Article

The Influences of Feature Sets on the Detection of Advanced Persistent Threats

Katharina Hofer-Schmitz ^{1,*} , Ulrike Kleb ² and Branka Stojanović ¹

¹ DIGITAL—Institute for Information and Communication Technologies, JOANNEUM RESEARCH Forschungsgesellschaft mbH, 17 Steyrergasse, 8010 Graz, Austria; branka.stojanovic@joanneum.at

² POLICIES—Institute for Economic and Innovation Research, JOANNEUM RESEARCH Forschungsgesellschaft mbH, 59 Leonhardstraße, 8010 Graz, Austria; ulrike.kleb@joanneum.at

* Correspondence: katharina.hofer-schmitz@joanneum.at

Abstract: This paper investigates the influences of different statistical network traffic feature sets on detecting advanced persistent threats. The selection of suitable features for detecting targeted cyber attacks is crucial to achieving high performance and to address limited computational and storage costs. The evaluation was performed on a semi-synthetic dataset, which combined the CICIDS2017 dataset and the Contagio malware dataset. The CICIDS2017 dataset is a benchmark dataset in the intrusion detection field and the Contagio malware dataset contains real advanced persistent threat (APT) attack traces. Several different combinations of datasets were used to increase variety in background data and contribute to the quality of results. For the feature extraction, the CICflowmeter tool was used. For the selection of suitable features, a correlation analysis including an in-depth feature investigation by boxplots is provided. Based on that, several suitable features were allocated into different feature sets. The influences of these feature sets on the detection capabilities were investigated in detail with the local outlier factor method. The focus was especially on attacks detected with different feature sets and the influences of the background on the detection capabilities with respect to the local outlier factor method. Based on the results, we could determine a superior feature set, which detected most of the malicious flows.

Keywords: APT; local outlier detection; feature selection; statistical network traffic features

Citation: Hofer-Schmitz, K.; Kleb, U.; Stojanović, B. The Influences of Feature Sets on the Detection of Advanced Persistent Threats. *Electronics* **2021**, *10*, 704. <https://doi.org/10.3390/electronics10060704>

Academic Editors: Changhoon Lee, Yu Chen and Jake (Jaeik) Cho

Received: 15 January 2021

Accepted: 14 March 2021

Published: 17 March 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Cyber attacks such as hacking, phishing and data breaches are posing an ever-increasing threat to different organizations, regardless of their size. An increasing number of actors and techniques are used to avoid such threats [1]. The ongoing changes in response to the COVID-19 pandemic have also had a major impact on cyber security. For example, the FBI stated that there has been a 300% increase in reported cyber crimes [2].

One of the most dangerous cyber-attack-types is called the advanced persistent threat (APT). It targets specific organizations, government institutions and commercial enterprises [3]. The name—APT—describes these attacks quite precisely [3,4]:

- **Advanced:** The attacks are goal-oriented, and performed by highly organized, advanced and well-resourced attacker groups using adaptive tools.
- **Persistent:** The goal of those attacks is not rapid damage. Such attacks are persistent and the attackers tend to stay undetected as long as possible in the system, in order to gain as much information as possible.
- **Threat:** The goal of these attacks is usually to get valuable data, e.g., sensitive data, strategic information or product information. Therefore, the attacks usually lead to great damage for the victims.

According to different reports [5,6], COVID-19 information has received attention from different APT attackers. Kaspersky [5] reports that despite an increasing number of

attackers, they do not believe this implies a meaningful change in terms of TTPs (tactics, techniques and procedures). The only noticeable change seems to be that this trendy topic is used for luring victims.

FireEye Mandiant [6] reported that in 2020 22% of targeted attacks aimed at data theft driven by intellectual property or espionage end goals, while 29% most likely aimed at direct financial gain, including extortion, ransom, card theft and illicit transfers. They also reported activity from hundreds of new threat groups that have emerged in the last year, with emphasized activity originating from FIN (financial threat groups) and APT groups.

It should also be emphasized that, regardless of the increasing numbers related to actors, there is a decreasing trend in the duration of attacks [6]. In 2019, 41% of the compromises investigated by Mandiant experts had dwell times of 30 days or fewer; compare that to 31% of attacks in 2018 [6]. This decrease in duration seems to be related to detection programs. This may have positive impact on cyber risk, usually measured through the potential economic impact from cyber attacks [7]. According to Mandiant [6], additional findings should be taken into consideration, such as the continued rise in disruptive attacks (such as ransomware and cryptocurrency miners), which often have shorter dwell times than other attack types, but potentially significant economical impacts.

The impacts of some of the recent and well-known attacks can also be described and quantified through the number of affected users, type of stolen data and financial damage. For example, the EasyJet (2020) attack (<https://www.bbc.com/news/technology-52722626> accessed on 2 February 2021) affected approximately nine million customers and stolen data, including email addresses and travel details. Further, credit and debit card details were “accessed” from 2208 customers. The Capital One (2019) attack (<https://www.bbc.com/news/world-us-canada-49159859> accessed on 2 February 2021), directed against the 10th largest bank in the USA, affected 10 million individuals in the USA and 5 million individuals in Canada. Stolen data included personal information, credit score, credit limit, self-reported income, payment history and balance. The Equifax (2017) attack (<https://www.bbc.com/news/business-41192163> accessed on 2 February 2021), directed at one of the largest credit reporting companies affected 145.5 million users. Stolen data included users’ personal information (social security numbers and driver’s license numbers).

Another example of an APT attack was the Carbanak (2013–2014) attack, an attack with the goal of stealing money from financial institutions [3,8]. The attack started in 2013 and stayed undetected until 2014. The initial infection started with malware attached in emails sent as spear-phishing attacks to the employees of the target banking/financial institution. The attackers studied their victims in detail and created fake transactions in the victim’s internal database to hide the attacker’s money transfer transactions. The attack seemed to have stopped in 2015. However, it turned out later that it continued to show up in different variations throughout 2017. According to Kaspersky (<https://www.kaspersky.com/blog/billion-dollar-apt-carbanak/7519/> accessed on 2 February 2021), it is assumed that this attack caused 1 billion USD damage.

It is characteristic that these attacks are able to bypass existing security systems using signature-based or anomaly-based detection and prevention approaches. Therefore, detecting such attacks poses several challenges; see, e.g., [9]. First, APTs hide in weak signals in huge amounts of data. Moreover, those attacks are very rare events spanning over long periods of time. Data containing those attacks are therefore usually quite imbalanced. This indicates that supervised detection methods do not seem to be feasible in practice.

