mentioned works, only five summarise the collected tweets. Le Sceller et al. [21] use Local Sensitive Hashing [41] to group tweets by similarity to calculate a relevance metric. Clusters with less than 10 elements are disregarded by the system. We believe this methodology is inappropriate for cybersecurity as many relevant but non-critical security issues will be discarded by the system. Trabelsi et al. [23] use *k*-means to group the tweets by threat. However, the authors do not expose the methodology used to find the critical parameter *k*, nor show any validation of the clustering methodology. Bose et al. [39] use the DBSCAN density-based clustering algorithm [42] to aggregate tweets by events. However, their results show high heterogeneity of threat types by event, meaning that, in practice, a SOC operator would have to manually inspect all elements of each event. Liu et al. [37] employ semantic trees to group tweets according to various formats, such as continuous occurrences or connecting seemingly disparate events. The authors set the number of clusters to form by observing cybercrime statistics. The presented methodology is clearly useful for detecting advanced persistent threats, but the authors do not make it clear how in can be useful for the daily operations of security operation centres. Furthermore, none of these works show a proper evaluation of the methodologies.

We may consider that the proposals that collect tweets based on taxonomies or topics [15,20–23,25,26,28,30,31] could present the collected data organised according to those topics. However, this organisation is too coarse grained to be used in practice. The following two tweets provide an example:

- "*#0daytoday #Joomla Easy Youtube Gallery 1.0.2 SQL Injection Vulnerability [webapps #exploits #Vulnerability #0…*";
- "*#0daytoday #Wordpress Ocim MP3 Plugin SQL Injection Vulnerability [webapps #exploits #Vulnerability #0day #Ex…*".

While both discuss the same *topic* (SQL injection), they clearly discuss different issues. To be useful, the data needs to be clearly organised so that each tweet set discusses *exactly the same subject*, and to the best of our knowledge, no proposal in the literature addresses this requirement.

Finally, as an extension to the previous challenge, a system considering a production environment has to process a *stream* of tweets. This means that batch aggregation techniques (of which the standard is clustering [43]) are not adequate as these are not designed to describe data evolution over time. Thus, a suitable stream clustering algorithm becomes necessary [44]. However, existing algorithms (*e.g.*, [45–47]) have two shortcomings for our context: they require *a priori* definition of the target number of clusters, and they discard outliers. When processing a cybersecurity news feed, the number of active threats under discussion is unknown in advance, and outliers cannot be discarded as they are likely to represent new threats. As can be seen in Table 1, SYNAPSE (described in this paper) is the only solution covering all techniques, and the only one considering stream processing.

2.2. Threat intelligence tools

Research-oriented work focus on gathering OSINT and transforming it into machine-readable IoCs for feeding Intrusion Detection Systems (IDS), anti-viruses, or other tools. Mathews et al. [48] employ traditional (*e.g.*, logs) and non-traditional (*e.g.*, forums, blog posts) data sources to create an ontology that infers the legitimacy of traffic flows, feeding an IDS with the results. Liao et al. [49] developed a framework for extracting IoCs from

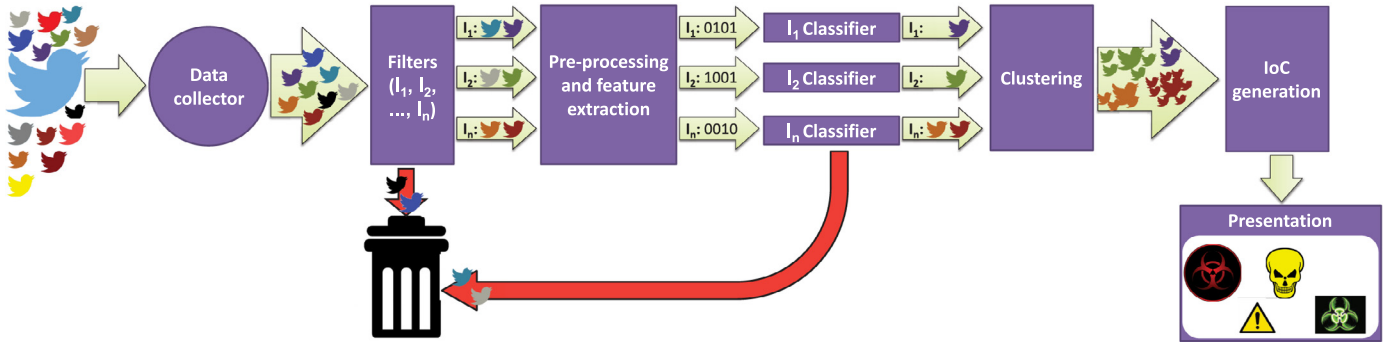

Fig. 1. SYNAPSE's architecture. Collected tweets pass through various stages and those classified as relevant are aggregated, transformed in IoCs, and delivered to analysts.

technical literature. In a different work, Zhu et al. [50] present a system that processes the scientific literature studying Android malware and extracts features describing the attacks to create a malware detector. The objective of these works is to extract machine-readable information from OSINT, which is different from our goal.

Besides the research-oriented efforts to include OSINT in protection systems, off-the-shelf tools are able to collect and deliver OSINT-based threat intelligence. SpiderFoot [18] is an OSINT automation tool that uses multiple sources (*e.g.*, Bitcoin addresses, Twitter) for three main purposes: target reconnaissance, assess an organisation's exposure on the Internet, and OSINT collection for security purposes. IntelMQ [19] is an open-source system for collecting and processing security-related OSINT feeds designed for organising data coming from various sources. It employs an ontology for data harmonisation and converts all events into a uniform `json` format. MISP [35] is a threat intelligence platform designed for sharing and correlating IoCs. It receives many types of threat inputs and exports its data into other MISP instances or threat intelligence tools. Generally speaking, these tools do not employ any advanced processing capability for filtering and matching threats, resorting only to keyword-based string comparisons.

2.3. Stream clustering

With the few exceptions discussed below, most stream clustering algorithms require the target number of clusters (k) to be defined as a parameter and discard elements that do not fit the clusters (outliers) [44]. Feng et al. [51] cluster only the tweets' hashtags, using text similarity to adapt the number of clusters to the collected data. However, this algorithm would potentially miss important information in the security field, as the clustering would not consider the full tweet text, only hashtags. Saki et al. [52] use a density-based clustering approach, therefore avoiding the definition of k. However, their technique discards outliers, which could lead to missing important emerging threats. Shou et al. [53] approach allows the value of k to vary up to an upper limit, but its outlier detection mechanism discards topics that do not gain traction, ignoring possibly important threats that remain unknown for long periods of time.

3. SYNAPSE pipeline

Fig. 1 presents SYNAPSE's architecture and data processing stages – tweet gathering, filtering, feature extraction, classification, clustering, and IoC generation – described next.

3.1. Data collection

The data collector module requires a set of accounts, from which it will collect every posted tweet using Twitter's stream API—an approach already found in the literature [29,32,33]. These accounts can be from security analysts and organisations, vendors, hackers, researchers, among others. They are chosen considering the likelihood of users tweeting about the security of elements belonging to the monitored IT infrastructure. Since usually security analysts already follow OSINT sources and Twitter accounts, it is just a matter of providing these sources to SYNAPSE.

Simply collecting tweets by keywords is a method likely to retrieve large amounts of irrelevant information. For instance, tweets with the word "windows" include all Windows-related topics (the OS) and all tweets referring glass windows. By collecting tweets only from selected security-related accounts, a more substantial fraction of tweets is related to cybersecurity.

3.2. Filtering

Despite the account-based collection approach, most likely the collected data will include tweets unrelated to the infrastructure under the analyst's care. These have to be dropped by a filter. The filtering approach assumes that a tweet referring a threat to a particular IT infrastructure asset has to mention that asset. Therefore, a second input is required: a set of keywords describing the assets of the monitored IT infrastructure. Only tweets that include at least one of the keywords will pass the filter. Keywords further restrict the scope of the security events, hence decreasing the number of irrelevant tweets beyond the filter.

To maximise the effectiveness of SYNAPSE, the keywords defining the monitored assets must be as complete and specific as possible. For example, if the analyst is in charge of securing a Linux cluster running virtual machines to serve a web service with a database, the keyword set could be {`linux`, `ssh`, `virtualbox`, `vbox`, `mysql`, `apache`, `php`}.

3.3. Pre-processing and feature extraction

Pre-processing normalises the tweet representation. First, all characters are converted to lower case, and stopwords and hyperlinks are removed—the latter are shortened URLs that provide little information. Numbers, dots, and hyphens are replaced by their textual representation (*e.g.*, "2" to "two"), as these are relevant to distinguish software versions (*e.g.*, Mozilla Firefox 4.5.1-2). Finally, all non `[a-z]` characters are removed. For instance, after pre-processing, the tweet "*#Oracle #Linux 6/7: Unbreakable Enterprise kernel (ELSA-2016-3573)* https://t.co/vLTel8NodG" becomes "*oracle Linux six seven unbreakable enterprise kernel elsa hyphen*

two thousand and sixteen hyphen three thousand five hundred and seventy three". The original tweets are stored for presentation.

The tweets must be converted to a numerical format to become suitable for supervised learning classification techniques. This work uses the well-known Term Frequency–Inverse Document Frequency (TF–IDF) method [54]. TF–IDF computes weights to words (features) based on their occurrence frequency in each document and on the group of documents considered. The weight of a word increases with its frequency of occurrence in a single document but is scaled down by the frequency of occurrence in all documents. By mapping each consecutive word token to a corresponding vector position, tweets are converted to a constant size, zero-padded, TF–IDF numeric vector. Finally, to limit the size of the vector we employ the hashing trick technique [55].

3.4. Classification

For the classification of tweets according to their security relevance, two classifiers have been explored: Support Vector Machines (SVM) [56] and Multi-Layer Perceptron (MLP) Neural Networks (NN) [57,58]. The SVM is a broadly-used classifier achieving good results across a multitude of application domains. We consider the SVM implementation available in the Apache Spark Machine Learning library (MLlib) [59], which employs a linear kernel, thereby assuming the input vectors are linearly separable.

Since MLlib does not provide a non-linear SVM kernel, MLlib's MLP NN implementation was considered to account for the assumption that input vectors may not be linearly separable. The MLP is a well-established and frequently used NN architecture that has a long track record of good and consistent results over a vast number of classification tasks.

3.5. Clustering

SYNAPSE uses clustering to aggregate similar tweets in the news feed stream. The Clustream algorithm [60] was chosen as the basis for this pipeline stage as its structure and characteristics were closest to our requirements. However, it required adaptation to SYNAPSE's context to achieve threat aggregation as described in the next section.

3.6. MISP compatible IoC generation

After the clustering phase, the clusters of tweets are transformed into the IoC format to allow their inclusion in SIEMs or threat intelligence platforms. There are several standards for sharing IoCs, such as STIX [61] or MISP [62]. The format must be extensible and adaptable as tweets are unstructured and contain unpredictable content. For these reasons, we selected both MISP and CEF [63] formats to generate IoCs.

