From Intrusion Detection to Attacker Attribution: A Comprehensive Survey of Unsupervised Methods

Antonia Nisioti, *Member, IEEE*, Alexios Mylonas, *Member, IEEE*, Paul D.Yoo, *Senior Member, IEEE*, Vasilios Katos, *Member, IEEE*

***Abstract*— Over the last five years there has been an increase in the frequency and diversity of network attacks. This holds true, as more and more organisations admit compromises on a daily basis. Many misuse and anomaly based Intrusion Detection Systems (IDSs) that rely on either signatures, supervised or statistical methods have been proposed in the literature, but their trustworthiness is debatable. Moreover, as this work uncovers, the current IDSs are based on obsolete attack classes that do not reflect the current attack trends. For these reasons, this paper provides a comprehensive overview of unsupervised and hybrid methods for intrusion detection, discussing their potential in the domain. We also present and highlight the importance of feature engineering techniques that have been proposed for intrusion detection. Furthermore, we discuss that current IDSs should evolve from simple detection to correlation and attribution. We descant how IDS data could be used to reconstruct and correlate attacks to identify attackers, with the use of advanced data analytics techniques. Finally, we argue how the present IDS attack classes can be extended to match the modern attacks and propose three new classes regarding the outgoing network communication.**

***Index Terms*— Anomaly IDS, correlation and attribution, attack reconstruction, digital forensics, network forensics, data analytics, unsupervised learning, feature selection**

I. Introduction

A significant rise of cyber attacks has been witnessed in the recent years. Some examples include the Sony data breach in 2014, the Ukraine attack on power grid in the end of 2015 and even the hack of the controversial cybersecurity group Hacking Team in 2015. According to [1], in the first quarter of 2017, DDoS attacks increased by 30% compared to the previous year. Twelve of these attacks exceeded a bandwidth of 100 Gigabits per second (Gbps), while two exceeded 300 Gbps against the media and entertainment sectors. Moreover, 43% of web traffic across their network was produced by bots and 63% was malicious. These attacks not only impair the normal operation of organisations and governments, but also have social consequences and may affect and impair critical infrastructures.

As stated in [2], targeted attacks are now an established part of the threat landscape and Advanced Persistent Threats (APTs) are one of the biggest security challenges as they target companies, infrastructures and governments. Indicatively, in 2015 Carbanak APT campaign successfully attacked up to 100 financial institutions across the globe with total losses as high as US$1 billion [2]. An APT is a sophisticated and premeditated method for an attacker to achieve specific predetermined goals. The word “persistent” in the APT acronym holds a double meaning: a) its ability to stay in the system/network until it fulfils its purpose and b) the persistence of the attacker who will not stop until her/his goal is reached. Attackers nowadays are highly motivated and most of the times have surplus time and money to devote for achieving their goal. Cyber-crime has become more organized and sophisticated, as criminals can easily purchase all the necessary means to carry an attack from the underground market [3]. Malware infection frameworks such as Zeus or SpyEye, can be purchased for US$4,000–US$7,000, while hosting Browser Exploit Packs (BEP) on a website to lure the victims, costs for US$1,500–US$3,000 [4]. According to Symantec [5], a drive-by download web toolkit, which includes updates and 24x7 support, can be rented for between US$100 and US$700 per week, while distributed denial-of-service (DDoS) attacks can be ordered from US$10 to US$1,000 per day.

Over the last two decades, many IDSs have been proposed, developed, reviewed and evaluated. An IDS processes the traffic of a network and potentially data from its host to detect any malicious activity, such as unauthorized access or a DDoS attack. In the early days of internetworking, intrusion detection was performed manually by analysts and system administrators, who used to review all the monitored activities in the network. As networks increased in size and complexity, the amount of network traffic that was produced made the manual monitoring of network traffic for intrusions inefficient. To overcome the

A. Nisioti, A. Mylonas, and V. Katos are with the Department of Computing and Informatics, Bournemouth University, Poole House, Talbot Campus, Fern Barrow, Poole, BH12 5BB, United Kingdom, E-mail: {anisioti, amylonas, vkatos}@bournemouth.ac.uk

P. D. Yoo is with the Centre for Electronic Warfare, Cyber and Information (CEWIC), Cranfield University, Defence Academy of the United Kingdom, Shrivenham, SN6 8LA, United Kingdom, E-mail: p.yoo@cranfield.ac.uk

shortfalls, misuse detection approaches using predefined attack patterns, were introduced. However, as the complexity and amount of new attacks increased, a different solution was needed and as a result data mining based approaches were brought in. At first, supervised methods were utilised for this purpose, but proved to be limited on detecting known attacks with very low false positive rate only. In recent years, unsupervised and hybrid (supervised or misuse combined with unsupervised) techniques are gaining more popularity. However, contrary to supervised IDS techniques, which have been extensively studied in the literature, there is no comprehensive review of the unsupervised and hybrid ones. This paper contributes by providing a comprehensive review of these techniques, also noting that they can raise the bar of security that the IDS as a countermeasure provides, as:

1) They can potentially detect unknown attacks (e.g., 0-day attacks) [35]. Supervised methods do not perform well on unknown attacks, but do well on known ones with a low false positive rate. Conversely, unsupervised methods are able to detect unknown attacks, but exhibit a high false positive rate. Therefore, combining the two approaches in a hybrid system may potentially result to both high detection and low false positive rates.
2) Unsupervised methods do not require the time consuming training of supervised methods on a regular basis.
3) Unsupervised methods do not require regularly data labelling for training purposes, which is both time consuming and resource demanding.
4) One of the most significant problems in a forensic investigation is the amount of data created from diverse sources, such as network traffic, host evidence and logs from different devices. The extraction of features from these evidence and their inclusion on the clustering process is much more convenient and efficient than the re-training of the supervised model.
5) Finally, as will be discussed in Section IV, the evolution of the attacks demands the extension of the IDS with the use of attribution and correlation. Clustering techniques fit well with this purpose, as the attribution questions we desire to answer can be described well by features extracted from diversified sources (network, host, log files, etc.), on which later the clustering process will be performed on.

However, it should be noted that this work does not suggest that unsupervised or hybrid IDS are the silver bullet of security for an organization. An attack might still occur even with the presence of an IDS for various reasons, e.g., due to the presence of a misconfiguration, a 0-day vulnerability or a non-security and technically savvy user. Moreover, as explained in [6], the problem of intrusion detection is intractable and in fact IDSs do not detect intrusions at all. They only identify evidence of intrusions and produce indicators of potentially malicious activities.

For this reason, further correlation of the detected instances is needed, to reconstruct the occurred attack and potentially identify parts of it and affected hosts that were previously missed [99]. For instance, a targeted attack consists of several stages, which may not be detected, especially if they do not produce network traffic. One of these stages could, for example, include host infection through a malicious USB device, which will certainly not be detected by the IDS, contrary to other instances of host infection that do require traffic, e.g., service exploitation with a buffer overflow. In both cases, these stages are parts of a larger attack. If other parts of the attack are detected and correlated, the missing stages could be identified manually or automatically through either the newly produced knowledge from the correlation or from other evidence sources. Consequently, we regard that further correlation of the detected instances, could lead to the reconstruction of the occurred attacks, the prevention of future ones and even the identification of the attacker. Such analysis is typically performed as part of the forensic investigation of an incident. However, with the extensive increase of produced data, this task is becoming overwhelming and manual reconstruction inefficient. As it will be discussed later in Section IV, time is critical factor in a forensic investigation, thus, combining data mining methods with the current forensics analysis techniques can be very effective. Moreover, such sophisticated and multistep attacks demand the bridging of technologies such as IDS with the forensic process.