The review of datasets and their creation for use in APT detection literature in [10] mentions the lack of publicly available data for APT detection in network infrastructures. Most victims of an APT attack have no interest in releasing data and details about the attack. Besides, datasets used for APT detection [10] also consider existing feature construction, selection and dimensionality reduction of existing approaches. The authors state that most of the literature provides only some basic information on the features used, but does not investigate them in detail. Further, existing approaches do not consider the influences of the features used on the detection rates of their algorithms.

Therefore, the focus of this paper is to investigate the influences of statistical network traffic features—the ones most commonly used in the literature—in detail on detecting APTs with the local outlier method. Related literature is covered in Section 2. Section 3 describes the methodology, including the composition of a suitable dataset following an approach of [11] in Section 3.1. For achieving high performance and limiting computational and storage costs, this paper focuses on network flows. In order to consider the influence of the benign data on the detection, three datasets with the same attacks but injected in different time slots were created. In Section 3.2 results of the correlation analysis as well the in-depth study of the different features for the selection of suitable feature sets, are given. Basics on the local outlier detection method and on the infrastructure used are described in Section 3.3. The results of the detection capabilities for APT detection of different feature sets, including details on their local outlier scores, are presented in Section 4. Their influences are considered in detail. We contribute to a better understanding of the attacks, provide ideas to reduce network traffic for recording and give suggestions for future work in Section 5.

2. Related Work

In practice, the prevention and reactions to cyber attacks are highly correlated with the cyber risks. However, that aspect is not addressed explicitly in many papers focusing on cyber attack detection. The cyber risk aspect was discussed in [12]. The paper investigated potential challenges in using machine learning for an improvement of organizational resilience and a better understanding of cyber risks. The authors therein modeled connections and interdependencies of the system's edge components to external and internal services and systems and provided a new conceptual framework based on the grounded theory. Cyber risks were also considered in [13], which proposed a self-assessment method for the quantification of IoT cyber risks based on an empirical analysis of twelve cyber risk assessment approaches. Such an approach is especially useful in order to establish proper prevention, prediction and response to cyberthreats.

Details on advanced persistent threats for modeling those attacks [4], comparisons of different attacks and tools used in APTs [14] and the survey and defense method reviewed in [3] provide crucial insights into those targeted cyber attacks. Signs of cyber attacks are visible in raw network data, containing raw IP packets with wrapped payloads in different headers and in log data. Due to their size and variety, these data are not suitable as inputs in machine learning methods. Therefore, in the first stage, a feature construction step is usually performed [10], which is crucial to detect those attacks as anomalous behavior.

Detection methods use either log data, network data or both. One of the first stages of an APT is the intrusion to the network. A survey of network-based intrusion detection datasets can be found in [15]. As stated in [10], literature addressing the whole APT life cycle usually only uses network data in order to detect those attacks, as in [9,11], with few exceptions where only log data are used—e.g., [16]. In [17] an APT attack detection method based on behavior analysis and deep learning with several steps is proposed. In their approach, network traffic is analyzed into IP-based network flows in the first step. In the second step, the IP information is reconstructed from the flow, and in the final step a combined deep learning model (bidirectional long short-term memory, graph convolutional networks) is used to extract features for the identification of IPs attacked by APTs.

There are also other recent papers focusing on another stage of an APT attack, the command and control stage [18]. The authors therein especially focused on APT attacks on mobile devices and so-called multiplatform APTs (attacking personal computers and mobile devices). For their approach, the authors used Domain Name System (DNS) records and extracted—depending on the device—several features of that traffic, namely, the total number of visits, the number of accessing hosts, domain length, solitariness of access, repeated requests, connection time, domain structure, access regularity and independent access. The authors considered two feature sets (one using all, the other with just some

selected), thereby showing that the highest f_1 score can be achieved when selecting the feature set depending on the platform.

As stated in [19], three kinds of feature-groups are mainly used in the literature: statistics-based, graph-based and time series-based features. Statistical features are the most common ones. They are usually calculated from the flow, and are defined as sets of packets having the same of IP source, IP destination, source port and destination port. Flows can be defined as unidirectional or bidirectional. Common statistical features are, e.g., the duration of the flow, the number of packets sent and their length. Graph-based features allow one to model or represent interactions of Internet networks into big connected graphs. Time series-based features can be seen as a sequence of events indexed in time order. For the detection, suitable characteristics of the network traffic lead to event-driven approaches and to patterns in the network. Although widely used in anomaly detection, those approaches are rarely used for network traffic classification.

This paper focuses on statistical features. Their advantage lies in—as stated in [19]—their computational simplicity. This is crucial in cases of dealing with high data throughput. Moreover, these features are suitable for encrypted network traffic. This is especially important for the detection of APTs.

While there are a wide variety of publications focusing on intrusion detection, e.g., [10,15], less publications consider later stages of an APT or the whole attack. In [3] an overview of approaches to detect APTs is given. Most of the approaches use supervised machine learning methods. Due to the structure of the data and the lack of (labeled) training data, such approaches are not feasible for being used in practice. In [20] a temporal correlation and traffic analysis approach for APT attack detection is used. The authors propose a filter method based on flow characteristics in combination with feature extraction and different anomaly detection methods—e.g., Support Vector Machine (SVN), k Nearest Neighbors (KNN), and gradient boosted decision trees. The approach presented in [21] is a detection scheme for multi-stage attacks based on multi-layer long and short-term memory networks. Although APTs are multi-stage attacks and therefore fall into that category, the dataset for testing the approach was an intrusion detection dataset, namely, the very widely used but already quite old and therefore outdated NSL-KDD dataset. Another approach to detect APTs in real-time was proposed in [22], based on a correlation of suspicious information flows. The tool of the authors generates a high-level graph which summarizes the attacker's steps in real-time. Besides [9], where the focus was on the data exfiltration step to identify a few hosts with suspicious activities, and three features based on statistics of the network flow (but host based) were proposed, none of those approaches investigated the influences of features on the detection capabilities in detail. Moreover, none of the approaches used outlier detection for APTs, although there are approaches using local outlier detection for network flow anomaly detection [23] and approaches using outlier detection on the over 20 year old NSL-KDD dataset for intrusion detection in [24].

3. Methodology

The methodology of the approach is illustrated in Figure 1. Network traffic data (pcaps) from two publicly available sources were used—one source containing benign data which serves as “background” data and another source containing network traffic from executed APT traces. More details are given in Section 3.1.

Figure 1. Methodology to investigate the influences of different features.