We use a combination of MISP items to generate the IoC. One MISP *Event* is composed of two *Objects* containing security indicators called *Attributes*: one describing the content of the exemplar tweet (Section 4); the other representing the cluster of tweets. Events are classified using tags, added according to a set of threat categories related to ENISA and VERIS cyberthreat taxonomies [64]. The OSINT tag is added to emphasise the automatic creation based on tweets. The classification is achieved by using regular expressions to match taxonomy elements in the exemplar's message, generating one tag for each match. Further, SYNAPSE includes in its IoCs security bulletin IDs (*e.g.*, CVE-IDs, Ubuntu security notice IDs) in a special field to streamline the correlation of OSINT with other events.

Fig. 2 depicts the taxonomy employed to represent IoCs in MISP (top of the figure). The exemplar tweet is the core of the IoC, while its cluster is an extra element to increase informativeness. The bottom of the figure shows a MISP Event generated from a cluster and its exemplar (the example cluster shown in Table 2). The OSINT object contains extracted information from the exemplar such as the tweet's message, any links therein, and the *Cluster_Analysis* object contains the remainder cluster data. A simple classification was applied: the OSINT tag marks the event as created from tweets, and the "Denial of Service" tag (from VERIS) classifies the threat.

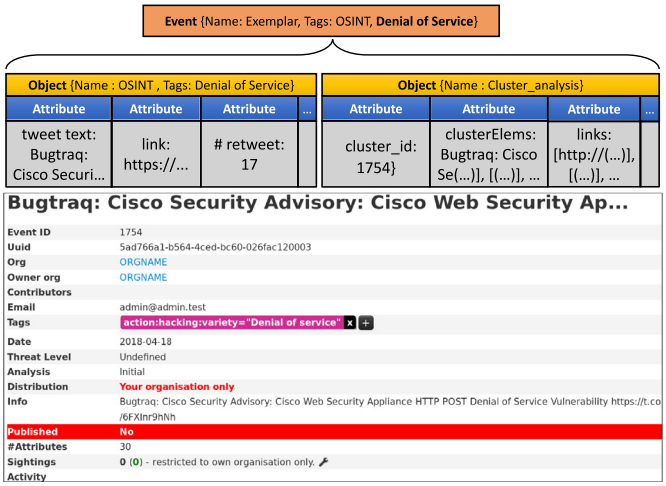

Fig. 2. Representation of a cluster into the MISP taxonomy [62] and an OSINT-generated event in MISP.

Table 2
An example of a cluster and its *exemplar* (in Bold).

Bugtraq: Cisco Security Advisory: Cisco Web Security Appliance HTTP POST Denial of Service Vulnerability https://t.co/6FXInr9hNh
Bugtraq: Cisco Security Advisory: Cisco Web Security Appliance HTTP POST Denial of Service Vulnerability https://t.co/6FXInr9hNh
Bugtraq: Cisco Security Advisory: Cisco Web Security Appliance HTTP Length Denial of Service Vulnerability https://t.co/TgU0T9vlZt #bugtraq
Bugtraq: Cisco Security Advisory: Cisco Web Security Appliance HTTP POST Denial of Service Vulnerability https://t.co/feZITxQKVC #bugtraq
#cybersecurity Bugtraq: Cisco Security Advisory: Cisco Web Security Appliance HTTP POST Denial of Service https://t.co/XUUctUnQ8F #infosec
#vulnerability #security : Bugtraq: Cisco Security Advisory: Cisco Web Security Appliance HTTP POST Denial of Serv https://t.co/9bW0ls00kx
#internet #security: Cisco Web Security Appliance HTTP POST Denial of Service Vulnerability https://t.co/cXQUTWUBbD

4. Tweet stream clustering

Since Twitter users can tweet or retweet about the same subject, SYNAPSE is expected to collect many similar tweets. Thus, to cover information about the IT infrastructure, the analyst would have to manually inspect a large amount of redundant data for each threat.

To alleviate this burden, clustering is used to group similar tweets classified as relevant for the protection of the IT infrastructure. Ideally, the information collected about a specific threat gets aggregated in one cluster, from which a single representative tweet – the *exemplar* – is presented to the analyst. By clustering the stream of relevant tweets, distinct active threats are summarised in a set of clusters and updated as more tweets are collected. It is through this mechanism that SYNAPSE can create an active threat monitor outlining the current threat landscape, *i.e.*, the current threats that potentially require more immediate attention from SOC analysts.

4.1. Data stream aggregation challenges

Clustering is commonly applied in batch, as an exploratory data technique where a static data set is clustered into k groups [43]. The number of clusters, k, is either defined *a priori* or estimated to satisfy performance metrics [43]. In a dynamic setting such as SYNAPSE's streaming context, defining k beforehand is not possible, as the number of threats being discussed at a given time is unknown. If at any moment SYNAPSE was processing t threats and clustering was set to find $k \neq t$ clusters, the result would contain clusters including unrelated threats, various clusters related to the same threat, or both cases. *Therefore, SYNAPSE requires a clustering algorithm able to adapt k over time.*

Furthermore, an essential feature of most stream clustering algorithms is the ability to detect and remove outliers that may disrupt the quality of the clustering. In the security context, performing outlier removal could prevent the discovery of emerging threats. Moreover, all tweets reaching SYNAPSE's clustering stage were classified as relevant, and should not be discarded. *Therefore, SYNAPSE requires a clustering algorithm capable of maintaining performance indicators (e.g., intra and inter-cluster cohesion) without removing outliers.*

4.2. DynamicClustream

The lack of solutions that fit the requirements of threat intelligence tools (see Section 2), motivated us to adapt the Clustream [60] algorithm for SYNAPSE, thus creating the DynamicClustream. The Clustream algorithm clusters a data stream in two phases. The online phase performs a simple and efficient clustering of the inbound stream by keeping only a summary of the data collected, thus abiding to the speed requirements of a data stream [43]. The offline phase is performed in background to provide a more complete analysis of the collected data through a more effective and computationally demanding clustering algorithm. Clustream includes an outlier detection mechanism that excludes data points unfit for any of the existing clusters by analysing the distance from that point to all clusters. A decision is only taken once it becomes clear if a data point is an element of a new trend or an isolated occurrence. The components that distinguish DynamicClustream from Clustream are detailed in the following.

4.3. High-level overview

Assume there is always a global cluster state S, defined as a set of sets, describing the clusters formed from a previously processed time-window of tweets. When a new tweet t is received, the online clustering component attempts to place t in one of S's clusters. If a direct placement is not possible, the offline clustering component is triggered to compute a new clean cluster state considering the tweets in the clusters of S plus t.

Once a new cluster state S is in place, a final step is taken to obtain each cluster's *exemplar* tweet, *i.e.*, the tweet representing the cluster, that will be shown to the analyst. The exemplar tweet is selected by choosing the tweet with the smallest Euclidean distance to the centroid of the cluster. An example of a generated cluster (and its exemplar) appears in Table 2. The online and offline components of DynamicClustream are presented in Algorithm 1, with locking details for ensuring atomic updates on S omitted for better readability.

Algorithm 1: DynamicClustream online/offline clustering.

```
1  S ← ∅                                                    // global cluster state
2  Function OnlineClustering(t):
3      i ← GetNumHits(S, t)
4      if i = 0 then
5          AddNewCluster(S, t)
6      else if i = 1 then
7          UpdateCluster(S, t)
8      else                                                 // needs offline clustering
9          PlaceInClosestCluster(S, t)
10         schedule OfflineClustering(S)
11 Function OfflineClustering(SavedState):
12     T ← Flatten(SavedState)
13     ε* ← +∞; k ← 2; Clusters ← ∅
14     S* ← ∅
15     while T ≠ ∅ do
16         do
17             Clusters, ε ← KMeansClustering(T, k)
18             if ε < ε* then
19                 ε* ← ε
20                 k ← k + 1
21         while ε = ε* and k < |T|
22         forall the C ∈ Clusters do
23             if WTS(C) ≥ τ then
24                 S* ← S* ∪ {C}
25                 T ← T \ C
26     S ← MergeClusterState(S*, Flatten(S) \ Flatten(S*))
```

4.4. Online clustering component

The online clustering component uses a lightweight approach to assign a new tweet t to the current clustering state S. To do so, the membership of t is tested in all clusters (line 3) by employing the WTS cohesion measure (introduced below). This is done by adding t to each cluster $C_i \in S$ and calculating the corresponding WTS value. t belongs to C_i when WTS is above a certain threshold τ. If t does not fit in one of the existing clusters, a new cluster solely containing t is created (lines 4–5). If t belongs to a single cluster, it is added to that cluster (lines 6–7). When t fits more than one cluster, it is added (temporarily) to the cluster with the highest membership rate, and the offline clustering is scheduled (lines 9–10).

In SYNAPSE's application scenario it makes no sense to remove outliers. Instead, when new tweets do not belong to S, we treat them as the onset of a threat by adding new clusters with a single element which in time may receive additional tweets. This outlier processing mechanism allows adapting the number of clusters, k, to the novelty in the dataflow. Furthermore, it is through the online component of DynamicClustream that the active threat monitor is implemented: the system categorises new tweets as *new threats* or as *updates* to known ones, thus maintaining an updated threat summary about an IT infrastructure.

4.5. Cohesion measure

Cluster cohesion and cluster separation are concepts used to assess the validity of a partition generated by a clustering algorithm [65], which in most cases have a purely geometric interpretation. In SYNAPSE, cohesion is based on the similarity of tweets within a cluster and not on a geometric measure such as the distance to the cluster centroid, thus defining a context-based cluster validation approach, argued to be more effective [66].

To reinforce the one-to-one relation between clusters and threats, the cohesion measure must detect clusters whose tweets refer to the same threat. Assuming that a threat is expressed by a minimum number of words appearing in all tweets, the proposed cohesion measure – named *Within-cluster Threat Similarity* (WTS)

– is defined as $\frac{\omega}{w_m}$, where ω is the number of words shared by all the cluster's tweets and w_m is the number of words of the smallest tweet in the cluster. WTS is 0 if no words are shared by the tweets of a cluster, and 1 when all tweets share the words of the smallest tweet in the cluster. It assumes that if all cluster tweets share a sufficiently large number of words, then they mention the same threat.

The degree of separation of two clusters C_i and C_j is measured by the Jaccard index [67]. It is determined as $J = \frac{|C_i \cap C_j|}{|C_i \cup C_j|}$, corresponding to the ratio between the number of common words to C_i and C_j and the number of unique words of C_i and C_j. The lower its value, the more separated the clusters are.

4.6. Offline clustering component

The offline component applies the *k*-means clustering algorithm [68] repeatedly to provide more robust clusters. *k*-means is a widely used algorithm that has provided good efficiency and empirical success over the last 50 years [69]. However, it is commonly employed for exploratory data analysis, not for automatic text summarisation.

The *k*-means algorithm requires the specification of the number of clusters, *k*, which is unknown in this case. At a given time we do not know how many potential threats to our infrastructure are being discussed. Therefore, we defined a novel strategy to find the so-called elbow point [70], *i.e.*, the point beyond which by increasing *k* there is no significant improvement in the clusters' Sum of Squared Errors (SSE). This procedure automatically determines *k*, thus avoiding the specification of a threshold to find the elbow point or the visual inspection of the within-class-variance versus *k* graph.

k-*means application strategy:* Starting at $k_1 = 2$, a *k*-means model is trained for each successive $k_i = i+1$ number of clusters, which produces a corresponding SSE, denoted by ε_i. As the initial cluster centres are randomly chosen, there is a given variance σ_i associated to ε_i. As we keep increasing k_i, we expect ε_i to decrease up to the point where the magnitudes of ε_i and σ_i become of the same order. At this point $\varepsilon_{i+1} - \varepsilon_i$ might become zero or even negative, indicating that there is no significant SSE improvement in increasing k_i. Therefore, the iteration is stopped when the error (ε) stops decreasing or (the limit case where) the number of clusters corresponds to the number of tweets to be clustered, and k_i is selected as the number of clusters (lines 16–21).