The major contributions of the paper are as follows:

1) Contrary to most surveys that cover all the facets of existing IDS (i.e., signature, statistical, supervised, unsupervised, soft-computing, knowledge-based, etc.), but with limited focus on unsupervised and hybrid IDS techniques, this paper provides a comprehensive review of those methods and take it to the next level of intrusion correlation and attribution.
2) Particularly with machine learning based approaches, we believe that feature extraction, construction, and selection techniques play important roles as they influence their learning processes significantly. Unlike other studies, we devote a subsection to discuss the topic thoroughly.
3) We do not limit ourselves on the presentation of the reviewed works, but also compare and analyse them and identify their strengths and weaknesses. In this way, we are able to produce recommendations to be considered when developing an intrusion detection system in the future.
4) We discover open issues regarding the current IDSs and discuss how such systems need to evolve from simple detection to correlation and attribution, to be able to effectively cope with threats like APTs, targeted attacks, data exfiltration, etc.
5) To the best of our knowledge, we are the first to discuss the concept of correlation of the produced indicators from the intrusion detection stage, with the ultimate goal of revealing the identity of the attacker.
6) Finally, we identify modern attacks that do not fall in any of the four known IDS attack classes. We discuss the behaviour of those attacks, as well as identify and propose characteristics to be considered for detecting them.

The rest of the paper is organised as follows. Section II provides a brief background in intrusion detection. Section III presents and compares the different unsupervised and hybrid intrusion detection methods and feature selection techniques.

Section IV introduces the concept of correlation and Section V presents related work. Finally, Section VI concludes the paper and discusses directions for future work.

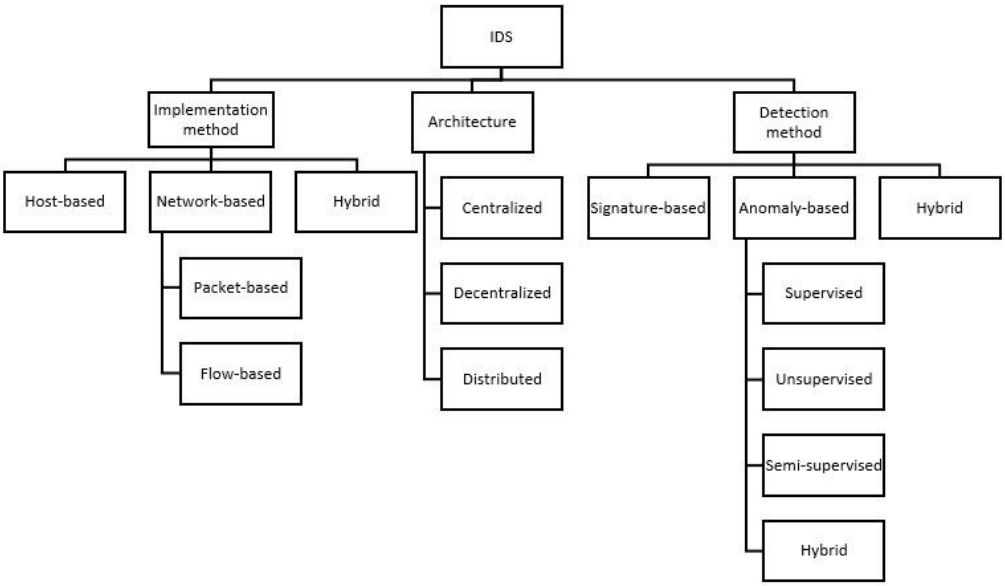

Fig. 1. General Classification of Intrusion Detection Systems

II. Background

This section discusses the fundamentals of intrusion detection technology according to its a) implementation method, b) detection mechanism and data analytic technique, and c) architecture as well as the most used measures and attack classes.

A. Implementation Classification

With regards to the implementation method, IDSs can be divided into two categories: a) host-based and b) network-based. A host-based intrusion detection system (HIDS) deploys a local agent on each host of the network. HIDS uses the local agents and application logs or raw system calls as data source to detect rogue processes, modification of critical system configuration files (e.g., registry keys), privilege escalations and any other unauthorized action that is against the system's policies. HIDS has the advantage of working with high quality data that are typically very informative [7]. However, processing the audit trail can have a significant impact on the performance of the host, when the data are processed locally [8], or on the bandwidth of the network, when a remote processing unit is used [9]. Lichodzijewski *et al.* in [10] successfully reduced the computational cost by utilizing "session information" instead of the traditional audit trails and so did Hu *et al.* in [11], by proposing a system based on a Hidden Markov Model with a pre-processing stage that removes similar sub-sequence system calls. Others who have proposed host-based IDSs are [12] and [13].

Although HIDS was the first area the literature explored, with the growth of computer networks and the proliferation of network attacks, the protection offered by HIDS was not enough, as they were restricted on a single host. Compared to a HIDS, a NIDS has several advantages [7]:

1. It is more resilient to attacks, as an HIDS depends on the logs produced by the system and other applications.
2. It is operating system and platform independent meaning that the same NIDS works on any platform without needing any modification.
3. It does not affect the performance of the network as it does not add any overhead to the network traffic by simply monitoring and processing it.

A NIDS monitors and analyses the network traffic on a packet or a flow level and attempts to detect anomalies, such as unauthorized access or DDoS attacks. On a packet level, an IDS performs a so called Deep Packet Inspection (DPI), which analyses both the header and the payload of each packet [7], [14]. Although inspecting the payload of the packets can be very informative, with today's high speed communication networks such approach is not only time consuming and inefficient, but also computationally costly. As will be discussed later in Section III.C, one of the desirable characteristics of IDS is the requirement of real-time, or near real-time performance. Moreover, in the case of encrypted packets, which are becoming prevalent with the rise of darknet [15] and the use of technologies like VPN, the analysis of the payload is not possible. On the contrary, a flow-based NIDS inspects only the packet headers and uses input data in the form of NetFlow or IPFIX [16]. A combination of both techniques could be used to improve the performance of IDS. For example, DPI could be applied only on the packets that were flagged as potentially malicious by the flow-based IDS. One of the main disadvantages of the NIDS is network scalability, i.e., the ability of the NIDS to adjust as the size and the complexity of the network is changing. This problem and its current proposed solutions will be detailed in Section III.C.

B. Detection Mechanism Classification

IDSs can be classified based on their detection mechanism as

[17]: (i) misuse or signature based, (ii) anomaly or behaviour based, and (iii) hybrid. Misuse or signature based systems maintain a database of predefined signatures (patterns) that correspond to known attacks and perform the detection by comparing these to the audit data stream. It is still the preferred method in today's industry, as it has a low false positive rate and constitutes an "out of the box" solution. There are many signature-based open source IDSs that are widely used in the enterprise world, such as SNORT [17], BRO [18] and Suricata [19]. Moreover, various security solutions exist that include network intrusion detection provided by known vendors, such as Unified Security Management (USM) [20] by AlienVault, Firepower Next-Generation IPS (NGIPS) [21] by CISCO and FireEye Network Security [22] by FireEye.

Despite its advantages a misuse intrusion detection system is as good as its database, which needs to be updated by a human expert and so it cannot detect unknown attacks [6]. Consequently, the performance of such an intrusion detection system is inseparably linked with the quality of its database. Also, as its size increases, so do the processing time (i.e. search time) and cost. According to Dreger *et al.* [23], popular misuse IDS, such as SNORT and BRO, consume a significant amount of resources (e.g. CPU and memory) when operating on a high speed network. Moreover, in today's ever changing and evolving cyber-physical ecosystems new attacks and more advanced variations of older ones appear on a daily basis. As such, maintaining an updated set of rules is not feasible, as well as is a time-consuming and insufficient process. Cheng *et al.* [24] tested five different IDS evasion techniques against known misuse-based systems like SNORT. Their results showed that this kind of IDSs are vulnerable to attacks like payload and shellcode mutation and simple variation of older attacks.

Anomaly detection refers to the problem of identifying instances in a dataset that do not conform to the "normal" behaviour. A behaviour is considered an anomaly or an outlier if the deviation from the "normal" one exceeds a predefined or dynamical calculated threshold. Anomaly detection finds extensive use in a wide variety of applications, such as fraud detection for credit cards, insurance, or health care, intrusion detection for cyber-security, fault detection in safety critical systems, and military surveillance for enemy activities [25]. It is important to stress that an anomaly does not necessarily correspond to an attack, but a suspicious observation. Anomaly-based IDSs do not rely on previously defined patterns, but they aim to model the normal behaviour/traffic in order to detect the abnormal, and so they are able to detect both known and unknown attacks. The price for that is the high false positive rate they produce and the tuning stage they require.