In the second step the CICflowmeter [25,26] was used to extract features from the network traffic. This tool has been used in several recent studies [27,28], especially for the extraction of many datasets widely used in the community. The CICflowmeter uses bidirectional flows by using so-called quadruples—connections between two IPs (with corresponding ports), where the first packet determines the forward (source to destination) and the second the backward (destination to source) flow. The CICflowmeter extracted around 80 features from TCP and UDP network traffic, including flags from TCP network traffic, and inter-arrival and idle based features. Based on a correlation analysis and boxplots (of single features), from that huge feature set, several suitable combinations were considered in detail, in order to investigate the influences of those features on the detection

approach. For the detection, an outlier detection method, namely, local outlier factor, was applied. For the evaluation, the focus was on a practical approach to ensure that at least a sign of each attack was detected, while the number of false positives was ensured to be small. Due to the type of attack, it is not feasible to catch all flows connected to the attack.

3.1. Dataset

Due to the lack of existing publicly available APT datasets [10], we followed the approach in [11], where two datasets—one containing APT data and another dataset containing benign data as background data—were combined. While [11] used the 20 year old DARPA dataset as “background” data, we used the more recent CICIDS2017 dataset [29]. There were several reasons for selecting this specific dataset as the background dataset. First, it is a publicly available dataset, well studied in the literature, and can be considered as a benchmark dataset in the intrusion detection research field. Usage of this dataset can contribute to the verifiability and comparability of results. Second, this dataset reflects a data type and network environment that is compatible with the Contagio dataset containing APT attacks, and reflects the research goals of work presented in this paper.

The CICIDS2017 dataset [29] includes a small enterprise network data of one-week duration, from Monday to Friday from 9:00 to around 17:00 each day. It is divided into five subsets according to the day of capture. The data were captured on a testbed architecture consisting of two separated networks, a victim-network with around 13 machines and an attacker network. Monday is the only subset without any attack. For the purpose of this experiment, network data from Monday were used.

The Contagio malware database [30] contains a collection of 36 files capturing raw network data. Each file recorded the traffic subject to attacks by different malware originating from APTs.

Both datasets contain raw network data, collected in .pcaps. In the first step, features of those .pcaps were extracted with the CICflowmeter. In the second, step several attacks (based on their duration) were selected from the Contagio malware database and combined with the features from Monday from the CICIDS2017 dataset. In order to combine those two datasets, victims in the network [29] were selected; see Table 1. The corresponding IP addresses of the Contagio files were then adapted in order to fit into the network.

Table 1. IPs of victims in the network.

Name	Attack Number	Victim IP
9002	1	192.168.10.5
Enfal Lurid	2	192.168.10.14
Nettravler	3	192.168.10.15
TrojanPage	4	192.168.10.15
Xinmic	5	192.168.10.15
Hupigon	6	192.168.10.14
Likseput	7	192.168.10.5
XtremeRAT	8	192.168.10.9

To avoid a dependence of the time and the attack on the detection, three different combinations were considered. While the background dataset stayed the same, the attacks were injected at different time slots; see Table 2. It was ensured that only one attack appeared within one hour. Four different machines were infected (Table 1) and the number of attacks on one machine was between one and three. The injection of the attacks into different time slots is given in detail in Table 2. The column *dur.* shows the duration of each attack. The time in the table refers to the start of the attack. The injection of the attacks is also visualized by the number of flows per hour for each of the later evaluation intervals (per hour); see Figure 2 for attack flows and Figure 3 for the benign data.

Table 2. Dataset, including the time of each attack.

Attack nr	Attack Name	dur.	Dataset A	Dataset B	Dataset C
1	9002	6.06	09:15	12:15	15:30
2	Enfal Lurid	2.87	12:15	15:30	09:15
3	Nettravler	33.17	15:15	09:15	12:15
4	TrojanPage	10.08	10:00	13:00	14:15
5	Xinmic	22.7	16:30	11:30	10:00
6	Hupigon	28.33	14:30	10:30	13:00
7	Likseput	4.62	11:15	14:45	16:30
8	XtremeRAT	1.27	13:45	16:30	11:00

Figure 2. Attack flows per hour on the different datasets, A, B and C. **Figure 3.** Benign flows per hour (all datasets).

The above described combination can easily be repeated for other combinations of benign and attack data. Since CICflowmeter used statistical features depending on the time a certain packet was sent (between two fixed IPs), and the extraction of features is quite

fast (on a common notebook the bigger (benign) pcaps are extracted within some hours), the injection of attacks on the file-level can be mainly performed with programmable routines and the adjustments of IPs. The only process which needs human expertise is the identification of the victim and attacker IPs, including the detection of potential other important members in the networks whose IPs have to be changed (e.g., DNS servers), and of course the decision of where to place a certain attack.

3.2. Features

In order to achieve high performance and limited computational storage as stated in [9], the proposed approach focuses on network data. Since computational power is quite limited, but there is—to the best of our knowledge—also a lack of the investigation of features for the detection of cyber attacks, the influences of those are considered. Several feature sets are considered in detail to evaluate whether there are any superior features or feature sets which outperform others or significantly help to detect APTs. Since these features are only based on statistics of the network traffic, all these feature sets are suitable for encrypted traffic use.

In the literature, various kinds of flows are used for the creation of statistical features (see [31] for a comparison of flow exporters) using different numbers of features. While some flow extractors, e.g., Maji, Softflowd and Transalyzer, use the whole unidirectional flow for the creation, there are also flow extractors such as CICflowmeter or Netmate which additionally provide features from bidirectional flow. In this paper we consider different features for the bidirectional flow provided with CICflowmeter.

With the CICflowmeter, in total 76 different features have been extracted, ranging from counters for different flags of TCP network traffic, to packet length, the average packet size, the active and idle time of a flow and the inter-arrival time. Moreover, some features are especially useful for identification, such as the flow ID, the source IP, the destination IP and the source port and the destination port. In order to avoid bias and to ensure that an attack is detected by its network traffic behavior and not by the IP (which could change easily), the features for identification are excluded in this study. A first investigation of these features showed that five features only contained zero values. Therefore, they have been removed (this applies for the features *Bwd PSH Flags*, *Bwd URG Flags*, *Fwd Bulk Rate Avg*, *Fwd Bytes Bulk Avg* and *Fwd Packet Bulk Avg*). Moreover, two pairs of features turned out to be identical. This addresses the pair *Bwd Segment Size Avg* and *Bwd Packet Length Mean*, and the pair *Fwd Segment Size Avg* and *Fwd Packet Length Mean*. Therefore, only the second pair of those features was kept.