By testing this approach, we found that small clusters had only very similar tweets, but other large clusters contained unrelated tweets. The cause might be two-fold: (1) *k*-means assumes spherical clusters that it tends to produce equally sized, which might not be adequate; and (2) the strategy to find *k* is not guaranteed to find the *best k*. To overcome this limitation, we use the WTS cohesion measure to quantify how closely related the tweets in a cluster are, and implement a *re-clustering method* that splits these clusters into smaller ones with related tweets. If WTS $\geq \tau$ (a specified threshold), indicating high cohesion, it enables the validation of clusters as *final*.

Re-clustering method: All tweets of non-final clusters are gathered (line 22–25) and re-clustered (lines 16–21) using *k*-means to allow similar tweets to be grouped. Then, the new clusters generated are again tested using their WTS, and the process is repeated for the non-final clusters. Eventually, all clusters are considered final, ideally each related to a single threat, and S^* is merged with S (line 26), i.e., S^* is updated with the tweets received since the algorithm started by executing a procedure similar to lines 3–10.

Offline clustering scheduling: At any time, there may be only one instance of the offline component in execution. Since multiple tweets received in a short time interval may trigger offline clustering, we employ the schedule keyword (line 10) to avoid overlapping executions. The idea is that each call to **schedule** OfflineClustering() notifies the system that offline clustering is required after this point, and saves the current cluster state for its next execution. Once the algorithm is started again (using the latest saved state), it process all tweets pending in S (line 12).

4.7. Time-window model

To fully adapt Clustream to our context we also changed the clustering ageing model used to remove clusters. This model is necessary to complete the adaptation of the cluster state to the data stream flow.

Clustream's window model is global in the sense that all data points are aged and removed using the same rule. However, this methodology does not fit SYNAPSE application domain, as different cybersecurity topics have different lifetimes. For example, news about an update are expected to last a few days, while advances about an active threat may continue for a month or more. Thus, in the cybersecurity field it makes more sense to adopt a local window model, monitoring ageing *by cluster* (by threat). As a consequence, whole clusters rather than single points should be removed in forthcoming clustering states.

In DynamicClustream a cluster C_i is removed from the cluster state S if it has been stale for a period of time longer than θ, *i.e.*, if θ time passes without C_i receiving a new data point. In this way, topics that no longer receive traction are stowed away, *while active topics retain all their elements, regardless of the time passed, which may be crucial for understanding the evolution of a threat.*

5. Experimental setup

This section describes the experimental work carried out to validate SYNAPSE. All code is written in Scala and deployed on the Apache Spark Framework [59].[1] We chose Spark as its data-structures are scalable and designed for large datasets. Also, Spark includes a scalable machine learning library called MLlib, used to implement all ML algorithms employed in this paper.

5.1. Infrastructure definition

We used a hypothetical IT infrastructure to set SYNAPSE's filter during its experimental evaluation. This infrastructure (presented in Table 3) is composed of software elements typically found in the IT world, such as the most common browsers and operating systems.

5.2. Tweet collection and labelling

We collected three datasets during three periods of time. Table 4 presents their collection periods, the sets of accounts used, and the number of tweets. After being collected and filtered using the keywords in Table 3, each tweet was manually labelled as positive or negative, thus creating labelled datasets suitable for supervised learning. The tweets were labelled as positive when mentioning information relevant for cybersecurity (*e.g.*, updates, vulnerabilties, exploits) to a part of the IT infrastructure.

Two sets of accounts, S1 and S2, were used for tweet collection, as shown in the third row of Table 4. The accounts are listed in Table 5.

[1] The source code is available at https://github.com/fernandoblalves/ScalaTweets.

Table 3
The hypothetical infrastructure designed for tweet collection and filtering.

Oracle, Cisco, Internet Explorer, Google Chrome, Chrome, Firefox, Microsoft edge, edge, WordPress, Joomla, wp, Microsoft windows, MS, Linux, Operating System, operating systems

Table 4
Datasets collection and labelling details.

Dataset:	D1		D2		D3	
Time period (from/to)	01/11/2015 01/04/2016		01/04/2016 15/05/2016		15/05/2016 10/07/2016	
Account sets	S1		S1, S2			
Total tweets collected	71 024		57 579		66 608	
Class distribution	Pos. 1697	Neg. 2008	Pos. 536	Neg. 4292	Pos. 1680	Neg. 2153

5.3. Classifier configuration

Supervised machine learning techniques require design tailored to the problem at hand. For each classifier employed, their relevant parameters and design variables were varied, namely the step size and the regularisation parameter (C) for the SVM, and the number of layers and neurons per layer for the MLP. The size of the TF–IDF feature vector considered was also varied for both classifiers. Through a Pareto-optimal search, ideal configurations were found: the best SVM uses a step size and C of 0.05 and 5, respectively, and the best MLP had 5 layers with 10 neurons each. Both models use feature vectors with a size of 3000, revealing a clear advantage in using high-dimensional feature vectors. A complete description of the methodology employed for the classifier's design can be found in Appendix A.

5.4. Clustering

SYNAPSE uses the *k*-means algorithm in the offline clustering component, configured with fifty iterations, a minimum of two clusters, and the remaining parameters with their default values. Clustering was performed on the set of tweets classified as positive.

The WTS cluster cohesion measure was set to $\tau = \frac{2}{3}$. This value was selected after preliminary experiments, reflecting the rationale that two tweets can be in the same cluster if and only if they share at least two-thirds of their words.

We compare our data presentation strategy with the one employed by threat intelligence tools and SIEMs capable of collecting OSINT (*e.g.*, AlienVault OTX [71], Spiderfoot [18]). For that, we set up a Logstash [72] instance fed by the same dataset as SYNAPSE, which selected as relevant tweets mentioning at least one of our infrastructure assets and containing at least one security concept.

The security concept keywords were selected using the following methodology. First, a list of documents is obtained by selecting all tweets labelled as positive from all datasets. After that, we removed stopwords, applied the TF–IDF method, and selected the words with TF–IDF value lower than a threshold ρ. Finally, the list was manually filtered for security-irrelevant content (such as numbers). We considered ρ values of 0.1, 0.2, and 0.3. After inspecting the results, $\rho = 0.2$ was chosen due to the provision of the most substantial amount of generic words without showing words related to a specific context. The Logstash security concept keyword set corresponding to $\rho = 0.2$ appears in Table 6.

For the time-window model we applied a θ value of seven days, *i.e.*, a cluster without updates for seven days is removed from the online clustering state. The same θ value was applied to the Logstash approach but globally, *i.e.*, all relevant tweets were removed from the active threat pool after a week.

Table 5
Sets of accounts used to create the datasets.

S1 **Accounts:** inj3ct0r, TrustedSec, Anomali, briankrebs, Secunia, exploitdb, alienvault, slashdot, dstrom, Info_Sec_Buzz, vuln_lab, threatintel, dangoodin001, ivspiridonov, ThreatFeed, pikisec, SANSInstitute, johullrich, drericcole, F1r3h4nd, MaldicoreAlerts, USCERT_gov, gcluley, hal_pomeran, SecurityWeek, SecurityNewsbot, sans_isc, e_kaspersky
S2 **Accounts:** TenableSecurity, securitywatch, securityaffairs, zer0element, notsosecure, CyberExaminer, SCMagazine, DMBisson, lennyzeltser, IT_securitynews, teamcymru, WordPress, MicrosoftEdge, JoomlaTips, sjzaib, SecurityMagnate, Cisco, Dell, linuxtoday, securityninja, cyberopsy, OWASP_Java, _WPScan_, d_plusk, threatpost, Rootsector, Microsoft, linuxfoundation, ChidoDike, Sec_Cyber, ptracesecurity, msftsecurity, LinuxSec, hack3rsca, CiscoSecurity, NytroRST, joomla, Windows, crackerhacker00, fstenv, HPE_Security, googlechrome, wordpressdotcom, packet_storm, RokaSecurity, Oracle, Firefox, wpbeginner, YoKoAcc, SecurityCrap, jasonlam_sec, threatmeter

Table 6
The words used in the Logstash filter.

Access, acl, admin, advisory, allow, arbitrary, aslr, assurance, attack, auth, buffer, bug, bypass, certificate, code, command, corruption, csrf, cve, cyber, denial, deployment, dereference, disclosure, execute, exploit, hack, heap, identity, injection, interception, leak, overflow, privilege, remote, root, scripting, security, stack, threat, unauthenticated, vuln, xss

6. Results

The tweet processing pipeline components were evaluated using the selected models and datasets D2 and D3. These consider only tweets in the future of those in the training set (D1), and include information posted by an additional and substantially larger set of accounts (S2) not considered in the training stage. This methodology embodies the idea that in a real deployment, models will classify future tweets possibly from a different set of accounts.

Considering that 10-fold cross-validation was employed during the model selection phase, it should be noted that the selected model configurations were trained for the evaluation phase using the whole D1 dataset. The feature vectors of D2 and D3 tweets were generated using the TF–IDF model determined using dataset D1. This guarantees that TF-IDF weights attributed to words in D2 and D3 will be coherent with those used to train the classifiers.

6.1. Classification

Fig. 3 shows the True Positive Rate (TPR) and True Negative Rate (TNR) of the SVM and MLP classifiers described in Section 5, considering also the average result of the 10-fold cross-validation over D1.

Overall, the results are slightly worse for D2 and D3 when compared to D1 (as expected), since new data presents unmodelled patterns to the classifiers. Focusing on the results obtained for D2 and D3, in general, the classifiers maintain very high TPR and TNR, except for the MLP TPR. In both cases, the TNR is higher than the TPR. The imbalance between positively and negatively labelled data in the training data sets (more negative samples) can explain a higher TNR.

In summary, *the SVM approach achieved the best results, displaying true positive and true negative rates around 90% and showing a small degradation of results in D2 and D3.* For these reasons, the SVM model was employed in all further experiments. These results support the application of a supervised classifier to select tweets relevant for cybersecurity.

6.2. Clustering

The following experiments evaluate SYNAPSE's ability to aggregate the dataflow into meaningful clusters, where each cluster

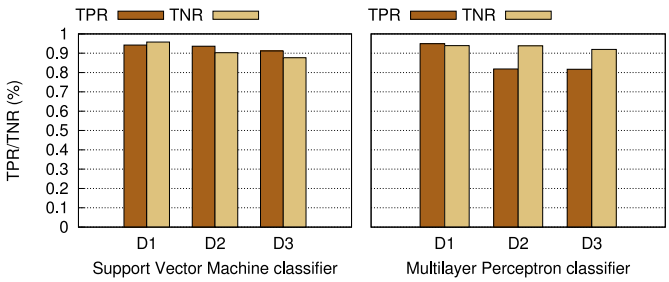

Fig. 3. SVM (left) and MLP (right) classifier results.