There are many studies in the literature aiming to combine the aforementioned techniques to inherit both of their advantages, improving the detection rate and minimizing the false positive rate. Barbara *et al.* [26] proposed one of the most well-known hybrid systems, named, the Audit Data Analysis and Mining (ADAM). ADAM has two stages of detection: it firstly uses association rule mining for the anomaly-based stage and then classifies the suspicious connections as normal, known attacks and unknown attacks with a misuse module. Kim *et al.* [27] combine the C4.5 decision tree for the misuse module with multiple one-class Support Vector Machines (SVMs) to model the normal behaviour. Similarly, Depren *et al.* [28], use self-organizing maps (SOM) for the anomaly module and C4.5 decision tree for the misuse module. Other proposed hybrid systems are the [29], [30] and [31].

C. Anomaly-based Intrusion Detection Systems

Anomaly-based IDS can be divided into the following categories according to the method they use: statistical, supervised (classification), unsupervised (clustering and outlier detection), soft computing, knowledge-based and combination of learners. This review focuses on the unsupervised and hybrid-based systems, while providing a short description of the supervised techniques. Detailed reviews on the rest categories can be found in [32], [33] and [34].

In a supervised IDS, a model is trained to learn from examples (i.e., labelled data). When a new instance is introduced, a classifier attempts to assign it to one of the predefined classes. Several classification algorithms, like Decision tree (C4.5), SVM, K-Nearest Neighbour, Bayes Classifier, Neural Networks, etc., have been used for network based intrusion detection tasks. According to Laskov *et al.* [35], the supervised algorithms exhibit excellent detection accuracy and low false positive rate on the detection of known attacks – with C4.5 achieving the best results. However, when unknown attacks are present in the dataset, most of the classification models fail to detect them, with SVM achieving the best results. Moreover, classification-based systems have a similar disadvantage to the signature-based ones, i.e., they need to be periodically trained to preserve their high detection rate. This is not feasible as it is extremely difficult to obtain labelled data, especially on a regular basis. In addition, even if labelled data do exist, it is uncertain if they include all the new attacks. In the past, lots of supervised models were proposed, like the ones in [36], [37], [38], [39] and [40].

Unsupervised anomaly detection (also known as outlier-based detection) uses clustering techniques to identify possibly malicious instances in a given dataset, without having any prior knowledge. The goal of clustering is to separate a finite unlabelled dataset into a finite and discrete set of "natural", hidden data structures, rather than providing an accurate characterization of unobserved samples generated from the same probability distribution [41]. In other words, clustering algorithms aim to partition the given data into groups (clusters) that achieve high inner similarly and outer dissimilarity, without any prior knowledge. For this purpose, all the clustering methods are based on the following assumptions. Firstly, the number of normal instances in a dataset vastly outnumbers the number of anomalies. Secondly, the anomalies themselves are qualitatively different from normal instances [43]. After the cluster formation, scores are assigned to the constructed clusters. If the score of a cluster exceeds the predefined or dynamically calculated threshold, it is considered as potentially malicious. Respectively, when clustering is used to detect network attacks, one is assuming that a) normal traffic outnumbers the malicious and b) normal traffic differentiates in some way from the malicious. For these reasons, as it will be explained later in this section, selecting the proper feature subset is of great importance. In other words, one has to select

the features that describe well enough the attacks to be identified. With regards to the detection process the goal of clustering is to group the network flows or packets without any prior knowledge, but solely based on the relations between them. As a result, large clusters of normal traffic will be created while malicious traffic will form smaller clusters and outliers, i.e. instances that do not belong to any cluster. Based on experiments and tuning of the algorithms in use, a dynamic or static threshold can be used to decide which clusters are considered malicious. The main advantage of clustering-based systems is their ability to detect unknown attacks without any prior knowledge, which eliminates the need for labelled data. The main drawback is the high false positive rate that they produce. A more extensive overview and comparison of unsupervised systems is provided in Section III.C.

One of the most important steps of the unsupervised anomaly detection is feature extraction or selection. Each instance in a dataset is represented by an array of characteristics, which are called features. Feature selection refers to the process of selecting a subset of the available features that are the most relevant and less redundant. On the other hand, the feature extraction aims to create (extract) new features of higher quality. Both processes can affect not only a system's detection rate, but also its performance. Section III.B provides a thorough discussion of the feature selection and extraction processes and compares the most commonly used techniques in network intrusion detection.

D. Architecture classification

The architecture of an IDS can affect its overall performance, thus is an important decision during the system's design. This is especially true due to the high speed networks that most organizations (e.g. companies, universities) use nowadays. Considering the architecture of the system, IDSs can be divided in the following three categories:

1) *Centralized*: Centralized IDSs consist of multiple sensors across the network that monitor and send data to the central processing unit (CPU), where the analysis of the collected data and the detection take place. This architecture has two main disadvantages. Firstly, it does not provide network scalability, which means that as the network expands the CPU is overloaded and at some point may become unable of keeping up with the workload. Secondly, one CPU constitutes a Single Point of Failure (SPoF) of the system [43].
2) *Decentralized*: In this architecture, multiple sensors and multiple processing units are scattered across the network, following a hierarchical structure. The collected data are sent to the closest processing unit, where they get pre-processed before they end up at the main processing unit. In this way, the SPoF and the scalability problems can be avoided. The performance of the system is also boosted due to the pre-processing stage.
3) *Distributed*: This architecture consists of a flat overlay of multiple autonomous agents that act as sensors and processing units at the same time. Data are collected and processed by the agents, which communicate with each other through a Peer-to-Peer (P2P) architecture [40]. In this architecture there is no main or CPU and the processing workload is distributed between all the agents, which boosts the system's performance and scalability.

In both decentralized and distributed architectures communication between the agents is crucial for the detection of certain types of attacks. For instance, the loss of communication between the agents may lead to the inability of the system to detect distributed attacks.

E. Measures

In the past different metrics and datasets have been used to measure how good a system is at successfully identifying attacks and normal traffic in a dataset, which makes it difficult to compare the results of the various proposed systems. The most common evaluation measures regarding the detection ability of the system are as follows:

1) *Confusion matrix*, also known as error matrix, is a way of visualizing the relationship between the actual results and the predicted results. It is mainly utilized in supervised learning to evaluate the prediction accuracy of a classifier. Each row of the table corresponds to a result predicted by the classifier, while each column corresponds to an actual result.
2) *Recall* represents the portion of the relevant instances (*i.e.* true positives) which are successfully retrieved. Conversely, *precision* is the proportion of retrieved instances that are correctly identified. Both recall and precision focus on the positive samples, but neither of them captures how well the model handles negative cases [43]. The harmonic mean of the two previous measures is called *F-measure* (F1). Although F1 is advocated as a single measure to capture the effectiveness of a system, it still completely ignores True Negatives (TN) [44].
3) *Accuracy* takes into account both the true positives and negatives and is defined as the ratio of the correctly classified samples to the total number of instances.
4) *Sensitivity*, also known as True Positive Rate (TPR), is the proportion of positives samples that are correctly classified as such. On the contrary, specificity or True Negative Rate (TNR) measures the proportion of instances that are correctly classifies as negative. Similarly, False Positive Rate (FPR) represents the proportion of sample that are incorrectly identified as anomalies.
5) The *Receiver Operating Characteristic* (ROC) is a technique originally used in signal processing theory for visualizing the TPR against the FPR for different parameter settings. It depicts relative trade-offs between benefits (true positives) and costs (false positives) [45].

Although the accuracy of an IDS is one of the most important requirements, it is not the only one. The response time of the system is a significant factor, as it will be used in fast enterprise networks where even a small latency can result in monetary

losses for an organization. Furthermore, the computational and communication cost (between the agents and the processing units), can not only negatively affect the response time, but also the financial cost of deploying and maintaining the system. As today's networks are large and their size dynamic and variable, the IDS should possess the ability to adjust to the changes of the network's size and structure. Finally, a system destined to protect other systems should be itself resilient to any attacks that aim to disrupt its operation and have a stable and consistent performance under different scenarios.