From the other features, descriptive statistics have been calculated and a correlation analysis (see Figure 4) has been performed. High correlations between some features, were taken into account for the feature selection. This applied especially to a higher correlation between the IAT features of the total flow and between the forward IAT the backward IAT flow, and the total, forward and backward packet length features. The influence of a combination of them is addressed by the features h_1 , h_2 , h_3 and h_4 in the experiments.

Figure 4. Pearson correlations of 66 features.

We dismissed features focusing on the minimum, since that value is per definition fixed for any statistical flow. Based on the boxplots (see Figures 5–8), we further dismissed the flag features, particularly because a detailed investigation also showed that most of the features (*CWR Flag Count*, *ECE Flag Count* and *URG Flags*) had very few non-zero values (all benign data), but all were zero for the attack data.

We selected feature sets for further investigation based on the detailed study of boxplots and the correlation analysis and by using knowledge of previous publications. In [11], for example, only two features were used for the detection of advanced persistent threats, namely, the duration of a flow and the total number of packets transferred (corresponding to feature set f_1).

Other features are not applicable for being used on bidirectional data flows, e.g., from log-based approaches or host-based features, as in [9]—which addressed the data exfiltration stage—using the features:

- numbytes: The number of megabytes uploaded by an internal host to an external address;
- numflows: The number of flows to an external host initiated by an internal host;
- numdst: The number of external IP addresses related to a connection initiated by an internal host.

Figure 5. Boxplots for different features—part 1.

Figure 6. Boxplots for different features—part 2.

Figure 7. Boxplots for different features—part 3.

Figure 8. Boxplots for different features—part 4.

Moreover, those features are not included in the CICflowmeter tool.

As stated in [19], the inter-arrival-time and the active and idle time of a flow seemed to be superior in previous work. Therefore, those features were included in different variants. As the number of features used is highly correlated with the processing time and potential storage time, the goal was to find a small superior feature set and avoid including just any (similar) features. That is the reason why we either used the median and standard deviation of a certain value together or the maximum of a value. We did not use the minimum (e.g., packet length), since the boxplots did not show any useful capabilities to distinguish benign and attack flows.

Other publications considering the whole APT life cycle (compare [10]) and not mainly focusing on intrusions detection, either used alerts for the detection [32], followed a graph-based approach [22] or lacked details of features used [33].

Based on that, different feature sets were considered: see Table 3 for features using (only) the whole bidirectional flow and Table 4 for features including some for the forward or backward flow only. The duration of each flow was limited in CICflowmeter by the activity timeout of 5 million seconds and the flow timeout of 12 million seconds.

Table 3. Bidirectional (whole) flow-based features.

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
duration	✓	✓	✓	✓	✓	✓		
pkt len max	✓			✓	✓	✓		✓
pkt len mean							✓	
pkt len std							✓	
IAT max				✓	✓	✓		✓
IAT mean			✓				✓	
IAT std			✓				✓	
active max		✓		✓	✓	✓		✓
active mean							✓	
active std							✓	
down up ratio						✓		
idle max					✓	✓		✓
idle mean							✓	
idle std							✓	

Table 4. Bidirectional flow-based features including some for forward and backward-flow only.

	h_1	h_2	h_3	h_4	h_5	h_6	h_7
pkt len max		✓	✓				
pkt len mean	✓				✓	✓	✓
pkt len std	✓				✓	✓	✓
bwd pkt len max			✓	✓	✓	✓	✓
fwd pkt len max			✓	✓	✓	✓	✓
IAT max		✓	✓				
IAT mean	✓				✓	✓	✓
IAT std	✓				✓	✓	✓
fwd IAT max			✓	✓	✓	✓	✓
bwd IAT max			✓	✓	✓	✓	✓
active max		✓	✓	✓			
active mean	✓				✓	✓	✓
active std	✓				✓	✓	✓
idle max		✓	✓	✓			
idle mean	✓				✓	✓	✓
idle std	✓				✓	✓	✓
average pkt size						✓	
fwd init win byte	✓	✓	✓	✓	✓	✓	
bwd init win byte	✓	✓	✓	✓	✓	✓	

3.3. Outlier Detection

This paper proposes an unsupervised method to catch signs of the attacks, namely, the local outlier factor [34]. For that method—as for outlier detection in general—the goal is to separate regular observations from some outliers. The algorithm computes a so-called local outlier factor (LOF)—a score—to reflect the degree of the abnormality of an observation for each object in the dataset. The approach is local in the sense that it is calculated only on a restricted neighborhood of each object, and the calculation of the LOF is only based on those neighbors. The approach is loosely related to density-based clustering, like, e.g., DBSCAN [35] and OPTICS [36].

According to [34], the local outlier factor of an object p is defined as

$$LOF_{MinPts(p)} = \frac{\sum_{o \in N_{MinPts(p)}} \frac{lrd_{MinPts(o)}}{lrd_{MinPts(p)}}}{|N_{MinPts(p)}|}$$

with the number of nearest neighbors used in the defined local neighborhood of p , namely, $MinPts$. The local reachability density lrd of an object p is defined as

$$lrd_{MinPts(p)} = \left(\frac{\sum_{o \in N_{MinPts(p)}} \text{reach-dist}_{MinPts}(p, o)}{|N_{MinPts(p)}|} \right)^{-1}$$

where $MinPts(p)$ denotes the nearest neighbors of p and the reachability distance of an object p with respect to object o is given as

$$\text{reach-dist}_k(p, o) = \max\{k - \text{distance}(o), d(p, o)\}$$

The k -distance is the distance of a point to its k th neighbor, i.e., the distance to its k th closest point.

In the proposed approach the outlier detection is applied on different time slots (as shown in Figure 9). While this paper focuses on how to select a suitable feature set, the approach presented here can also be used for anomaly detection (with a selected feature set). In such a case, suitable features need to be extracted from network traffic. Figure 9 shows network traffic at a central point. Depending on the environment, the extraction from features from different endpoints would be possible as well. In this case, the anomaly

detection should be applied to each endpoint separately, in order to enable a potential user-specific behavior.

Figure 9. Detection approach with the local outlier factor method.

For different feature sets, anomaly detection with the local outlier factor method is used in different time slots, where each time slot contains exactly one APT attack. Furthermore, as shown by this analysis, this approach is applicable for security at runtime, to give security administrators hints for further investigations.

Moreover, due to the training in different time slots, characteristics such as more e-mail activity in the morning, are taken into account. It has to be noted that in addition to APT detection, the proposed algorithm is expected to detect other anomalies as well, such as software updates, uploading huge files for partners in a project and other tasks usually appearing in a large network.