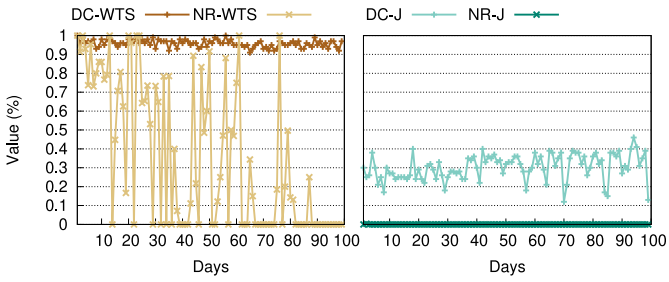

Fig. 4. Comparing WTS and Jaccard distance over time, for DynamicClustream with and without the re-clustering step.

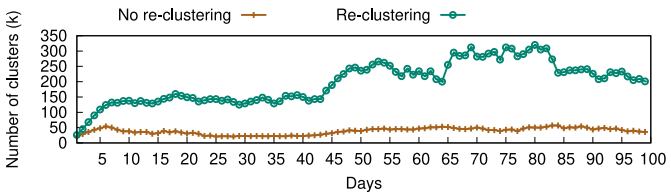

Fig. 5. Number of clusters obtained by the DynamicClustream algorithm with and without the re-clustering step.

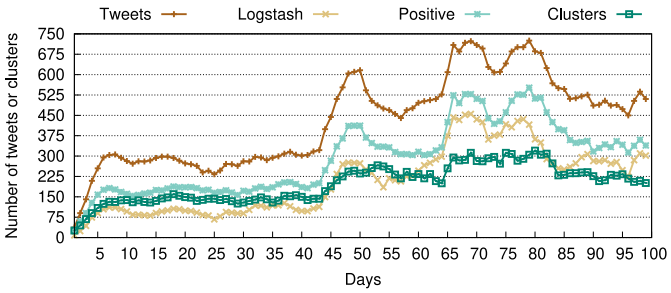

Fig. 6. The number of tweets collected and those filtered by Logstash, classification only, and classification and clustering.

is expected to describe a single threat. Further, the DynamicClustream's window model is evaluated to assess its capability to detect the continuous discussion of threats.

The initial clustering evaluation focuses on the basic algorithm's capability of properly aggregate tweets, *i.e.*, producing clusters with high internal cohesion and low inter-cluster similarity. Then we analyse the end-to-end benefit of SYNAPSE and discuss the effectiveness of the proposed outlier detection mechanism and time-window model which convey the active threat monitor functionality to SYNAPSE.

Datasets D2 and D3 were merged and fed to SYNAPSE. At the end of each day, for all clusters in the current cluster state, we calculated the average WTS and the Jaccard distance between all pairs of clusters. For the latter, we saved the largest value, which corresponds to the most similar cluster pair. Since SYNAPSE's objective is to obtain distinct clusters, each devoted to a single threat, the WTS should always be high (*i.e.*, the elements in each cluster are very similar), and the maximum Jaccard distance should be low (*i.e.*, there are no clusters that should be merged).

Fig. 4 shows the WTS and maximum Jaccard distance obtained, comparing the proposed DynamicClustream clustering algorithm (DC-WTS and DC-J) to its execution in clustering only mode, without considering re-clustering (NR-WTS and NR-J). The importance of including the re-clustering step (lines 22–25 of Algorithm 1) is clear since it raises the WTS to above 90% independently of the number of clusters and tweets present in the cluster state. The Jaccard distance, although with small values, is higher when using the re-clustering algorithm. Yet, this is an expected result. First, re-clustering produces significantly more clusters, therefore naturally decreasing their degree of separation. Second, since tweets in clusters mentioning different threats are likely to share commonly used security concept words and sentence structure, their similarity is increased.

Regarding the number of clusters obtained using either approach, the re-clustering algorithm naturally increases the number of clusters, as shown in Fig. 5. Nevertheless, we argue that in practice, the DynamicClustream algorithm improves the balance between maximising the relevance of the information presented and minimising the time required for its analysis. The WTS results provide guarantees that each cluster has similar tweets, likely about a single threat. Therefore, we can be confident that the set of cluster exemplar tweets provides a complete and accurate summary of the current threat landscape, thus not requiring additional time to analyse more tweets. Without the WTS cohesion validation, each cluster may discuss various threats – a highly plausible assumption based on the very low WTS values in Fig. 4 for the NR-WTS case – meaning that all tweets of each cluster would have to be analysed.

6.3. End-to-end benefit

The results presented in Fig. 6 highlight the end-to-end benefit of using SYNAPSE, and reinforces the importance of its clustering stage. The figure shows the reduction in the number of tweets that have to be analysed, when compared to the tweet stream, to the classifier output and to the naive Logstash filter described in Section 5.

The results show the need for efficient OSINT retrieval tools. Even with the naive keyword-based approach provided by the Logstash filter, the number of tweets marked as relevant would be extremely high, rendering the approach useless to SOC analysts. The introduction of a trained classifier decreases the amount of information by 65%. By attaching a clustering stage, we further reduce the information to be shown by almost 80%, which is a significant improvement.

6.4. Active threat monitor

To demonstrate the necessity of the active threat monitor implemented by the proposed stream clustering algorithm, we measured the active time for each of the 820 clusters formed during SYNAPSE's operation on the union of datasets D2 and D3. We define the duration of a cluster as the difference in days between the date of its creation and the date of the last added tweet. Fig. 7 depicts the distribution of the number of clusters over the cluster duration in days. The results clearly show that a global time-window model enforcing a fixed duration for each tweet would fail to detect active topics through time, since the threat discussion duration varies greatly (between 1 and 57 days), even in a dataset that covers only 100 days.

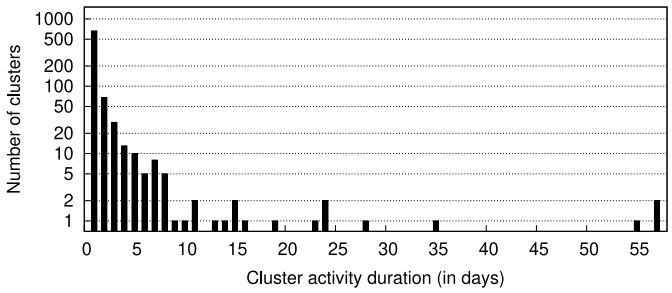

Fig. 7. The distribution of the number of clusters over the cluster duration in days.

6.5. Analysis of generated IoCs

Besides the ability to accurately select and aggregate tweets relevant to the security of an IT infrastructure, SYNAPSE provides useful threat intelligence for SOC analysts. To demonstrate this, we present some information about the timeliness, actionability, and relevance of the IoCs generated from the dataset used in previous experiments.

From the data collected over 3 months, SYNAPSE generated 820 clusters (IoCs) containing 1754 tweets. From these, we selected those with 5 or more tweets for analysis, obtaining 65 clusters comprising 466 tweets. These clusters are listed in Appendix B. The remaining 755 clusters have 1 (577 clusters), 2 (101), 3 (55), and 4 tweets (22). Our focus on larger clusters was motivated by the expectation that relevant threats are probably those that attract more attention and, ultimately, are mentioned in more tweets.

All tweets within each cluster were manually analysed. From these, as well as from any hyperlink therein, we extracted all CVEs mentioned (if any) and their Common Vulnerability Scoring System v3.0 (CVSS) [73] impact score, the types of actions that can be performed to respond to the alarm, and a comparison between the date of the earliest tweet in the cluster and the CVE's publication date on NVD.

The actionability information was divided into three categories: a patch is available (45 occurrences); a configuration to avoid the vulnerability exploitation is suggested (2 occurrences); and no directly actionable information is provided (14 occurrences). The latter is mostly associated with clusters mentioning exploits to vulnerabilities, with the tweet hyperlinks leading to proofs-of-concept. However, an expert might still make use of this information to prevent exploitation, as discussed in previous work [15]. Patches are mostly announced together with their associated vulnerabilities, regardless of indexing on NVD. In the end, 71% (46) of the clusters provided directly useable intelligence, including exploits whose vulnerabilities were not matched to NVD entries.

Among the 65 clusters, 36 mentioned a total of 122 different CVEs (15 clusters mentioning more than one CVE). Of these, only two have low impact score, about a quarter have medium impact (33), more than half are categorised with high impact (68), and more than a tenth have critical impact (14).

Considering their relevance, 43% (28) of the IoCs were related to CVSS scores above or equal to 7 (high severity) and 12% (8) to scores above or equal to 9 (critical severity). Regarding timeliness, 20% of the alerts were raised 8 days (on average) before their corresponding vulnerabilities were published on NVD.

As an illustration of the richness of the obtained data, Table 7 shows 10 representative IoCs selected from those analysed. In the table, the date column shows the date of the earliest tweet in the cluster and, when a number is shown within parenthesis, it denotes the number of days before publication on NVD. Two additional columns provide information about the threat type (as automatically classified by SYNAPSE) and relevant notes about the cluster content.

From the 10 clusters presented, 6 announce vulnerabilities before publication on NVD, all of them with patches available. Further, 7 are classified with a *high* CVSS and two with *critical* impact. For example, the 7th IoC of the table shows a critical Cisco router vulnerability patched and published three days before its inclusion on NVD. Finally, since not all occurrences are patched at disclosure time, some actionable IoCs contain suggested configurations to avoid exploitations. As an example, the last row in the table shows a WordPress exploit with suggested remediations.

These results show the edge obtained by using Twitter as a security data source. A SOC analyst using SYNAPSE would obtain timely and relevant data about patches to known vulnerabilities, thus possibly reducing the vulnerable system's exposure time. Further, the results also show that vendors publish important impact data before it is included in NVD.

7. SOC integration

An essential aspect of threat intelligence tools such as SYNAPSE is the integration in a SOC. In the following, we describe practical issues related to this integration.

7.1. Adversarial model

When using Twitter as a cybersecurity information source, it is important to consider what would happen if some of the monitored accounts fall under the control of the adversary. In a nutshell, two things can happen: (1) the adversary may not tweet about the threats he is interested in exploiting using the accounts he controls; or (2) the adversary may create tweets with false threats to make SOC analysts waste their time in solving potential non-existent problems. Both attacks should not create significant problems as long as the amount of accounts controlled by the adversary is relatively small, and the analysts take into account the reputation of the accounts monitored by the system.

7.2. Training the system

Our approach requires the creation of labelled datasets for training the classifiers. To do that, the SOC analysts need first to configure the keywords defining the infrastructure. A second configuration step is to define the Twitter accounts that will be monitored.

After those two steps, the system should present all filtered tweets as if they are important, and a button for the analyst to mark a tweet as "irrelevant".[2] Notice that, to avoid bias, it is relevant to inform the analysts that the system is under training. When enough positively-labelled tweets are collected, the classifiers can be trained in background and then placed in operation.

It is expected that the classifier's performance decreases with time, as the operational data gets progressively different from the training data. To maintain the utility of the classifiers in use, it is essential to minimise this effect. Incremental learning is a technique that can be used for this purpose, where the classifier's model is continuously trained with new labelled examples [74]. By training the model with the latest events, it is continuously

[2] The "irrelevant" button must always be available, even when the system is not being trained, in order to collect wrongly classified tweets for future retraining.