F. Attack Classes

Four main categories of attacks have been proposed in the intrusion detection literature, which an IDS needs to be able to detect:

1) *Denial of Service (DoS)*: In a DoS attack the targeted system is flooded with a large amount of requests originating from a single connection, until all the target's resources are exhausted and thus is not capable to handle legitimate requests anymore. In a *Distributed Denial of Service* (DDoS) attack the attacker is using multiple connections that are distributed across the Internet and are likely part of a botnet network. Attacks of this kind, target the availability of an infrastructure by making a service or resource unavailable to its users. Small or medium DoS and DDoS attacks are often used as smokescreens by the attackers to conceal smaller but more dangerous malicious activities, or to take down security appliances, such as firewalls.
2) *Probe*: This type of attack (*e.g.,* port scanning) is used to explore the target network and collect information about the hosts, such open ports, running services, etc.
3) *User to Root (U2R)*: In this case, the attacker already has local access to the targeted system and aims to exploit a system vulnerability to escalate her privileges from those of a simple user to super user/admin. One of the most common U2R types is buffer overflow, in which the attacker tries to overfill a buffer and execute malicious code under root privileges.
4) *Remote to Local (R2L)*: In this attack class, the attacker does not have an account in the targeted machine and tries to gain local access. Remote to local attacks are usually combined with U2R attacks. An example of a R2L attack is SSH brute force.

III. Methods And Systems For Unsupervised And Hybrid Intrusion Detection

This section presents a comparison of unsupervised and hybrid methods, as well as feature techniques used for intrusion detection. Moreover, we summarize the different datasets that have been used by researchers for intrusion detection. Finally, a review of the limited research on attack reconstruction and correlation is presented.

A. Collection Methodology

This subsection describes the methodology that we used for compiling the list of papers that were considered. In the case of papers focusing on intrusion detection our target was to collect those published in the last five years that use either unsupervised or hybrid (combinations of supervised and unsupervised) techniques. Feature selection literature is somewhat relatively limited, therefore we included the most known or promising works in our list, but again tried to limit ourselves to the recent seven years only. Firstly, an initial pool of papers was created based on:

- Searches on Google Scholar, IEEE Xplore and ACM Digital Library with keywords like "unsupervised intrusion detection", "anomaly intrusion detection", etc.
- Browsing the proceeding of top security conferences and journals like ACM Symposium on Computer and Communications Security, IEEE Communications Surveys & Tutorials, IEEE Symposium on Security and Privacy, Network and Distributed System Security Symposium.
- Recommendations of specific papers based on the authors' personal knowledge.

The selection was expanded based on:

- Considering papers that were in the reference section of the already selected papers.
- Browsing the proceeding of conferences or journals in which the selected papers were published, only if they have not been considered.

B. Feature Selection

Over the past decades, many researchers have attempted to improve the detection rate and performance of IDS, by focusing on the detection algorithm and proposing different techniques or by combining both. However, they may have neglected the process that must be preceded, *i.e.,* feature selection (FS). FS is the process of identifying an optimal subset of the relevant features that represent each class better than the original set. In many cases feature selection is more important than the choice of the detection algorithm.

Using an optimal subset that describes efficiently the input data, instead of the whole feature space can not only enhance the accuracy of the system, but also decrease the false positive and computational time. FS does not create new features, but selects the ones that are relevant and non-redundant. The inclusion of irrelevant and redundant feature in the classification or clustering process can lead to poor generalization and overfitting [46]. The feature selection process has two main components: a search strategy and an evaluation criterion. The chosen search strategy is responsible for selecting the features to be considered as part of the optimal subset. The evaluation criterion assigns a score to each feature. If this score exceeds a threshold, then it is considered relevant and included in the subset.

FS methods can be divided in two main categories: filters and wrappers. Filters do not take into consideration the classification technique, but assign score to the proposed features with statistical and information theory methods, such as mutual information, information gain, correlation coefficient, and information entropy. Therefore, a filter is a fast and simple method. In contrast a wrapper, whose general methodology can be observed in Figure 2, evaluates the candidate subset with a predicted model based on the detection algorithm in use. In each iteration a feature subset is used by the classifier on the training set and depending on the results the features are either accepted or rejected. Although wrappers take the detection algorithm into consideration and consequently produce a subset adjusted to the specific algorithm and IDS, they can cause overfitting and can be computationally intensive, especially in the case of network data, which are high dimensional.

Wrapper feature selection processes differentiate according to the detection technique. In the case of supervised learning, the classes are predefined and the data are labelled. Consequently, it is easier to evaluate the proposed subset after the classification process has been applied. Selecting features in unsupervised learning scenarios is considered to be a much harder problem, due to the absence of class labels that would guide the search for relevant information [47]. In such cases, the most common criterion is the cluster's quality, their intra and inter cluster distances.

A comparison of several feature selection methods for intrusion detection is given in Table I.

With regard to the use of FS in [48], Fahad *et al.* compared six different techniques: Information Gain (IG), Gain Ratio (GR), Principal Component Analysis (PCA), Correlation based Feature Selection (CBF), Chi-square, and Consistency-based search (CBC). For the evaluation of the methods three measures were chosen: *i)* goodness, which corresponds to the detection accuracy, *ii)* stability, which evaluates the robustness of the subset to the variation in the traffic data and *iii)* similarity, to compare the behaviour of different FS techniques on the same dataset. According to their results, no feature selection technique can be considered as the "best" along all the metrics and datasets. In more detail, CBF achieved the highest goodness value on all of the datasets except one. Also, Chi-square and IG achieved very high values on many datasets. The lowest results were achieved by GR and CBC. Regarding the stability almost all the techniques were considered unstable. The best results (0.87%) were achieved by IG and the worst by CBC and CBD, as they do not highly consider the interdependencies of the features. The similarity between the six selected techniques was low in all cases, which suggests that one subset that is optimal for one technique can be considered not optimal form another. All the above lead to the conclusion that one FS technique cannot satisfy all the criteria across all the datasets. Consequently, the writers propose the Local Optimisation Approach (LOA), a combination of five FS methods. PCA was excluded as it transforms the features and therefore falls under the feature extraction category. LOA firstly extracts an optimal feature subset for each one of the 5 FS techniques and then

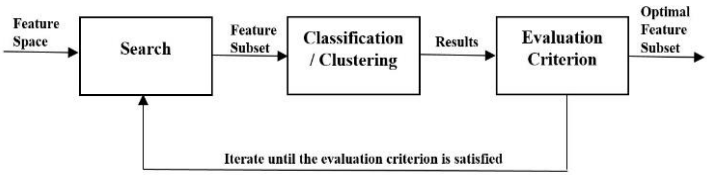

Fig. 2. Wrapper Method Topology

calculates the support for each feature. Finally, if this value is higher than a predefined threshold, the feature is included in the final optimal subset.

Moreover, Fahad *et al.* [49] improved their previous method by proposing the Global Optimization Algorithm (GOA). Firstly, the five aforementioned FS techniques were combined to filter out the irrelevant features. Then, an adaptive threshold based on maximum entropy was used to select the robust features from the unstable subset produced by the first stage. Finally, a Random Forest filtering was applied that utilised forward sequential selection to guarantee the quality of the final subset and avoid overfitting. GOA outperformed Backpropagation Neural Network (BNN) and Fast Correlation Based Filter (FCBF-NB) not only by increasing the detection rate, but also by producing a smaller and highly descriptive subset that decreases computational time. The main advantage of this approach is that it combines multiple techniques and thus the one that performs better in terms of stability and optimality is chosen each time. Moreover, using an adaptive threshold instead of a static one makes the method more robust against the network drift.