As a pre-processing step and in order to avoid detection rates only based on the scaling of some features, robust scaling is performed. Therefore, as for the local outlier detection too, Python's *sklearn* library is used. The built in robust scaling is robust to outliers, removes the median and scales the data according to the quartiles range. Each feature is centered and scaled independently.

The experiments have been performed on a Windows notebook with a 2.6 GHz CPU. For the choice of the parameters in a pre-study, several parameters, especially the number of neighbors and the distance measured, were evaluated. For the number of neighbors, the values $\{10, 15, \dots, 55, 60\}$, and for the metric minkowski, manhattan and cityblock, were used. Based on these experiments, the best choice is to set the number of neighbors to 40 and to use the *minkowski* norm.

The evaluation of the results is always per hour, i.e., in the intervals 9–10 o'clock, 10–11 o'clock, ..., 15–16 o'clock and 16–17 o'clock. In each of these time slots, exactly one attack exists. All attacks consist of a different number of corresponding flows (see Figure 2). Local outlier scores were around 1 (in fact in the results -1 , since the negative outlier score is used to recommendations of the used library) are clearly inliers. However, there is no rule for setting an adequate threshold for identifying significant outliers. A proper threshold highly depends on the dataset.

4. Results and Discussion

The goal of the approach is to find the set of features, which supports the detection of anomalies related to APTs best. Therefore, benign and malicious flows are distinguished. A malicious flow is a feature vector coming from an attacker IP address. Flows not related to attackers' IP addresses are considered as benign. The goal is to detect malicious flows. It has to be emphasized that it is not feasible to detect all (and only) flows related to an attack, as stated in [9]. The overall goal, therefore, is to detect at least one sign of an attack, and mark it as suspicious activity, which has to be investigated by security administrators later on. Moreover, the number of false positives—which gives a measure for the effort a security administrator is faced with—should be as low as possible.

One of the critical parameters for the evaluation is the threshold, since the number of outliers highly depends on it. As pointed out, there is no rule to define a threshold for identifying significant outliers. Therefore, in an initial step, experiments with different thresholds were performed. For the thresholds, a constant one, set to -1.2 for all time slots considered, and quantile-based ones, i.e., using the quantiles 0.15 and 0.1, were used.

For a better comparison of the results obtained with those thresholds, weighted true negative rate (TNR) values (see Table 5) for selected feature sets were calculated, i.e., the TNR was calculated for each time slot separately and then summed up and divided by the number of time slots. This ensures that each attack is weighted equally and not too much focus is put on attacks with more flows. An only flow-based evaluation could bias the results in a sense that attacks with more flows have a much higher weight and missing attacks with only a few flows could lead to very promising results, although some of those attacks are completely missed. In order to keep the number of true negatives as high as possible and because the weighted recall was similar for the different thresholds, we set the threshold to quantile 0.1 for further experiments.

Table 5 shows the highest TNR for the quantile 0.1. For the two quantile-based thresholds we did not see any influence on the TNR. For the constant threshold set to -1.2 , a small difference depending on the feature set could be observed.

Next, results on the different time slots with different feature sets are given.

Table 5. Weighted true negative rates (TNRs) for different thresholds.

Threshold	f_5	f_6	f_7	f_8
const.	83.0%	82.3%	80.98%	83.9%
quantile (0.15)	85.01%	85.01%	85.02%	85.01%
quantile (0.1)	90.02%	90.02%	90.02%	90.02%

4.1. Comparison of Different Feature-Sets

Results of the experiments with different feature sets from Tables 3 and 4, evaluated on dataset A, are visualized in Figure 10a,b. While the TNR is for all the feature sets around 90% (the false positive rate is correlated to the TNR and therefore not displayed separately), there are remarkable differences in the number of detected malicious flows. While for some of the attacks, such as for *TrojanPage*, some feature sets only catch a quite small number of malicious flows (f_1 to f_4), the results increase for different feature sets (f_5 to f_6). For other attacks, such as the attacks *Xinmic* and *Hupigon*, the situation is different, and only a few flows are caught. While most of the feature sets at least catch some signs of the attacks, f_3 misses it completely. Comparing the results presented in Figure 10a,b shows that feature sets from Table 4 in general cover far more flows of the attack. However, even for those feature sets it is hard to deal with attacks such as *Xinmic* and *Hupigon*. While the recall can be increased a little, most of the attack flows are still under the radar.

Figure 10. Recall for different feature sets.

A detailed look into the different features shows the influences of individual features on the capability of detecting a certain attack (compare also Table 6). While the duration—although widely used—is not crucial, the idle time really pays off for the detection of most of the attacks. The inter-arrival time (IAT), which is mentioned explicitly in the literature for the detection of certain cyber attacks, had some contribution (especially concerning f_7). However, it does not pay off in combination with the duration alone (compare f_3 and the missed attack in the time slots 14 and 15). The idle time, nevertheless, contributes strongly to the detection of an attack. The decision on whether to use a mean/std combination of values or only the maximum depends on the cyber attack (f_7 and f_8). Using information on forward and backward flow separately improves particularly the detection of attack XtremeRAT. A closer look on the features h_5 to h_7 shows that the features average packet size, fwd init win byte and bwd init win bytes do not improve the attack detection capabilities.

Table 6. Weighted recall on dataset A for feature sets f_1 – f_8 .

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
recall	16.72%	35.96%	23.59%	33.92%	75.80%	75.80%	79.10%	76.71%

Overall it can be said that the feature sets f_7 , f_8 and h_5 , h_6 and h_7 performed best. Since the latter three lead to the exact same recall, for further experiments only the feature set h_7 was used. The reason for this is that this feature set has less features than the other two; see Table 4.

Besides the influence on the detection capabilities, especially missed attacks, the invested resources such as the time for the calculation of the local outlier factor score and the resources for storing the features for a later analysis, also had to be taken into account. Since all the features have been extracted with the CICflowmeter tool, we did not investigate the effect of calculating only a limited set of features.

One expectation might be that a higher number of features usually increases the detection capabilities. We investigated that by comparing several feature sets (see Tables 7 and 8). The reason for the choice of these two is a kind of basic feature set which is contained in each table, enriched by additional features. This is important, since the different features have quite different influences (compare the boxplots) on the detection.