Table 7
Examples of tweets whose content has high impact or important actionability.

Cluster exemplar text (without links)	#	Asset	Date	Action	Threat type	Notes
#ubuntu #security : USN-3006-1: Linux kernel vulnerabilities	19	Linux	10/06	Patch	Vulnerabilities	Several vulnerabilities patched, some were not yet included on NVD, half with CVSS $\geq$ 7.0
High - USN-3016-1 - Linux kernel vulnerabilities A security issue affects these releases of Ubuntu and its derivat	12	Linux	27/06	Patch	Vulnerabilities	Several vulnerabilities patched, some were not yet included on NVD, half with CVSS $\geq$ 7.5
Microsoft Internet Explorer CVE-2016-3205 Scripting Engine Remote Memory Corruption Vulnerability Type: Vulnerabil	8	IE	14/06 (1)	Config	Vulnerability, remote	This cluster contains various threats with CVSS $\geq$ 7.5; configurations are suggested to mend the issue before it is patched
#CISCO fixed severe #vulnerabilities in Network Management and #Security Products #SecurityAffairs	9	Cisco	30/06 (2)	Patch	Vulnerabilities	Patch for critical vulnerabilities (CVSS $\geq$ 8.6) announced on Twitter before being published on NVD
Bugtraq: [security bulletin] - Linux Kernel Flaw, ASN.1 DER decoder for x509 certificate DER	6	Linux	06/06 (21)	Patch	Certificate	A highly important Linux kernel flaw (CVSS 7.8) was disclosed 21 days before being included in NVD
Vuln: Oracle Java SE and JRockit CVE-2016-3427 Remote Security Vulnerability Vulnerable:Red Hat Enterprise Linux	21	Oracle	05/07	Patch	Vulnerability, remote	This cluster contains three different threats (one with CVSS 9.0); patches are available
Bugtraq: Cisco Security Advisory: Cisco RV110W, RV130W, and RV215W Routers Arbitrary Code Execution Vulnerability	5	Cisco	15/06 (3)	Patch	Vulnerability, execution	A critical vulnerability (CVSS 9.8) was disclosed and patched before its inclusion on NVD
Bugtraq: Cisco Security Advisory: Cisco Products IPv6 Neighbour Discovery Crafted Packet Denial of Service	5	Cisco	25/05 (4)	Patch	Denial of service	A high impact vulnerability (CVSS 7.5) was disclosed and patched before its inclusion on NVD
#ubuntu #security : USN-2975-2: Linux kernel (Trusty HWE) vulnerability	5	Linux	16/05 (42)	Patch	Vulnerability	A high impact vulnerability (CVSS 7.8) was disclosed and patched before its inclusion on NVD (42 days in advance)
Bugtraq: Wordpress Levo-Slideshow 2.3 - Arbitrary File Upload Vulnerability	9	WPress	07/06	Config	Vulnerability	An exploit is provided; a software correction is suggested

adapted to changes in input format (in this case, changes in tweet format or language).

Another possibility is to replace the model with a new model trained with only the latest data, *e.g.*, the last three months of tweets. This way the model is periodically adapted to the current threat landscape, so that old data will not impact the classifier's quality.

7.3. Changing keywords and monitored accounts

Adding or removing keywords from the datasets require retraining the classifier. Removing a keyword requires removing the tweets that were filtered by this keyword and retrain the model without them. To add a keyword, one needs first to complement the existing labelled dataset (in the same way as described before) with tweets related to the new keyword, and then retrain the model with the reformulated data set. Changing the set of monitored Twitter accounts is not a burden for the system since the structure of threat descriptions is expected to be similar across all security accounts. The datasets employed in our experimental evaluation consider this possibility.

7.4. SYNAPSE integration with a real SOC/SIEM

In the following we present some highlights of SYNAPSE integration with the SIEM of a nation-wide electric power utility. The first step was to provide a dashboard so the SOC operators could visually inspect the latest security events. We provide access SYNAPSE's web dashboard [36].

Figs. 8 and 9 present the developed dashboard, representing respectively the enriched tweet clusters and the collected data volume. However, SOC operators require data centralisation in the SIEM to guarantee streamlined workflows; data originating from various sources has to be correlated and their attention cannot be dispersed throughout multiple dashboards. Therefore, we developed a connector to place SYNAPSE's IoCs in the SIEM (which was trivial as SYNAPSE was designed considering integration with threat intelligence tools), and the SOC operators developed a new SIEM dashboard so they could observe SYNAPSE data.

Once SYNAPSE was directly connected to the SIEM we raised our attention to how to use tweets together with infrastructure events. Discussing with the SOC operators how the data could improve the SIEM capabilities, three integration actions were implemented. The first was to create a rule in the SIEM that triggered an alarm when a cluster with more than five tweets was received. Although simple, this rule selects only events likely to be of importance.

The second action complements internal events with external descriptions. The SOC includes in its infrastructure a firewall that tags identified threats with their corresponding CVE-IDs. Since SYNAPSE's IoCs include security bulletin IDs (such as CVE-IDs), the SIEM was able to match the events. The connection between internal events (firewall detected threats) and external events (tweets with security bulletins for those threats) improved the quality of SOC operation as the firewall events only mention IDs, lacking an accompanying description of the threats—something provided by the tweets.

The third action is related to prioritising security actions. Managing a large IT infrastructure raises many complex problems. One of them is updating software in many different machines with different purposes. Updating system images (composed of operating system and various software elements) implies a patching phase followed by a testing and compliance phase aimed at detecting incompatibilities with the different software in use. In practice, updating system images can take about a month,

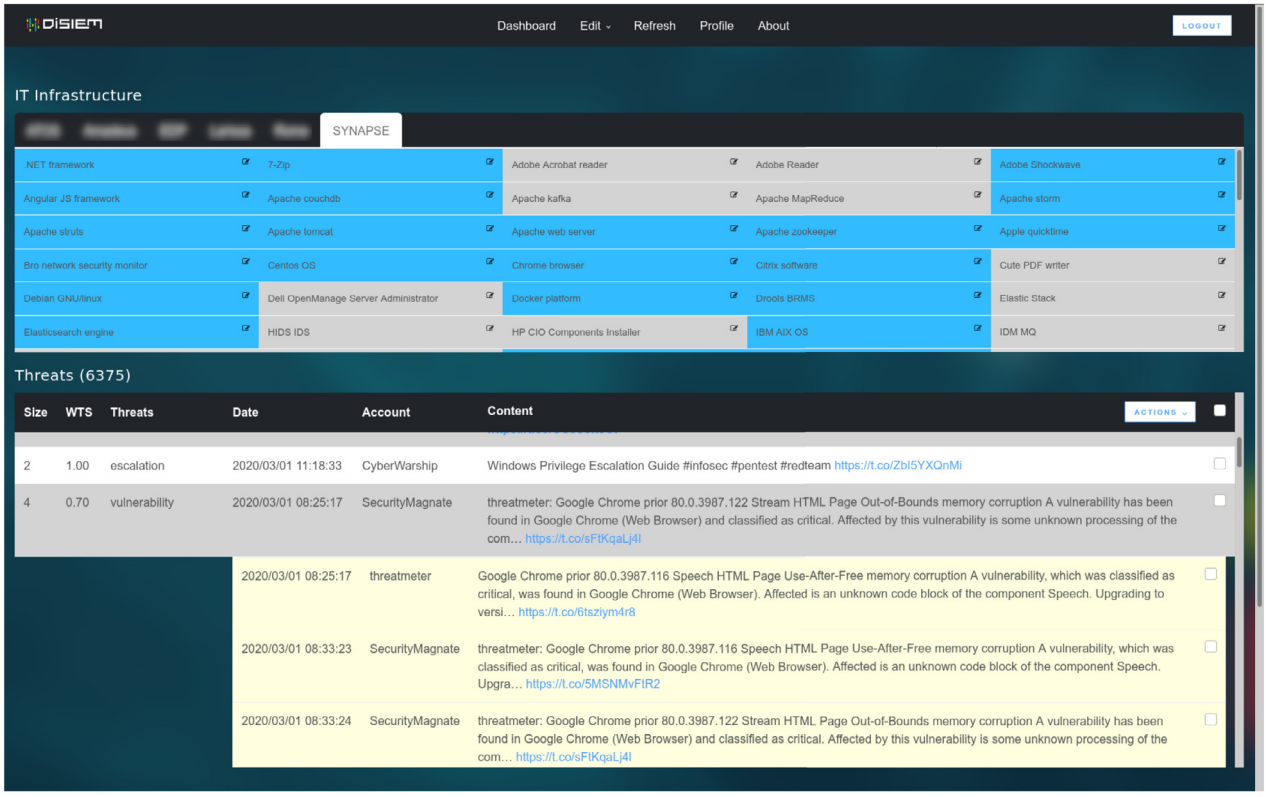

Fig. 8. An overview of SYNAPSE's dashboard. It is possible to view all collected threats (as depicted), or to select only some assets. Furthermore, each threat can be analysed only by its exemplar or in its entirety.

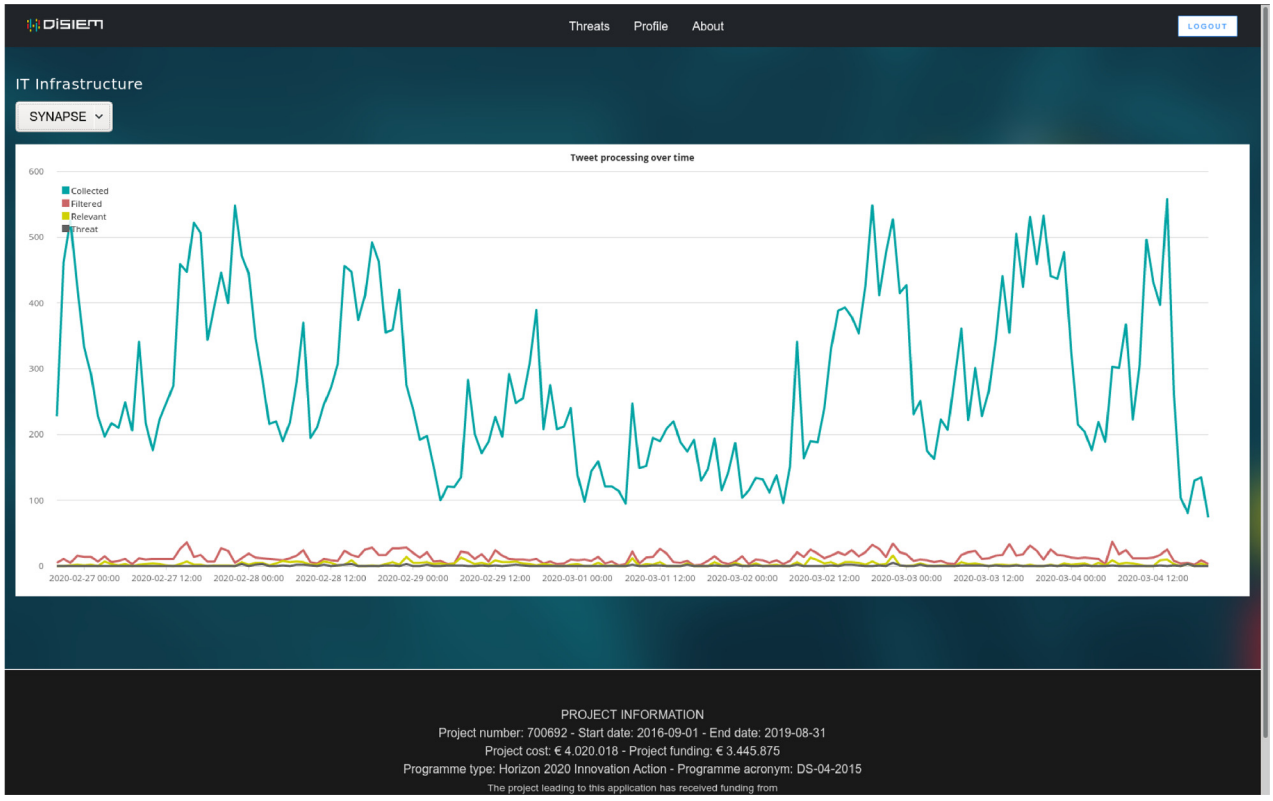

Fig. 9. The collected data volume. The lines represent: in blue, the total number of tweets collected; in red, how many refer the mentioned infrastructure assets; in yellow, the number of tweets deemed relevant for cybersecurity; and in brown, the number of threats detected by SYNAPSE (post aggregation). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

sometimes more. SYNAPSE was praised by the SOC operators as it provided them with much needed awareness of the current cyberthreats, and thus they were able to better prioritise the patching process because it was simpler for them to understand the criticality of each patch.