Liu *et al.* [50] proposed a two phased feature selection technique, called Class-Oriented Feature Selection (COFS). Phase one searches for an optimal subset for every attack class in two steps: firstly calculates the weighted symmetric uncertainty (WSU) for each value and the local correlation metric (LCM) between each feature and each class. Using both a local and a global metric ensures not only relevancy between features of the same class, but also among different ones. The second phase uses a predefined threshold to select a feature subset for each class. This is accomplished by removing the redundant features from all selected subsets. The disadvantage of this approach is the use of a static threshold in contrast with [49]. Their results illustrated that the first phase improves the detection accuracy of the classifier, whereas the second decreases the computational cost. COFS was compared with other known FS schemes like Global Optimization Algorithm (GOA), WSU_AUC and BFS and in most cases had the best or second best results, with GOA having the worst.

Similarly, Zhang *et al.* [51] proposed a wrapper method that combines WSU for prefiltering most of the features and the area under the ROC curve (AUC) to choose the optimal features for a specific classifier. At the last stage, Selection Robust Stable Features (SRSF) algorithm was used to choose the most robust feature subset of the previous step. According to their results, this approach improved the TPRs for most minority classes and reduced the FRP for the majority class. Moreover, according to the authors' experiments, the server port, the total number of bytes sent in the initial window and the minimum segment size seem to be the three most important features across the different

datasets.

De la Hoz *et al.* proposed a multi-objective wrapper method in [52] that combined NSGA-II [53] and the Jaccard's coefficient as the evaluation criterion for selecting the optimal subset of features. In each iteration, a new subset is selected for each of the five classes (DoS, Probe, U2R, R2L, Normal). Each subset is then evaluated through the classifier using the Jaccard coefficient and the population is evolving through the calculation of the Pareto front of the five non-dominant solutions (subsets). Results showed that the use of the selected subset provided better accuracy for all the classes and particularly high performance for the U2R and R2L classes, which are considered the less probable and so the most difficult to detect. Compared to other FS methods, such as [49] and [51], the proposed method selects a different subset for each class (attack or normal), which leads to higher accuracy. Similarly, authors in [54] proposed an improved version of NSGA-III, called I-NSGA-III, for feature selection. The proposed scheme overcomes the imbalance problem using bias-selection based on probabilities and removes redundant features using fit-selection. As before the evaluation criterion is the Jaccard coefficient. For the evaluation of the produced feature subset, the authors use GHSOM on the detection stage. According to their results, NSGA-III+GHSOM has slightly better overall performance (99.27%) in terms of detection accuracy than I-NSGA-III+GHSOM (99.24%). However, the main advantage of the proposed method, which is not reflected in the overall detection rate, is that by solving the imbalance problem it produced higher accuracy for the smaller classes (U2R and R2L). Finally, I-NSGA-III+GHSOM produced a smaller subset of features, which leads to less computational time.

A wrapper-based method was proposed in [55] by Li *et al.* that combines a Modified Random Mutation Hill Climbing (RMHC) algorithm with multiple linear SVMs to build a lightweight IDS. Firstly, an initial subset is generated followed by an iterative process of using modified RMHC to generate a subset and linear SVMs to compare it with the previous one. The current best subset is considered optimal if the iterations reach the maximum number or the predefined criterion is satisfied. Modified RMHC was chosen to improve the wrapper's dimensionality reduction ability and decrease its computational complexity. The experiments showed a significant speed up of the feature selection process and improvement of the overall detection ability of the system. Moreover, similarly to [50] and [52], RMHC constructs one subset for each attack and normal class. However, unlike [48], [49] and [50], the method only uses one evaluation criterion instead of multiple ones.

Information theory and statistical criteria used by Amiri *et al.* [56] create a FS method that selects features with maximum relevancy and minimum redundancy. Modified mutual information-based feature selection (MMIFS) algorithm selects the feature with the maximum mutual information (MI) as the first element of the subset and then uses a greedy approach to select features depending on their feature-to-feature MI. The proposed method was compared with the linear correlation-based feature selection (LCFS) and the forward feature selection (FFSA). It proved to be the most effective among the three in detecting R2L and Probe attacks and FFSA was the most effective for DoS, U2R and Normal. Likewise, the approach in [57] uses mutual information for feature-class relevancy and generalized entropy to achieve feature-to-feature non-redundancy.

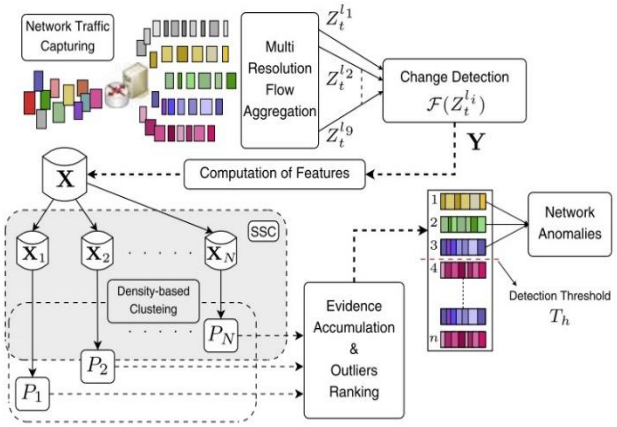

Fig. 3. Topology of proposed IDS in [59]

The following conclusions arise from the study of the literature:

1) One feature selection technique is not enough to achieve stability through the different datasets, as the behaviour of the network traffic is shifting ([48] ,[49] ,[50])
2) For each of the five classes of intrusion detection, one optimal subset should be obtained as one global feature subsets is not capable to describe satisfactorily all the different classes ([50], [52], [55]).
3) FS can considerably improve not only the detection rate, but also the computational performance. As mentioned earlier, features that are irrelevant or redundant may lead to poor generalization and overfitting. Moreover, more features regarding each data point translate to higher computational cost and complexity ([49]-[55]).
4) Finally, U2R and R2L classes are considered the most difficult to detect because they are discrete enough and they can be misidentified as normal traffic. Experiments showed that FS can offer the solution to this problem by identifying a feature subset adjusted to the characteristics of each class ([52], [54], [55]).

C. Unsupervised and Hybrid IDS

As discussed in Section II, unsupervised learning attempts to distinguish malicious traffic from normal without any prior knowledge. In this subsection the unsupervised and hybrid (combinations of supervised and unsupervised) approaches, which have been published in the last 5 years, are presented and compared (Table II). The reviewed papers and their proposed methods in this subsection regard a typical corporate network. The reader may refer to Section V for security solutions that focused on different network types, such as WSN and SCADA, which fall outside the scope of this work.

Casas *et al.* [58] propose a Sub-Space Clustering and Evidence Accumulation algorithm (SSC-EA) using DBSCAN as the clustering method. Instead of partitioning the whole

feature space, SSC-EA divides the feature space X in N different sub-spaces Xi ⊂ X of smaller dimensions and applies DBSCAN on each partition. The results from the clustering of the multiple sub spaces are then combined to a new similarity measure, which is capable of clearly highlighting both the outliers and small-size clusters that were simultaneously identified in different sub-spaces. Later in [59], the authors used their proposed clustering techniques into a complete intrusion detection system, whose topology can be found in Figure 3. Firstly, network packets are captured and converted into flows. Next, flows are being aggregated at different flow-resolution levels before arriving to the change detection module, which uses a time series criterion for detecting a potentially malicious flow and activating the clustering module. The use of multiresolution flows enables the system to detect both small, large, single sourced and distributed attacks. If a change is detected, SSC-EA clusters the data and ranks the produced clusters. Finally, a predefined threshold is used to decide if a cluster is malicious or not. Their results indicate that the proposed methodology can achieve high detection rates not only for the large volume attacks like DoS and Probe, but also for the U2R and R2L. This is most likely due to the use of multi resolution flows and sub clustering for the detection of smaller attacks, which can be cloaked by largest ones. Moreover, the use of sub clustering enables system parallelization, lowering the computational time and resulting to real time detection.