Table 7. Time and computational resources for selected bidirectional (whole) flow.

	f_4	f_5	f_6	f_8
number of features	4	5	6	4
storage of flow records	26 MB	35.6 MB	37.6 MB	32.7 MB
time lof score computation	34.57 s	38.18 s	45.17 s	35.34 s
weighted recall	33.92%	75.8%	75.8%	76.71%
weighted TNR	89.9994%	90.0168%	90.0168%	90.0185%

A detailed look at Tables 7 and 8 shows that the storage of the flows and the computational time for the local outlier factor scores increases with the number of features. Concerning the recall, we can see that the number of features is not crucial for the detection capabilities, as the best results can be obtained with feature set f_8 , which only has four features. In order to estimate the work of a security administrator, we calculated a so-called filter factor:

$$\text{filter factor} = \frac{tp + fp}{tp + tn + fp + fn'}$$

the percentage of flows a security administrator has to check. Independently of the feature set, the amount of flows for a further check was reduced to 10%. The TNR was mainly the same; there was only a small change. For the feature sets in Table 8 the situation was similar. There was a small change in the TNR for increasing the number of features up to 12; however, there was no change in the recall, nor the TNR, nor the filter factor for the feature sets with 14 or 15 features. Concerning resources as storage and calculation time, the best feature set was therefore h_7 .

Table 8. Time and computational resources for selected bidirectional flow.

	h_1	h_5	h_6	h_7
number of features	10	14	15	12
storage of flow records	60.2 MB	71.4 MB	75.5 MB	68.1 MB
time lof score computation	116.68 s	185.09 s	196.39 s	138.65 s
weighted recall	78.0%	79.80%	79.80%	79.80%
weighted TNR	90.0164%	90.0195%	90.0195%	90.0195%

Based on the values obtained, it can be estimated that with feature set f_8 , approximately 35 GB were needed, and with feature set h_7 , approximately 74 GB of storage in a network of the given size is necessary. Storing the whole features extracted with the CICflowmeter tool can be estimated to be around 314 GB.

4.2. Evaluation on Different Datasets

In the next steps the best three feature sets are evaluated on the three different datasets in order to evaluate the influence of the background. The corresponding results are shown in Table 9 per time slot and in Table 10 for some selected attacks. Since the TNR stays the same for the different feature sets, it is only given in Table 11 in a weighted form for completeness. The weighted recall and TNR on the different datasets show that f_7 , which focuses on using the mean and the std of statistical measures instead of the max only (used by f_8), is superior. While f_7 and f_8 lead to similar results on dataset A, the results on dataset C and dataset B differ significantly. The detailed evaluation in Table 9 shows a satisfying recall for most of the attacks. However, for some attacks almost all malicious flows cannot be recognized as outliers. These attacks are displayed separately in Table 10. While most flows of the attacks *Xinmic* and *Hupigon* for any dataset and any background (compare Table 2) cannot be recognized as outliers, a dependence on the feature set is visible. For *Xinmic*, less flows were detected with the features f_8 for dataset B, but for dataset A, the feature sets f_8 and h_7 performed better than feature set f_7 . For dataset C the results with f_7 were also worse than for the two others. Another remarkable point is that for the attack *XtremeRAT* and the feature set f_8 , not all flows were detected on all the datasets (with different recall). The situation was similar for the attack *9002*, but in this case the recall on dataset B and C was less (on dataset A all flows were detected as outliers). For *Hupigon*, feature set f_7 was able to capture the behavior best on dataset A. For dataset B, the feature set h_7 led to the best recall; for dataset C the feature sets f_7 and f_8 led to the same detection of malicious flows.

Table 9. Recall for feature sets f_7 , f_8 and h_7 on different datasets.

Dataset\Time		9–10	10–11	11–12	12–13	13–14	14–15	15–16	16–17
f_7	A	100%	100%	100%	100%	100%	17.5%	100%	15.32%
f_8	A	100%	100%	100%	100%	76.92%	15.00%	100%	21.77%
h_7	A	100%	100%	100%	100%	100%	15.00%	100%	23.39%
f_7	B	100%	16.26%	21.77%	100%	100%	100%	100%	100%
f_8	B	100%	15.00%	15.32%	33.33%	100%	100%	100%	76.92%
h_7	B	100%	17.5%	23.39%	100%	100%	100%	100%	100%
h_7	C	100%	14.51%	100%	100%	17.5%	100%	100%	100%
f_8	C	100%	19.35%	84.62%	100%	17.5%	100%	66.67%	100%
h_7	C	100%	19.35%	100%	100%	15.00%	100%	100%	100%

Table 10. Recall for feature sets f_7 , f_8 and h_7 on different datasets for selected attacks.

Dataset\Attack		9002	Xinmic	Hupigon	XtremeRAT
f_7	A	100%	15.32%	17.5%	100%
f_8	A	100%	21.77%	15.00%	76.92%
h_7	A	100%	23.39%	15.00%	100%
f_7	B	100%	21.77%	16.26%	100%
f_8	B	33.33%	15.32%	15.00%	76.92%
h_7	B	100%	23.39%	17.5%	100%
f_7	C	100%	14.51%	17.5%	100%
f_8	C	66.67%	19.35%	17.5%	84.62%
h_7	C	100%	19.35%	15.00%	100%

Table 11. Weighted recall and TNR for feature sets f_7 , f_8 and h_7 on different datasets.

	Dataset	Weighted Recall	Weighted TNR
f_7	A	79.10%	90.02%
f_8	A	76.71%	90.02%
h_7	A	79.80%	90.02%
f_7	B	79.75%	90.02%
f_8	B	67.57%	90.01%
h_7	B	80.11%	90.02%
f_7	C	79.00%	90.02%
f_8	C	73.52%	90.02%
h_7	C	79.29%	90.02%

Details to the local outlier scores are given in Figure 11, where for each of the datasets the absolute values of the local outlier scores (benign and malicious) for the attack *Hupigon* are visualized on a log-scale. It can be observed that the benign data look different for each dataset. For dataset B, there are less outliers in the benign flows. For dataset A there are significant outliers from the benign and the malicious flows, with similar values. In the case of dataset C, the situation is different, since benign flows have higher outlier score values than malicious flows. That shows the huge influence of the background data and the ongoing activity there, for detecting signs of attacks, since in all of those datasets the same attack data with the same feature set were used.

Figure 11. Local outlier score values of f_7 for the attack *Hupigon*.

The influences on the outlier scores from the different feature sets can be observed in Figure 12. While for the feature set f_7 a quite clear threshold could be set, the spreading of the outlier scores is wider for the feature set h_7 . Nevertheless, for feature set f_8 , some of the malicious scores are—comparing the scores only—very close to the scores of benign flows. Furthermore, it has to be stated that in those evaluation intervals, outliers from benign flows have smaller scores than almost all malicious flows.

Figure 12. Local outlier score values for different features on dataset *A* for the attack *TrojanPage*.