8. Conclusions

This paper proposes SYNAPSE, a Twitter-based streaming threat monitor for threat detection in security operation centres. It implements a pipeline that gathers tweets from a set of accounts, filters them based on the monitored infrastructure, and classify the remaining tweets as either relevant or not. Relevant tweets are grouped in dynamic clusters and presented as indicators of compromise that can be either manually inspected or fed to SIEMs and other threat intelligence tools. Results show that our system maximises the relevant information (true positive rate of 90%), minimises irrelevant information (false positive rate of 10%), and aggregates related information (only 21% of the relevant tweets are presented). We performed an evaluation of the IoCs generated by SYNAPSE, showing that highly relevant, timely and actionable information was collected, illustrating the value of our end-to-end approach. Finally, we present how SYNAPSE was integrated with a SIEM and their events correlated, together with a set of insights for future works.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We thank André Correia for collecting and labelling the data set employed in this paper, and Eric Vial for his contributions on SYNAPSE's SIEM integration.

Funding

This work was supported by the H2020 European Project DiSIEM (H2020-700692) and by the Fundação para a Ciência e a Tecnologia (FCT) through project ThreatAdapt (FCT-FNR/0002/2018) and the LASIGE Research Unit (UIDB/00408/2020 and UIDP/00408/2020).

Appendix A. Pareto figures

A.1. Feature extraction

We used Spark's implementation of TF-IDF with default parameters, except for the feature vector size. In order to find a suitable vector size to describe the tweets, eleven values were tested: {30, 50, 80, 100, 200, 300, 500, 750, 1000, 1500, 3000}. This range covers from low to high dimensional vectors, and with it, we should be able to find an appropriate vector size for the datasets.

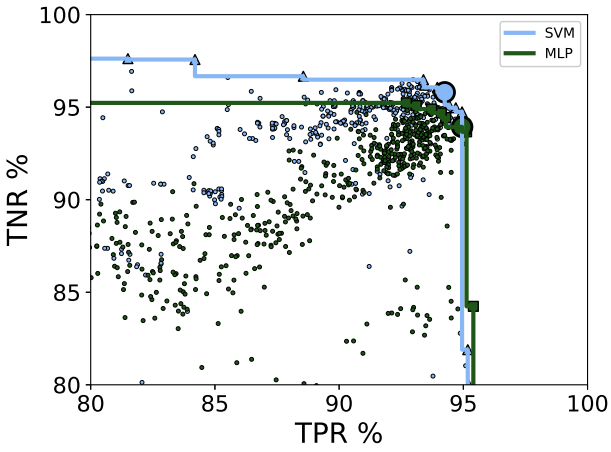

Fig. A.10. The Pareto fronts for SVM and MLP cross-validated using *D1*.

A.2. Classification

As mentioned in Section 3, two classifiers were employed: a linear SVM and an MLP Neural Network. Relevant hyper-parameters and design variables were varied to find a good design for this application. For the SVM, we varied C (the regularisation parameter) within $\{0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5\}$, and the step size (a parameter for the Stochastic Gradient Descent) within $\{0.1, 0.5, 1, 1.5, 2, 5\}$. For the MLP, the number of layers varied from 2 to 8 and the number of neurons per layer within $\{5, 7, 10, 12, 14, 16, 18, 20\}$.

Each model was evaluated through a 10-fold cross-validation procedure using dataset `D1`. The maximum number of training iterations was set to 100 for the SVM and 200 for the MLP, which were deemed to achieve parameter convergence for the range of the design parameters.

To select the best classifiers, we performed a Pareto-optimal search. For each type of classifier we plotted a Pareto front figure (Fig. A.10), with lines connecting the dominant configurations regarding *True Positive Rate* (TPR, x-axis) and *True Negative Rate* (TNR, y-axis). Each point shows the average value obtained by a specific configuration over the 10-fold cross-validation procedure. The highlighted triangular and circular points are, respectively, the dominant configurations and the configurations chosen to be used (the SVM case) in the experiments. We use the classical true positive definition: a sample labelled as positive and classified as positive; in our case, a tweet manually labelled as relevant and classified as relevant. The negative samples use the equivalent definition.

Based on this analysis, we selected the parameter configurations with the best TPR×TNR balance: those with the smallest distance to the optimum. The best SVM configuration uses a step size and C values of 0.05 and 5, respectively, and the best MLP had 5 layers with 10 neurons each. Both models use feature vectors with a size of 3000, revealing a clear advantage in using high-dimensional feature vectors.

Appendix B. Complete cluster data

Table B.8 presents the 65 IoCs largest clusters generated by SYNAPSE, as described in Section 6.2.

By running SYNAPSE's IoC generation module, each cluster was tagged with the type of threats mentioned by its tweets. The most common tags are "vulnerability" (23) and "vulnerabilities"

Table B.8
Largest generated clusters represented as IoCs.

Cluster exemplar text (without links)	#	Asset	Date	Action	Threat type	Notes
High - USN-3016-1 - Linux kernel vulnerabilities A security issue affects these releases of Ubuntu and its derivat	12	Linux	27/06	Patch	Vulnerabilities	Several vulnerabilities patched, some were not yet included on NVD, half with CVSS $\geq$ 7.5
#0daytoday #Sun Secure Global Desktop and Oracle Global Desktop 4.61.915 - ShellShock Exploit [#0day #Exploit]	11	Oracle	06/06	None	Exploit, 0day	An exploit is presented; an expert might use this data for protection
#ubuntu #security : USN-2993-1: Firefox vulnerabilities	10	Firefox	09/06 (4)	Patch	Vulnerabilities	Patches are available for vulnerabilities, half with CVSS $\geq$ 8.8
Bugtraq: CM Ad Changer 1.7.7 Wordpress Plugin - Cross Site Scripting Web Vulnerability	10	WPress	13/06	Patch	Vulnerability	A patch is available; an exploit is provided
Bugtraq: Wordpress Levo-Slideshow 2.3 - Arbitrary File Upload Vulnerability	9	WPress	07/06	Config	Vulnerability	An exploit is provided; a software correction is suggested
Bugtraq: Oracle Orakill.exe Buffer Overflow	9	Oracle	14/06	Patch	Buffer overflow	A patch is available; an exploit is provided
#CISCO fixed severe #vulnerabilities in Network Management and #Security Products #SecurityAffairs	9	Cisco	30/06 (2)	Patch	Vulnerabilities	Patch for critical vulnerabilities (CVSS $\geq$ 8.6) announced on Twitter before being published on NVD
#ubuntu #security : USN-3016-1: Linux kernel vulnerabilities	8	Linux	27/06	Patch	Vulnerabilities	Several vulnerabilities patched, some were not yet included on NVD, half of the vulns with CVSS $\geq$ 7.5
Microsoft Internet Explorer CVE-2016-3205 Scripting Engine Remote Memory Corruption Vulnerability Type: Vulnerabil	8	IE	14/06 (1)	Config	Vulnerability, remote	This cluster contains various threats with CVSS $\geq$ 7.5; configurations are suggested to solve the issue before it is patched
NA - CVE-2016-2825 - Mozilla Firefox before 47.0 allows remote... Mozilla Firefox before 47.0 allows remote attack	8	Firefox	13/06	Patch	Attack, remote	A patch is available for a vulnerability with CVSS 6.5
Vuln: Oracle Java SE and JRockit CVE-2016-3427 Remote Security Vulnerability Vulnerable:Red Hat Enterprise Linux	21	Oracle	05/07	Patch	Vulnerability, remote	This cluster contains three different threats (one with CVSS 9.0); patches are available
#ubuntu #security : USN-3006-1: Linux kernel vulnerabilities	19	Linux	10/06	Patch	Vulnerabilities	Several vulnerabilities patched, some were not yet included on NVD, half with CVSS $\geq$ 7.0
#0daytoday #Cisco EPC 3928 - Multiple Vulnerabilities [webapps #exploits #Vulnerabilities #0day #Exploit]	16	Cisco	07/06	None	Exploit, vulnerabilities, 0day	An exploit is presented; an expert might use this data for protection (half of the vulns with CVSS $\geq$ 7.5)
#0daytoday #Joomla En Masse com_enmasse Component 5.1 - 6.4 - SQL Injection Vulnerability [#0day #Exploit]	12	Joomla	15/06	None	SQL injection, exploit, injection, vulnerability, 0day	An exploit is presented; an expert might use this data for protection
#0daytoday #WordPress Social Stream Plugin 1.5.15 - wp_options Overwrite Vulnerability [#0day #Exploit]	8	WPress	14/06	Patch	Exploit, vulnerability, 0day	A patch is available; an exploit is provided
Microsoft Internet Explorer 11 Garbage Collector Attribute Type Confusion #exploit	8	IE	18/06	Patch	Exploit	A patch is available for a vulnerability with CVSS 8.8; an exploit is provided
CVE-2016-1388 Cisco Prime Network Analysis Module (NAM) before 6.1(1) patch.6.1-2-final and 6.2.x before 6.2(1) an	8	Cisco	03/06	Patch		This cluster contains 4 threats, 3 with CVSS $\geq$ 7.8; patches are available
#Oracle #Linux 6 : #openssl (ELSA-2016-0996) #Nessus	8	Linux	16/05	Patch		This cluster contains seven threats: 3 critical (CVSS 9.8) and 3 high (CVSS 7.5); patches are available
Vuln: Linux Kernel Multiple Local Memory Corruption Vulnerabilities	7	Linux	08/07	Patch	Vulnerabilities	Patches are available for vulnerabilities with CVSS 7.1 and 7.8
Vuln: Linux Kernel CVE-2016-0723 Local Race Condition Vulnerability	7	Linux	08/07	Patch	Vulnerability	A patch is available for vulnerability with CVSS 6.8
Vuln: Linux kernel CVE-2013-7446 Use After Free Denial of Service Vulnerability	7	Linux	05/07	Patch	Denial of service, vulnerability	A patch is available for vulnerability with CVSS 5.3
Bugtraq: Cisco Security Advisory: Cisco Firepower System Software Static Credential Vulnerability	7	Cisco	29/06 (3)	Patch	Vulnerability	A patch is available for vulnerability with CVSS 8.6

(*continued on next page*)

Table B.8 (*continued*).