Amoli *et al.* [60] created a two engine unsupervised intrusion detection system based on SSC-EA. The first engine is responsible for the intrusion detection using SSC-EA with a Dynamic Self-Adaptable Threshold, which is computed based on the previous behaviour of the network. To avoid losing small attacks the threshold takes in consideration four different network prefixes: /0, /8, /16 and /24. The second engine is a botnet detection module that clusters specific network features to detect centralized and decentralized botnet C&C communication. Different time windows for the previous behaviour were tested and the optimal value was observed to be five minutes. Moreover, the proposed model achieved a 98.39% accuracy and 3.61% FP rates after optimizing the threshold parameters and outperformed the classic DBSCAN and K-means implementations.

Bohara et al. [61] used both K-means and DBSCAN for hybrid intrusion detection, using system and network logs. Firstly, the performance of both algorithms was evaluated with firewall data. When K-means was used, data points of different feature distributions were clustered together and smaller clusters were absorbed into their larger neighbouring clusters. That was likely due to the non-normal distribution of the data. Conversely, DBSCAN performed exceptionally on the data and so it was used for the rest of the experiments. In addition, system and network logs were combined to study the effect of host features in the detection ability of the system. The results prove that some attacks can only be detected through the combination of host and network features.

Bhuyan et al. [57] proposed a multistep outlier-based detection approach, which utilizes their previous work in [62] and [63]. The proposed framework consists of a mutual information and generalized entropy feature selection (MIGE-FS), which was presented in section III.B, a tree based subspace clustering technique called TreeCLUS (TCLUS) and an anomaly detection, based on the ROS' score. As discussed in the previous subsection, MIGE-FS attempts to select the most relevant optimal subset of features in each case to reduce the cost and improve accuracy. TCLUS algorithm generates a tree where each node represents a cluster. At first, all the features are part of the root node. Then, the algorithm divides the feature space in more nodes with respect to the maximum feature-class relevancy and the minimum feature-feature redundancy in a depth-first manner. When clusters are fully formed a reference point is calculated for each of them in order to be used for the profile creation. In the last stage, an outlier score ROS' is calculated for each cluster with respect to its profile and reference point. If the score exceeds a predefined threshold, the cluster is considered anomalous. The proposed methodology outperformed known approaches like C4.5 and ID3, with an especially higher detection rate in Normal and U2R classes.

Optimum Path Forest Clustering (OPF) was deployed by Costa *et al.* in [64]. To overcome the OPF's problem with different concentrations and scales and to speed up the detection, the authors optimize OPF through different nature-inspired optimization techniques. The best results were obtained by Particle Swarm Optimization (PSO) and Bat Algorithm (BA).

Bostani *et al.* [65] used a modified Optimum Path Forest (MOPF) algorithm, which consists of three modules for partitioning, pruning and detecting. The first module utilizes k-means to create training subsets to be used later by the detection module. The pruning module is responsible for pruning the training subsets by identifying the most informative samples in order to improve the speed of OPF. Finally, the detection module is based on an advanced OPF algorithm which achieved 14.86% better performance than the original OPF and training time 6.9 times less.

An intrusion detection system based on combining cluster centres and nearest neighbours (CANN) was introduced in [66] by Lin *et al.* Firstly, the cluster centres are extracted using a clustering technique. As the second step, the distance measures are calculated, namely the distance between all data of the given dataset and the cluster centres (dist1) and the distance between each data point and its nearest neighbour in the same cluster (dist2). The sum of dist1 and dist2 leads to a new one dimension feature. Finally, k-NN is used to classify the data represented by the newly constructed feature. Experiments show that the approach succeeds in detecting DoS and probe attacks, but does not perform well on U2R and R2L.

Unsupervised learning and artificial immune system were combined by Hosseinpour *et al.* in [67] to create a distributed hybrid IDS. Unsupervised learning is used as the primary innate immunity, where clustering divided the data in self (normal) and non-self (attack). Next, supervised learning represents the secondary adaptive immunity, where detectors are generated based on the clustering results and are distributed on the hosts of the networks when they become mature. These detectors can be used in the future by the hosts to stop known attacks.

Another immune system model for intrusion detection is presented by Jha *et al.* in [68], which consists of two layers: a T-cell and a B-cell layer. At the first layer a probabilistic model, which utilizes a Hidden Markov Model (HMM), identifies possible attacks. At the second layer, a decision tree uses the output from the previous phase together with each own feature

recognition algorithm to confirm true attacks. While the second stage is unsupervised, the first one requires the training of the T-cells. Finally, one advantage of the proposed model is its adaptiveness, as the T-cell utilize previous experience to enhance the detection rate.

The authors in [69] tried to exploit the advantages of both the misuse and anomaly detection by combining random forest and weighted K-means. The proposed methodology consists of two phases: the online and the offline. At the online phase, the traffic is compared against the misuse signatures through the random forest algorithm and if no match exists the connection is sent to the offline module, where the anomaly detection module will try to decide if it is a novel attack or normal traffic using k-means. Moreover, the offline phase generates signatures from the results of the anomaly detection, which are used by the misuse module.

A Particle Swarm Optimization (PSO) approach using the MapReduce technique was proposed in [70]. After the data pre-processing, the PSO was used for the clustering process by taking in account the global optimal centroids. The use of the MapReduce technique gives the IDS the ability to adjust on a large scale network and be parallelized in order to reduce the computational time. The proposed system achieved a 0.963 AUC with a 0.013 FPR, but its capabilities are limited on distinguishing the normal from malicious traffic, but not the specific attack classes.

Song *et al.* in [71] present a hybrid IDS composed of a training and a testing phase, which aims to detect attacks through modelling the normal behaviour. In the training phase, data are firstly filtered to isolate the normal traffic and then they are clustered. For every formed cluster, an one-class SVM model is created. In the testing phase, the opposite process takes place. The data are clustered and the formed clusters are compared with the previously created one-class SVM models. If a cluster does not match any of the models then it is considered an attack. The disadvantage of this approach is that normal behaviour of a network is constantly shifting as new applications and hardware are added, which could lead to a high false positive rate. Moreover, the training stage should be performed regularly, which is computationally costly.

Ashfaq *et al.* [72] propose a semi-supervised divide-and-conquer model that uses the magnitude of fuzziness to categorize unlabelled data. The authors used neural network with random weights (NNRw) as the classifier. Although the proposed method does not require a great amount of labelled data, the model still needs to be trained and has an accuracy of 84% ($KDDTest^{+}$) and 68% ($KDDTest^{-}$).

An intrusion detection system that uses K-means, SVM and fuzzy neural networks was proposed by Chandrasekhar *et al.* in [73]. The methodology consists of four stages: firstly K-means is used to generate training clusters from the initial dataset. For each training subset a different neuro-fuzzy model is trained and subsequently a vector for SVM classification is generated. Finally, a radial SVM classifier is used to detect the attacks. The proposed method achieved especially high rates in the two low-frequency attack classes, U2R and R2L.

Gogoi *et al.* [74] proposed a multistep approach that consists of three stages. Firstly, the CatSub+ classifier is used to detect DoS and Probe attacks. Then, K-point clustering is used to detect the normal traffic and finally GBBK outlier based classifier detects the R2L and U2R attacks. Both flow and packet level detection were performed in the experiments and flow level achieved higher accuracy in all classes.

The proposed method in [75] consists of three modules: feature selection, unsupervised clustering and supervised classification. Firstly, an entropy based feature selection method is applied on the data to remove the irrelevant features with poor prediction ability and the redundant features that are inter-correlated with one or more features. Then, k-means is used to cluster the data into normal and anomalous. Finally, k-NN and Naïve Bayes classifiers detect the specific attack class of the anomalous instances. The detection rate of the proposed method reached 98.18% with a 0.830% FP rate.

A graph-based intrusion detection algorithm that uses an outlier detection method based on local deviation coefficient (LDCGB) was introduced in [76]. LDCGB uses graph-based algorithm (GB) to cluster the data and an outlier method based on the local deviation coefficient to decide which clusters are malicious and which are normal. The advantage of this method is that compared to other approaches this algorithm does not depend on the initial cluster number and is able to detect arbitrary shaped clusters.