4.3. Comparison with Related Work

In [11], a supervised approach, namely, k -Nearest Neighbors and a correlation fractal dimension-based algorithm were used to detect APTs with only two features, namely, the duration of a flow and the total number of packets transferred (corresponding to our feature set f_1). The authors report a recall over 92%. However, there the 20 year old NSL-KDD dataset in combination with Contagio were used. Based on our experiments with the feature set f_1 , we do not expect to get any similar detection rate for that features set (compare Figure 10a, especially the scores for f_1).

In [9], where a specifically tailored host-based feature set was used for a detection of the data exfiltration stage, the authors reported that they could analyze about 140 million flows related to approximately 10,000 internal hosts in about 2 minutes. Due to the type of the features, those results are not directly comparable to our case.

In [19] it is stated that inter-arrival-time and the active and idle time of a flow seem to be superior in some previous work. We therefore included those features in different variants (all except f_1). The results (see Table 6) indicate that feature sets with either using inter-arrival-time or active and idle time are remarkably better than without. According to the weighted recall, using the active time also has a visible effect on the detection rate (compare f_1 and f_3 , which do not contain the active time, with the other feature sets). Moreover, we see a benefit in using the idle time too (feature sets f_5 to f_8). Results containing features only based on the forward or backward flow indicate in Figure 10b that there is no effect on the detection rate of using the features *average pkt size* and *init win byte*.

We want to emphasize that the scope of this work was the investigation of features for the APT detection and not on APT detection methods. Therefore, and due to the lack of benchmark APT datasets in literature, the possibilities for a direct comparison of the proposed approach and the results with further related literature are limited.

5. Conclusions

This paper focused on the effect of features to detect advanced persistent threats with the local outlier factor method. Since APT attacks try to hide in network traffic, it is crucial

to focus on the design (and recording) of suitable features to detect these attacks. Due to huge amounts of network (and log) traffic, and the need to record some of the traffic for more detailed investigations in the future, it is important to know on what to focus. Therefore, several state-of-the-art network traffic features have been investigated, using correlation analysis and boxplots for a first insight into those features. Then, different suitable feature sets and their influences on the detection in combination with the local outlier factor method were investigated in detail. The results show a remarkable impact of the choice of the feature sets on detecting signs of APTs. Moreover, an influence of the background data on the detection capabilities has been shown. Two feature sets (f_7 and h_7) stand out as the best selection and are capable of detecting most of the malicious flows for the majority of the attacks. Nevertheless, the most challenging attacks—*Xinmic* and *Hupigon*, still do not achieve a satisfying detection rate, which presents open points for future work.

Our results and this in-depth investigation contribute to a better understanding of APTs and the capabilities of local outlier methods to detect signs of those attacks. Moreover, our experiment showed, as expected, a clear connection and a trend of more resources needed for processing larger feature sets. On the other hand, the experiments indicate that detection performance, measured through weighted recall and TNR, do not have necessarily the same trend. The main conclusion, therefore, is that investigating features in more detail and to build optimized and customized detection solutions can lead to saving processing resources and increasing detection performance at the same time. In order to improve the application of local outlier detection, further work should also investigate automatic hyperparameter tuning methods and studies to select a suitable threshold. Moreover, the proposed local outlier factor approach could be used as the first step in a multi-stage approach for the APT detection. The obtained anomalous flows can then be investigated in more detail in a second step by using other machine learning algorithms. Potential candidates, therefore, are either classical methods or neural networks in order to separate anomalous malicious flows from anomalous benign flows (related to anomalous behavior in the network traffic, e.g., software updates). In such a multi-stage approach, several scenarios are imaginable—either the use of additional identifier-based features such as source IP, destination IP, source port and destination port in combination with the local outlier score; and using the local outlier score in combination with all previously used features or an additional selection of those.

From the application point of view, the presented local outlier factor approach could be used for central network traffic (only), but also on different end points, or in a scenario using central and edge-based network traffic. In case the local outlier factor is used on end points, the method is expected to capture a rather end point specific behavior. The anomaly scores obtained in such a way could then be processed at a central point together with features for identification such as source IP, destination IP, port IP and source IP.

Future work will therefore consider a correlation of the outlier flows with different machine learning methods to increase the true negative rate and to decrease the false positive rate in the second stage. Additional work should also consider concepts to include log data.