Cluster exemplar text (without links)	#	Asset	Date	Action	Threat type	Notes
#0daytoday #WordPress Ultimate Membership Pro Plugin 3.3 - SQL Injection Vulnerability [#0day #Exploit]	7	WPress	29/06	Patch	SQL injection, exploit, injection, vulnerability, 0day	A patch is available; an exploit is provided
#0daytoday #Google Chrome - GPU Process MailboxManagerImpl Double-Read Vulnerability [#0day #Exploit]	7	Chrome	15/06	Patch	Exploit, vulnerability, 0day	A patch is available; an exploit is provided
#0daytoday #WordPress Gravity Forms Plugin 1.8.19 - Arbitrary File Upload Exploit [#0day #Exploit]	7	WPress	17/06	None	Exploit, 0day	An exploit is presented; an expert might use this data for protection
#0daytoday #WordPress Uncode Theme 1.3.1 - Arbitrary File Upload Exploit [webapps #exploits #0day #Exploit]	7	WPress	06/06	N/A	Exploit, 0day	All tweet links are broken; nothing can be inferred
#0daytoday #WordPress Double Opt-In for Download Plugin 2.0.9 - SQL Injection Vulnerability [#0day #Exploit]	7	WPress	06/06	Patch	SQL injection, exploit, injection, vulnerability, 0day	A patch is available; an exploit is provided
#cybersecurity Hackers offering Microsoft Windows zero-day exploit for $90000 #infosec	7	Windows	01/06	N/A	Exploit	Just informative tweets
#Oracle ATS Arbitrary File Upload #PacketStorm	7	Oracle	24/05	None		An exploit is presented; an expert might use this data for protection
Vuln: Linux Kernel 'usb/core/hub.c' NULL Pointer Dereference Denial of Service Vulnerability	6	Linux	08/07	Patch	Denial of service, vulnerability	A patch is available for vulnerability with CVSS 6.8
#0daytoday #Linux - ecryptfs and /proc/$pid/environ Privilege Escalation Vulnerability [#0day #Exploit]	6	Linux	21/06 (6)	None	Exploit, escalation, vulnerability, 0day	An exploit is early presented for a vulnerability with CVSS 7.8; an expert might use this data for protection
CVE-2016-3221 The kernel-mode drivers in Microsoft Windows Vista SP2, Windows Server 2008 SP2 and R2 SP1, Windows	6	Windows	16/06	Patch		A patch is available for a vulnerability with CVSS 7.8
NA - CVE-2016-3201 - Microsoft Windows 8.1, Windows Server 2012 Gold... Microsoft Windows 8.1, Windows Server 2012	6	Windows	16/06	Patch		A patch is available for a vulnerability with CVSS 6.5
#0daytoday #Joomla com_affiliatetracker - SQL Injection Vulnerability [webapps #exploits #Vulnerability #0day	6	Joomla	13/06	N/A	SQL injection, exploit, injection, vulnerability, 0day	All tweet links are broken; nothing can be inferred
[shellcode] - #Linux x86_64 Shellcode Null-Free Reverse TCP Shell #ExploitDB	6	Linux	16/06	None	Exploit	An exploit is presented; an expert might use this data for protection
Bugtraq: [security bulletin] - Linux Kernel Flaw, ASN.1 DER decoder for x509 certificate DER	6	Linux	06/06 (21)	Patch	Certificate	A highly important Linux kernel flaw (CVSS 7.8) was disclosed 21 days before being included in NVD
[webapps] - WordPress WP Mobile Detector Plugin 3.5 - Arbitrary File Upload: WordPress WP Mobile Detector Plu...	6	WPress	06/06	Patch		A patch is available; an exploit is provided
Bugtraq: Cisco Security Advisory: Cisco Prime Network Analysis Module IPv6 Denial of Service Vulnerability	6	Cisco	01/06 (1)	Patch	Denial of service, vulnerability	A patch is available for a vulnerability with CVSS 5.3
Bugtraq: Cisco Security Advisory: Cisco Prime Network Analysis Module Unauthenticated Remote Code #bugtraq	6	Cisco	01/06 (1)	Patch	Remote	A patch is available for a critical vulnerability with CVSS 9.8
WordPress Patches Zero Day in WP Mobile Detector Plugin #InfoSec	6	WPress	03/06	Patch	Zero day	A patch is available
CVE-2016-1381 Memory leak in Cisco AsyncOS 8.5 through 9.0 before 9.0.1-162 on Web Security Appliance (WSA) device	6	Cisco	25/05	Patch	Leak	A patch is available for a vulnerability with CVSS 7.5

(*continued on next page*)

Table B.8 (*continued*).

Cluster exemplar text (without links)	#	Asset	Date	Action	Threat type	Notes
Oracle E-Business Suite Vulnerabilities Related To Common Components Oracle E-Business Intelligence component in O	6	Oracle	23/05	None	Vulnerabilities	The tweet links provide no useful information
NA - cisco-sa-20160518-wsa4 - Cisco Web Security Appliance Connection Denial of Service Vulnerability A vulnerabil	6	Cisco	18/05 (6)	Patch	Denial of service, vulnerability	A high impact vulnerability (CVSS 7.5) was disclosed and patched before its inclusion on NVD
#ubuntu #security : USN-2947-1: Linux kernel vulnerabilities	6	Linux	06/04	Patch	Vulnerabilities	A patch is available to solve multiple vulnerabilities, one of them critical (CVSS 9.8)
Vuln: Cisco Video Communication Server and Expressway CVE-2016-1444 Authentication Bypass Vulnerability	5	Cisco	08/07	Patch	Vulnerability	A patch is available for a vulnerability with CVSS 6.5
Vuln: Google Chrome Prior to 49.0.2623.75 Multiple Security Vulnerabilities	5	Chrome	06/07	Patch	Vulnerabilities	A patch is available to solve multiple high to critical vulnerabilities (5 with CVSS 8.8 and 5 with CVSS 9.8)
[webapps] - WordPress Real3D FlipBook Plugin - Multiple Vulnerabilities: WordPress Real3D FlipBook Plugin - M...	5	WPress	04/07	None	Vulnerabilities	An exploit is presented; an expert might use this data for protection
Vuln: Linux Kernel 'btrfs/inode.c' Information Disclosure Vulnerability	5	Linux	05/07	Patch	Vulnerability	A patch is available for a vulnerability with CVSS 4.0
Medium - CVE-2016-5835 - WordPress before 4.5.3 allows remote attackers... WordPress before 4.5.3 allows remote at	5	WPress	29/06	Patch	Attack, remote	A patch is available for a vulnerability with CVSS 7.5
#vulnerability #security : WordPress Contus Video Comments 1.0 File Upload	5	WPress	22/06	None	Vulnerability	An exploit is presented; an expert might use this data for protection
[webapps] - WordPress Ultimate Product Catalogue Plugin 3.8.1 - Privilege Escalation: WordPress Ultimate Produc...	5	WPress	20/06	Patch	Escalation	A patch is available; an exploit is provided
#0daytoday #WordPress Premium SEO Pack 1.9.1.3 - wp_options Overwrite Exploit [webapps #exploits #0day #Exploit]	5	WPress	21/06	None	Exploit, 0day	An exploit is presented; an expert might use this data for protection
CVE-2016-0200 Microsoft Internet Explorer 9 through 11 allows remote attackers to execute arbitrary code or cause	5	IE	16/06	Patch	Attack, remote	The cluster contains two different threats; patches are available to solve 4 vulns with CVSS 8.8
Bugtraq: Cisco Security Advisory: Cisco RV110W, RV130W, and RV215W Routers Arbitrary Code Execution Vulnerability	5	Cisco	15/06 (3)	Patch	Vulnerability, execution	A critical vulnerability (CVSS 9.8) was disclosed and patched before its inclusion on NVD
#0daytoday #WordPress Newspaper Theme 6.7.1 - Privilege Escalation Exploit [webapps #exploits #0day #Exploit]	5	WPress	06/06	Patch	Exploit, escalation, 0day	A patch is available; an exploit is provided
[webapps] - WordPress Simple Backup Plugin 2.7.11 - Multiple Vulnerabilities: WordPress Simple Backup Plugin ...	5	WPress	06/06	None	Vulnerabilities	An exploit is presented; an expert might use this data for protection
CVE-2016-1701 The Autofill implementation in Google Chrome before 51.0.2704.79 mishandles the interaction between	5	Chrome	06/06	Patch		All tweets refer a different vulnerability, all from the same date, all with CVSS $\geq$ 7.5; patches are available
#0daytoday #WordPress WP PRO Advertising System Plugin 4.6.18 - SQL Injection Exploit [#0day #Exploit]	5	WPress	06/06	None	SQL injection, exploit, injection, 0day	An exploit is presented; an expert might use this data for protection
[webapps] - WordPress Creative Multi-Purpose Theme 9.1.3 - Stored XSS: WordPress Creative Multi-Purpose Theme...	5	WPress	06/06	Patch	XSS	A patch is available; an exploit is provided
#WordPress WP Mobile Detector 3.5 Shell Upload #PacketStorm	5	WPress	04/06	Patch		A patch is available; an exploit is provided
#hackers Selling Unpatched Microsoft Windows Zero-Day Exploit for $90.000	5	Windows	03/06	N/A	Exploit	Just informative tweets
Oracle E-Business Suite Vulnerabilities Related To E-Business Intelligence Oracle E-Business Intelligence compon	5	Oracle	30/05	None	Vulnerabilities	The tweet links provide no useful information
Bugtraq: Cisco Security Advisory: Cisco Products IPv6 Neighbour Discovery Crafted Packet Denial of Service	5	Cisco	25/05 (4)	Patch	Denial of service	A high impact vulnerability (CVSS 7.5) was disclosed and patched before its inclusion on NVD
#ubuntu #security : USN-2975-2: Linux kernel (Trusty HWE) vulnerability	5	Linux	16/05 (42)	Patch	Vulnerability	A high impact vulnerability (CVSS 7.8) was disclosed and patched before its inclusion on NVD (42 days in advance)
Bugtraq: Cisco Security Advisory: Cisco Web Security Appliance HTTP POST Denial of Service Vulnerability	5	Cisco	18/05 (6)	Patch	Vulnerability	A high impact vulnerability (CVSS 7.5) was disclosed and patched before its inclusion on NVD

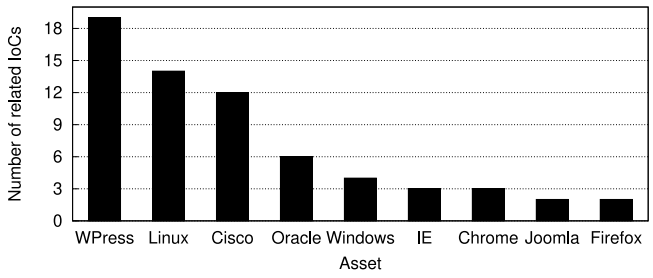

Fig. A.11. Number of IoCs for each asset.