The following conclusions and suggestions arise from the aforementioned literature:

1) As can be observed in Table III, network traffic data are high dimensional. Specifically, most datasets have between 19 to 50 features for each data point (Table III). Moreover, as the technology evolves this number could be increased even more. Consequently, clustering using the whole feature space is not only time consuming, but can also lead to poor detection rate. Thus, feature selection is crucial as explained in section II.A.
2) Clustering algorithms that have the ability to detect clusters of arbitrary shapes, like DBSCAN, perform better than the ones that can detect only circular clusters ([57]-[61]). This is also confirmed from Buczak et al. [111].
3) The parallelization of the detection method could reduce the computational time and lead to real-time detection. For this reason, high performance cloud computing techniques could be utilized, to achieve better time performance and distribute the processing workload.
4) The use of host data can improve the detection rate, but one has to consider if the extraction of these data from a large-scale network in real-time is feasible ([61], [67]).
5) The need of initializing and tuning the parameters of the system is one of the main drawbacks that keeps these techniques from being applied in the industry [33]. Therefore, density-based methods such as DBSCAN that require less parameterization appear most suitable for deployment in real networks.
6) The combination of unsupervised and supervised techniques could potentially lead to a high detection rate and a low false positive rate ([67]-[69], [71]-[75]).
7) U2R and R2L attacks are the hardest to detect as they resemble normal traffic. Sub clustering, feature selection, such as [52], [54] and [55], and the use of different network resolutions seems to improve the detection rate of these attack classes. Moreover, the combination of neural networks [77] and unsupervised

methods, as in [73], performs very well in these two low-frequency attacks.

D. Datasets

Realistic network traffic is a prerequisite for developing, testing, tuning and evaluating IDSs. As new attacks and network technologies arise, there is a compelling need for newly captured high speed traces. Most of the datasets used in intrusion detection are outdated and do not reflect real-world conditions, as discussed in [78]. In this subsection the different datasets that have been used by researchers for intrusion detection are presented and summarized in Table III.

1) UNB ISCX [78]: The UNB ISCX IDS 2012 dataset consists of labelled network traces, including full packet payloads in pcap format, which are publicly available along with the relevant profiles. Real traces were analysed to create profiles, which contain detailed descriptions of intrusions and abstract distribution models for applications, protocols, or lower level network entities. These profiles were used by agents to generate real traffic for HTTP, SMTP, SSH, IMAP, POP3, and FTP. The dataset consists of seven days of network traffic, namely:
   - Three days of non-malicious traffic.
   - One day containing a U2R infiltration from an inside attack and normal Activity.
   - One day of DDoS attacks using an IRC Botnet.
   - One day of HTTP DoS and normal activity.
   - One day of normal activity and R2L attack (Brute Force SSH).
2) ISOT Botnet [79]: The ISOT dataset is a combination of three previously existing datasets:
   - Two datasets from the French chapter [80] of the honeypot project, containing Storm and Waldek botnet traffic were used to simulate the malicious traffic.
   - Traces from the Traffic Lab of the Ericsson Research [81] in Hungary and the Lawrence Berkeley National Lab [84] were used to simulate the normal traffic.
3) CAIDA [82]: These datasets contain passive traffic traces from CAIDA's equinix-chicago monitor on high-speed Internet backbone links. Traffic traces are anonymised and the payload is removed. Each yearly dataset (2014-2016) contains one trace per quarter
4) MAWI [83]: This dataset contains packet traces from the WIDE backbone, which connects several research institutions in Japan as well as commercial ISPs and universities in US. The traces are unlabelled and all the IPs are scrambled to protect user anonymity.
5) LBNL [84]: The Lawrence Berkeley National Lab (LBNL) dataset consists of more than 100 hours of anonymised network activity from a total of several thousand internal hosts for the time period of October 2004 to January 2005.
6) UNIBS [85]: These traces were collected on the edge router of the campus network of the University of Brescia on three consecutive working days and contain traffic from twenty different workstations. The dataset is composed of 79000 flows of TCP (99%) and UDP (1%) anonymised and payload-free traffic.
7) DARPA [86]: DARPA dataset consist of labelled traffic from two experiments: LLDOS 1.0 and LLDOS 2.0 conducted at MIT Lincoln Laboratory. Both scenarios include an attacker probing the network, breaking into a host, installing malicious Trojan software and launching DDoS attacks.
8) KDD99 [87]: KDD99 is the most widely used dataset in intrusion detection. It was first prepared by MIT Lincoln Labs for the Third International Knowledge Discovery and Data Mining Tools Competition and consists of five million connection records on the training set and 2 million records on the testing set. It contains a total of 24 training attack types, with an additional 14 types in the test data only and its flow is described by 41 features.
9) NSL-KDD [88]: As discussed in [89], KDD99 has two important issues that highly affect the performance of evaluated systems, and result in a very poor evaluation of anomaly detection approaches. For this reason NSL-KDD was proposed, which consists of selected flows of the KDD99 and has the following advantages:
   - It does not include redundant records in the training set, so the classifiers will not be biased towards more frequent records.
   - There are no duplicate records in the proposed test sets. Therefore, the performance of the learners is not biased by the methods that have better detection rates on the frequent records.
   - The number of selected records from each difficulty level group is inversely proportional to the percentage of records in the original KDD data set. As a result, the classification rates of distinct machine learning methods vary in a wider range, which allows more accurate evaluation of different learning techniques.
   - The number of records in the training and testing sets are reasonable, which makes it affordable to run the experiments on the complete set without the need to randomly select a small portion. Consequently, evaluation results of different research works can be consistent and comparable.
10) TUIBS: The TUIBS dataset is at the time of writing this paper inaccessible and therefore is excluded from the discussion. For more information the reader may refer to [32].
11) METROSEC [90]: These traces were collected as part of the METROSEC project from the French RENATER network and contained simulated DDoS attacks. Since the project ended in 2006 the dataset is not available anymore as it was considered outdated by its creators.
12) DEFCON: The DEFCON dataset contains traffic from a CTF exercise, but is currently inaccessible. For more information the reader can refer to [32].

In conclusion, the most well-known and commonly used datasets are the various versions of the original KDD99 ([87], [88]), which nowadays are obsolete as they do not reflect neither the current types of attacks, nor the methodology of the attackers. However, other datasets, namely [78]-[82], are more representative of the current threat landscape. Another

drawback for many researchers could be the format of the dataset. While the well-known KDD99 is already pre-processed, other datasets such as [78] are in raw (.pcap) format. Therefore, they require technical skills to perform the process of feature engineering. Nevertheless, there is a constant need for the construction of new datasets for intrusion detection, which are both realistic in terms of topology and balance, as well as follow the current attacks and trends. Finally, it is evident that currently there is no dataset that includes network traffic as well as logs and host evidence.

E. Correlation

As discussed earlier, the need for further correlation of the detected malicious instances has emerged the last years. Nonetheless, the relevant literature is rather limited. This subsection presents published works in attack reconstruction and correlation of the indicators produced by the IDS.

One of the first and few approaches on the detection and identification of multistep attack scenarios was presented by Cheung *et al*. [91]. The authors present the Correlated Attack Modelling Language (CAML) and showcase how it can be used to model multistep attack scenarios. The methods and the models produced by CAML are extensible in order to handle new attacks and data types and independent of the sensor technology. Limmer *et al.* [92] present a classification of event correlation techniques published in the literature and present a requirement analysis from an early warning system's point of view. In [93], the authors apply network attack graphs for correlating events and constructing scenarios. They map known exploits to events through the graph and use inverse distance between each event in a path as the correlation measure. A low-pass filter was also applied to the sequences of inverse distances in order to provide resiliency against detection errors. Also, a threshold was used to the filtered distances in order to separate the event paths into highly correlated attack scenarios. Dwivedi *et al.* [94] attempt to correlate SNORT events with the same alert name but different source IP. They succeeded on removing the duplication of the alerts, which is useful in reducing the workload of the security administrator. In [95] the authors develop a Process Query System (PQS), which is able to scan and correlate distributed events with the use of signatures and temporal-based event correlation. Unlike the others, Morin *et al.* [96] propose the application of chronicles instead of rules on event correlation in intrusion detection. They illustrate how this approach may be able to decrease the alarm overload and the false positives. Finally, in [97] Quicksand, a decentralized approach for gathering and correlating events from multiple points within the network is proposed. The system considers an intrusion as a pattern of events that occurred on different hosts of the network and uses signatures to identify patterns of an attack.