Author Contributions: Conceptualization, K.H.-S., U.K. and B.S.; methodology, K.H.-S. and U.K.; validation, K.H.-S.; formal analysis, K.H.-S. and U.K.; investigation, B.S.; project administration, K.H.-S.; writing—original draft preparation, K.H.-S. and U.K.; writing—review and editing, K.H., U.K. and B.S.; visualization, K.H.-S. and U.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Austrian Federal Ministry of Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Fortinet. *Cybersecurity Statistics*; Fortinet: Sunnyvale, CA, USA, 2020.
- TheHill. *FBI Sees Spike in Cyber Crime Reports during Coronavirus Pandemic*; TheHill: Washington, DC, USA, 2020.
- Alshamrani, A.; Myneni, S.; Chowdhary, A.; Huang, D. A Survey on Advanced Persistent Threats: Techniques, Solutions, Challenges, and Research Opportunities. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 1851–1877. [\[CrossRef\]](#)
- Chen, P.; Desmet, L.; Huygens, C. A study on advanced persistent threats. In *IFIP International Conference on Communications and Multimedia Security*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 63–72.
- GReAT. *APT Trends Report Q1 2020*; Technical Report; Kaspersky: Moscow, Russia, 2020.
- M-Trends 2020, FireEye Mandiant Services Special Report*; Technical Report; FireEye Mandiant: Alexandria, VA, USA, 2020.
- Radanliev, P.; De Roure, D.C.; Nurse, J.R.; Montalvo, R.M.; Cannady, S.; Santos, O.; Burnap, P.; Maple, C. Future developments in standardisation of cyber risk in the Internet of Things (IoT). *SN Appl. Sci.* **2020**, *2*, 1–16. [\[CrossRef\]](#)
- Johnson, A.L. Cybersecurity for financial institutions: The integral role of information sharing in cyber attack mitigation. *NC Bank. Inst.* **2016**, *20*, 277.
- Marchetti, M.; Pierazzi, F.; Colajanni, M.; Guido, A. Analysis of high volumes of network traffic for advanced persistent threat detection. *Comput. Netw.* **2016**, *109*, 127–141. [\[CrossRef\]](#)
- Stojanović, B.; Hofer-Schmitz, K.; Kleb, U. APT Datasets and Attack Modeling for Automated Detection Methods: A Review. *Comput. Secur.* **2020**, *92*, 101734. [\[CrossRef\]](#)
- Siddiqui, S.; Khan, M.S.; Ferens, K.; Kinsner, W. Detecting advanced persistent threats using fractal dimension based machine learning classification. In *Proceedings of the 2016 ACM on International Workshop on Security and Privacy Analytics*; ACM: New York, NY, USA, 2016; pp. 64–69.
- Radanliev, P.; De Roure, D.; Walton, R.; Van Kleek, M.; Montalvo, R.M.; Santos, O.; Burnap, P.; Anthi, E. Artificial intelligence and machine learning in dynamic cyber risk analytics at the edge. *SN Appl. Sci.* **2020**, *2*, 1–8. [\[CrossRef\]](#)
- Radanliev, P.; De Roure, D.; Van Kleek, M.; Ani, U.; Burnap, P.; Anthi, E.; Nurse, J.R.; Santos, O.; Montalvo, R.M. Dynamic real-time risk analytics of uncontrollable states in complex internet of things systems: Cyber risk at the edge. *Environ. Syst. Decis.* **2020**, 1–12.
- Ussath, M.; Jaeger, D.; Cheng, F.; Meinel, C. Advanced persistent threats: Behind the scenes. In *Proceedings of the 2016 Annual Conference on Information Science and Systems (CISS)*, Princeton, NJ, USA, 16–18 March 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 181–186.
- Ring, M.; Wunderlich, S.; Scheuring, D.; Landes, D.; Hotho, A. A Survey of Network-based Intrusion Detection Data Sets. *Comput. Secur.* **2019**, *86*, 147–167. [\[CrossRef\]](#)
- Friedberg, I.; Skopik, F.; Settanni, G.; Fiedler, R. Combating advanced persistent threats: From network event correlation to incident detection. *Comput. Secur.* **2015**, *48*, 35–57. [\[CrossRef\]](#)
- Do Xuan, C.; Nguyen, H.D.; Dao, M.H. APT attack detection based on flow network analysis techniques using deep learning. *J. Intell. Fuzzy Syst.* **2020**, *39*, 4785–4801. [\[CrossRef\]](#)
- Xiang, Z.; Guo, D.; Li, Q. Detecting mobile advanced persistent threats based on large-scale DNS logs. *Comput. Secur.* **2020**, *96*, 101933. [\[CrossRef\]](#)
- Pacheco, F.; Exposito, E.; Gineste, M.; Baudoin, C.; Aguilar, J. Towards the Deployment of Machine Learning Solutions in Network Traffic Classification: A Systematic Survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 1988–2014. [\[CrossRef\]](#)
- Lu, J.; Chen, K.; Zhuo, Z.; Zhang, X. A temporal correlation and traffic analysis approach for APT attacks detection. *Clust. Comput.* **2017**, *22*, 7347–7358. [\[CrossRef\]](#)
- Charan, P.S.; Kumar, T.G.; Anand, P.M. Advance Persistent Threat Detection Using Long Short Term Memory (LSTM) Neural Networks. In *Proceedings of the International Conference on Emerging Technologies in Computer Engineering*, Jaipur, India, 1–2 February 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 45–54.
- Milajerdi, S.M.; Gjomemo, R.; Eshete, B.; Sekar, R.; Venkatakrishnan, V. Holmes: Real-time apt detection through correlation of suspicious information flows. In *Proceedings of the 2019 IEEE Symposium on Security and Privacy (SP)*, San Francisco, CA, USA, 19–23 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1137–1152.
- Paulauskas, N.; Bagdonas, A.F. Local outlier factor use for the network flow anomaly detection. *Secur. Commun. Netw.* **2015**, *8*, 4203–4212. [\[CrossRef\]](#)
- Auskalnis, J.; Paulauskas, N.; Baskys, A. Application of local outlier factor algorithm to detect anomalies in computer network. *Elektron. Elektrotechnika* **2018**, *24*, 96–99. [\[CrossRef\]](#)
- Lashkari, A.H.; Draper-Gil, G.; Mamun, M.S.I.; Ghorbani, A.A. Characterization of tor traffic using time based features. In *Proceedings of the ICISp, Porto, Portugal*, 9–21 February 2017; pp. 253–262.
- Draper-Gil, G.; Lashkari, A.H.; Mamun, M.S.I.; Ghorbani, A.A. Characterization of encrypted and vpn traffic using time-related. In *Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP)*, Rome, Italy, 19–21 February 2016; pp. 407–414.
- Pawlicki, M.; Choraś, M.; Kozik, R.; Hołubowicz, W. On the Impact of Network Data Balancing in Cybersecurity Applications. In *International Conference on Computational Science*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 196–210.
- Neuschmied, H.; Winter, M.; Hofer-Schmitz, K.; Stojanovic, B.; Kleb, U. Two Stage Anomaly Detection for Network Intrusion Detection. In *Proceedings of the ICISp 2021, Vienna, Austria*, 11–13 February 2021.

29. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In Proceedings of the ICISSP, Funchal, Portugal, 22–24 January 2018; pp. 108–116.
30. Parkour, M. Contagio Malware Database. Available online: <http://contagiodump.blogspot.com/> (accessed on 2 July 2019).
31. Haddadi, F.; Zincir-Heywood, A.N. Benchmarking the effect of flow exporters and protocol filters on botnet traffic classification. *IEEE Syst. J.* **2014**, *10*, 1390–1401. [[CrossRef](#)]
32. Ghafir, I.; Hammoudeh, M.; Prenosil, V.; Han, L.; Hegarty, R.; Rabie, K.; Aparicio-Navarro, F.J. Detection of advanced persistent threat using machine-learning correlation analysis. *Future Gener. Comput. Syst.* **2018**, *89*, 349–359. [[CrossRef](#)]
33. Vance, A. Flow based analysis of Advanced Persistent Threats detecting targeted attacks in cloud computing. In Proceedings of the 2014 First International Scientific-Practical Conference Problems of Infocommunications Science and Technology, Kharkov, Ukraine, 14–17 October 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 173–176.
34. Breunig, M.M.; Kriegel, H.P.; Ng, R.T.; Sander, J. LOF: Identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, TX, USA, 16–18 May 2000; pp. 93–104.
35. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Kdd, Portland, OR, USA, 2–4 August 1996; Volume 96, pp. 226–231.
36. Ankerst, M.; Breunig, M.M.; Kriegel, H.P.; Sander, J. OPTICS: Ordering points to identify the clustering structure. *ACM Sigmod Rec.* **1999**, *28*, 49–60. [[CrossRef](#)]