(13), reflecting that most threats are related to vulnerability disclosure. Other two common tags are "exploit" (18) and "0day" (15) (or "zero-day"), which indicate exploitable vulnerabilities. Less used tags include "remote" (6) (remote execution attacks), "denial of service" (6), "SQL injection" (5), and "Buffer overflow" (4) (or BO).

Out of the 13 assets composing the hypothetical IT infrastructure described in Table 3, only 9 (∼ 70%) had related IoCs. The distribution of IoCs over the assets is shown in Fig. A.11. WordPress is the asset with more related IoCs (19), followed by Linux (14) and Cisco (12). All analysed IoCs mentioned a single asset.

References

[1] Cyber intelligence | SenseCy, 2018, https://www.sensecy.com/, [Accessed 13-06-2018].
[2] Threat analysis - Intelligence | Monitor - Track cyber threats, 2018, https://www.surfwatchlabs.com/threat-intelligence-products/threat-analyst, [Accessed 13-06-2018].
[3] R.D. Steele, Open source intelligence: What is it? why is it important to the military, Amer. Intell. J. 17 (1) (1996).
[4] Threatpost | The first stop for security news, 2018, https://threatpost.com/feed/, [Accessed 13-06-2018].
[5] How people use Twitter in general - American press institute, 2018, https://www.americanpressinstitute.org/publications/reports/survey-research/how-people-use-twitter-in-general, [Accessed 13-06-2018].
[6] M. Cataldi, et al., Emerging topic detection on twitter based on temporal and social terms evaluation, in: Proceedings of the 10th MDM/KDD, 2010.
[7] T. Sakaki, et al., Earthquake shakes twitter users: Real-time event detection by social sensors, in: Proceedings of the 19th WWW, 2010.
[8] Y. Kim, K. Shim, TWILITE: A recommendation system for Twitter using a probabilistic model based on latent Dirichlet allocation, Inf. Syst. 42 (2014).
[9] M.G. Armentano, et al., Followee recommendation based on text analysis of micro-blogging activity, Inf. Syst. 38 (8) (2013).
[10] B. Li, et al., Discovering public sentiment in social media for predicting stock movement of publicly listed companies, Inf. Syst. 69 (2017).
[11] F. Alves, et al., Follow the blue bird: a study on threat data published on twitter, in: ESORICS 2020, 2020, (Forthcoming).
[12] N. McNeil, et al., PACE: Pattern accurate computationally efficient bootstrapping for timely discovery of cyber-security concepts, in: Proceedings of the 12th ICMLA, 2013.
[13] R. Campiolo, et al., Evaluating the utilization of Twitter messages as a source of security alerts, in: Proceedings of the 28th ACM SAC, 2013.
[14] O.C. Moholth, et al., Detecting cyber security vulnerabilities through reactive programming, in: Proceedings of the 52nd HICSS, 2019.
[15] C. Sabottke, et al., Vulnerability disclosure in the age of social media: exploiting twitter for predicting real-world exploits, in: Proceedings of the 24th USENIX Security Symp., 2015.
[16] C. Sauerwein, et al., Threat intelligence sharing platforms: An exploratory study of software vendors and research perspectives, in: Towards Thought Leadership in Digital Transformation: 13. Internationale Tagung Wirtschaftsinformatik, 2017.
[17] N. Serketzis, et al., Actionable threat intelligence for digital forensics readiness, Inf. Comput. Secur. (2019).
[18] Spiderfoot, open source intelligence automation, 2018, http://spiderfoot.net/, [Accessed 13-06-2018].
[19] IntelMQ, 2018, http://github.com/certtools/intelmq/, [Accessed 13-06-2018].
[20] S. Mittal, et al., Cybertwitter: Using twitter to generate alerts for cybersecurity threats and vulnerabilities, in: Proceedings of the 8th IEEE/ACM ASONAM, 2016.
[21] Q. Le Sceller, et al., Sonar: Automatic detection of cyber security events over the twitter stream, in: Proceedings of the 12th ARES, 2017.
[22] A. Ritter, et al., Weakly supervised extraction of computer security events from twitter, in: Proceedings of the 24th WWW, 2015.
[23] S. Trabelsi, et al., Mining social networks for software vulnerabilities monitoring, in: Proceedings of the NTMS, 2015.
[24] N. Dionísio, et al., Cyberthreat detection from twitter using deep neural networks, in: Proceedings of the IJCNN, 2019.
[25] V. Behzadan, et al., Corpus and deep learning classifier for collection of cyber threat indicators in twitter stream, in: Proceedings of the 2018 IEEE Big Data, 2018.
[26] S. Yagcioglu, et al., Detecting cybersecurity events from noisy short text, 2019, arXiv:1904.05054.
[27] A. Sapienza, et al., Early warnings of cyber threats in online discussions, in: Proceedings of the 2017 IEEE ICDMW, 2017.
[28] A. Tonon, et al., ArmaTweet: detecting events by semantic tweet analysis, in: Proceedings of the European Semantic Web Conference, 2017.
[29] K.-C. Lee, et al., Sec-buzzer: cyber security emerging topic mining with open threat intelligence retrieval and timeline event annotation, Soft Comput. 21 (11) (2017).
[30] A. Okutan, et al., Predicting cyber attacks with bayesian networks using unconventional signals, in: Proceedings of the 12th CISRC, 2017.
[31] R.P. Khandpur, et al., Crowdsourcing cybersecurity: Cyber attack detection using social media, in: Proceedings of the 26th ACM CIKM, 2017.
[32] B.-D. Le, et al., Gathering cyber threat intelligence from twitter using novelty classification, in: Proceedings of the 18th CW, 2019.
[33] A. Sapienza, et al., Discover: Mining online chatter for emerging cyber threats, in: Companion Proceedings of the the Web Conference 2018, 2018.
[34] IBM QRadar SIEM, 2019, https://www.ibm.com/pt-en/marketplace/ibm-qradar-siem, [Accessed 15-02-2019].
[35] MISP - Open source threat intelligence platform & open standards for threat information sharing, 2018, http://www.misp-project.org/, [Accessed 13-06-2018].
[36] SYNAPSE Dashboard, https://disiem-otd.lasige.di.fc.ul.pt/index, Username: synapse; password: 5w#*0uV5.
[37] X. Liu, et al., Event evolution model for cybersecurity event mining in tweet streams, Inform. Sci. (2020).
[38] T. Ji, et al., Feature driven learning framework for cybersecurity event detection, in: IEEE/ACM ASONAM, 2019.
[39] A. Bose, et al., A novel approach for detection and ranking of trendy and emerging cyber threat events in Twitter streams, in: Proceedings of the 2019 IEEE/ACM ASONAM, 2019.
[40] A. Niakanlahiji, et al., Iocminer: Automatic extraction of indicators of compromise from twitter, in: Big Data, 2019.
[41] P. Indyk, R. Motwani, Approximate nearest neighbors: towards removing the curse of dimensionality, in: Proceedings of the 30th ACM STOC, 1998.
[42] M. Ester, et al., A density-based algorithm for discovering clusters in large spatial databases with noise, in: Proceedings of the KDD, 1996.
[43] C.C. Aggarwal, Data Mining: The Textbook, Springer, 2015.
[44] J.A. Silva, et al., Data stream clustering: A survey, ACM Comput. Surv. 46 (1) (2013).
[45] T. Zhang, et al., BIRCH: an efficient data clustering method for very large databases, in: ACM Sigmod Record, 1996.
[46] S. Guha, et al., Clustering data streams, in: Proceedings 41st FOCS, 2000.
[47] A. Zhou, et al., Tracking clusters in evolving data streams over sliding windows, Knowl. Inf. Syst. 15 (2) (2008).
[48] M.L. Mathews, et al., A collaborative approach to situational awareness for cybersecurity, in: Proceedings of the 8th CollaborateCom, 2012.
[49] X. Liao, et al., Acing the IOC game: Toward automatic discovery and analysis of open-source cyber threat intelligence, in: Proceedings of the 23rd ACM CCS, 2016.
[50] Z. Zhu, T. Dumitras, FeatureSmith: Automatically engineering features for malware detection by mining the security literature, in: Proceedings of the 23rd ACM CCS, 2016.
[51] W. Feng, et al., STREAMCUBE: hierarchical spatio-temporal hashtag clustering for event exploration over the twitter stream, in: Proceedings of the 31st ICDE, 2015.
[52] F. Saki, N. Kehtarnavaz, Online frame-based clustering with unknown number of clusters, Pattern Recognit. 57 (2016) 70–83.
[53] L. Shou, et al., Sumblr: continuous summarization of evolving tweet streams, in: Proceedings of the 36th ACM SIGIR, 2013.
[54] J. Leskovec, et al., Mining of Massive Datasets, Cambridge University Press, 2014.
[55] K. Weinberger, et al., Feature hashing for large scale multitask learning, in: Proceedings of the 26th ICML, 2009.
[56] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995).
[57] F. Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain, Psychol. Rev. 68 (6) (1958).

[58] D.E. Rumelhart, et al., Learning Internal Representations by Error Propagation, Tech. rep., DTIC, 1985.
[59] Apache spark, 2018, http://spark.apache.org, [Accessed 13-06-2018].
[60] C.C. Aggarwal, et al., A framework for clustering evolving data streams, in: Proceedings of the 29th VLDB, 2003.
[61] Introduction to STIX, 2018, https://oasis-open.github.io/cti-documentation/stix/intro, [Accessed 13-06-2018].
[62] MISP data models, 2018, http://www.misp-project.org/datamodels/, [Accessed 13-06-2018].
[63] N. Marwaha, System and method for providing common event format using alert index, 2006, US Patent 7, 139, 938.
[64] MISP taxonomies, 2018, http://www.misp-project.org/datamodels/, [Accessed 13-06-2018].
[65] O. Arbelaitz, et al., An extensive comparative study of cluster validity indices, Pattern Recognit. 46 (1) (2013).
[66] I. Guyon, et al., Clustering: Science or art, in: Proceedings of the 9th NIPS Workshop on Clustering Theory, 2009.
[67] M.J. Zaki, et al., Data Mining and Analysis: Fundamental Concepts and Algorithms, Cambridge University Press, 2014.
[68] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the 5th BSMSP, 1967.
[69] A.K. Jain, Data clustering: 50 years beyond K-means, Pattern Recognit. Lett. 31 (8) (2010).
[70] R. Tibshirani, et al., Estimating the number of clusters in a data set via the gap statistic, J. R. Stat. Soc. B 63 (2) (2001).
[71] Alienvault OTX, the world's first truly open threat intelligence community, 2019, https://otx.alienvault.com/, [Accessed 13-02-2019].
[72] Logstash: Collect, parse, transform logs, 2018, https://www.elastic.co/products/logstash, [Accessed 13-06-2018].
[73] Common vulnerability scoring system SIG, 2018, https://www.first.org/cvss/, [Accessed 13-06-2018].
[74] X. Geng, K. Smith-Miles, Incremental learning, in: Encyclopedia of Biometrics, Springer, 2015.