IV. Open Issues

As discussed in the precious sections, each type of detection mechanism has its own disadvantages. Signature-based systems are as good as their databases, which are hard to keep updated especially nowadays. Supervised methods perform well only on known attacks and also require regular training with labelled data. Unsupervised methods are capable of identifying unknown attacks without regular training, but have a higher false positives rate. Moreover, regardless the detection mechanism, designing a system with not only the ability to scale in networks of different sizes and structures, but also to perform satisfactorily is not a trivial task [32]. Thus, although many systems and methods have been proposed for intrusion detection in the relevant literature, designing a robust, scalable and high-performance IDS is still challenging [32].

The majority of the research to date focuses on developing a system capable of producing attack indicators. These attack indicators are first-level isolated security alerts, based on the observation of activity that corresponds to a single attack step (exploit, probe, or other event) [91]. However, the shift in the threat landscape over the last few years demands the evolution of such a system from simple detection to correlation and attribution. The rest of this subsection identifies open issues related to them and provides directions for future research.

A. Attack Reconstruction

As previously discussed, IDSs identify potentially malicious instances, but do not actually detect the complete chronicle of an attack. Although this was sufficient in the past, today it is not due to the rise of DDoS and ransomware attacks, botnets, Advanced Persistent Threats (APT) and the use of privacy enhancing technologies (PET), such as Virtual Private Network (VPN) and Tor. As the current widely used security technologies (IDS and antivirus) are not capable of detecting such targeted and advanced threats as APTs [98], the need for a new generation of intrusion detection and prevention tools arises. Consequently, there is a compelling need for further correlation of the indicators produced by the IDSs and other similar technologies, such as integrity checkers and Web Application Firewalls (WAFs), in order to identify every step of an attack and be able to reconstruct it. As it will be discussed in depth later in this section, sophisticated attacks such as APTs consist of multiple steps which can be mapped on a kill-chain, a model containing the different stages of an attack [99]. Thus, the correlation of data from different sources (such as logs, performance indicators, network traffic) can enable one to identify and group as many of these steps as possible. These techniques could bring a plethora of benefits. Firstly, a holistic view of the cyber threats and network attacks against a business could be provided, whereas current IDS can only provide individual indicators. This is important as in most cases the severity and significance of an incident can only be perceived through examination of the attack as a whole. Similarly, attempting to reconstruct an attack can trigger further investigations and reveal undetected parts of it, such as live connections that can potentially offer traceability to the attacker, other infected hosts, etc. For instance, in case of a botnet infection, reconstructing the attack may lead to the identification of other infected hosts. Furthermore, cyber space is a dynamic environment thus studying previously occurred attacks, both successful and unsuccessful (near misses), could lead to the development of a better defending strategy and the

early detection or even prevention of future attacks.

Moreover, IDSs are becoming an integral part of digital forensics and incident response processes. This is done through the correlation of the evidence by feeding the IDS outputs to the digital investigation processes in order to construct the timeline of an attack and identify the perpetrators. As with all incident response cases, time is a critical factor. In most digital forensic investigations specific instances recovered from evidence such as potentially malicious domains, become inaccessible over time. Consequently, automating part of the correlations, which the analyst would normally perform manually, could lead to the retrieval of time-sensitive evidence. The sooner an attack is reconstructed and all its pieces identified, the sooner the targeted company or organization could react and take extra precautions. This may include revising or adding new safeguards, such as populating firewall rules or educating employees, with the aim of avoiding further damage caused from the same or a different attacker. Finally, correlating the produced indicators could reduce the false alarms produced by the IDS, as they will not fit in any malicious campaign and constitute outliers.

In a seminal paper, Hutchins *et al.* [99] introduced the attack kill chain consisting of seven phases, namely: reconnaissance, weaponization, delivery, exploitation, installation, command and control (C&C) and actions. These refer to the different stages of an APT attack vector and all together form a campaign. Previously produced IDS indicators can be used to reconstruct the occurred attacks. Also, parts of the IDS attack classes correspond to specific campaign stages. For instance, probe class corresponds to the reconnaissance stage. Other stages such as the installation can only be detected through host data. Hence, features extracted from the host machines could be crucial in some cases. Specifically, sudden changes in the CPU or the memory consumption, or modification of crucial registry keys are strong indicators of malware infection. Intensive parsing of different file types in many different folders in a very short time, is a solid indicator of ransomware activity. Open connections which belong to processes that normally would not communicate through the Internet or orphan processes (those that do not have a parent process), are other indicators of a malware infection.

Nevertheless, their collection should be done in a way that does not disrupt the normal operations and does not raise any privacy violations. Ideally the collection should be able to achieve both maximum privacy for the users and gather the required data to attain maximum visibility within the infrastructure. Unfortunately, this is not a feasible and realistic goal in all situations, therefore, a trade-off between privacy and security is inevitable [100]. The criticality of the infrastructure and the types of threats against it, are the main factors that will influence such a trade-off. Depending on its location, the organization or company has to conform to a different set of laws and regulations. For instance, in the EU complete transparency is required as demanded by the new EU General Data Protection Regulation[1] (EU GDPR). Each organization or company is compelled to explicitly declare the data that are collected and justify why they are of importance against specific threats. Finally, the users of the infrastructure should be fully aware of the data that are collected and stored for security purposes.

As Hutchins *et al.* [99] stated, defenders may take advantage of the "persistent nature" of the attackers and use it against them. By their nature, APT actors attempt multiple intrusions that may well be scattered through time and use the feedback (successful or unsuccessful) of these attempts to adjust their strategy. Even for a well-resourced attacker the idea of completely changing her modus operandi is not feasible or profitable. The authors also present a case study to prove that comparing different stages of multiple campaigns can determine if they came from the same attacker. Such complex correlations require the extension of the current detection methods to answer attribution questions like the following:

- *Is a reconstructed campaign part of a persistent threat?* Two or more attack campaigns that belong to the same attacker could be part of a larger targeted attack or an APT. In this case, the number of successful and unsuccessful attempts of the attacker and the timing of the attacks could be used for the correlation. An attacker may launch multiple campaigns over a certain period, but may also wait for days even for months before trying again to avoid detection. This decision depends on the importance of the aim, the incentives and the resources.
- *Given any two attacks on a set of resources, can we identify if they are connected?* To answer this question we could consider features that describe the similarities of two attacks in case of campaigns and the timestamps related to the attacks. The similarities could refer to the targeted host, the exploit in use, the delivery method, etc. The evident method of correlating two attacks would be the source IP address. However, a large percentage of attacks use nowadays services such as Tor, VPN and proxies, which make the identification of the real source IP particularly challenging. Moreover, IP correlation is not helpful in case of distributed attacks as the attack originates from multiple infected hosts, which are controlled by a botmaster. Besides, small or medium DDoS attacks are sometimes used as a smokescreen to cover smaller but more significant activities, such as data exfiltration. In this case, time-related features and the size of the attack could be used to identify this kind of relations between attacks.
- *Can we use additional information from external sources such as social media or Dark Web to identify the attacker?* Often attackers such as hacktivists tend to announce and discuss their attacks through social networks, *e.g.*, Twitter and Facebook [101], [102]. Similarly, threat intelligence can be gathered from the Dark web, where many threat actors use forums to communicate, exchange vulnerabilities or coordinate their attacks [103]. Moreover, in many cases researching Dark Web forums may lead to the identification of attacks that were previously missed by

[1] https://www.eugdpr.org/

the victim company or organisation. The correlation of such information with the detected attacks and constructed campaigns could possibly lead closer to the identification of the attacker.

- *Is an attack manual or automated?*

When considering the attribution of an attack, *i.e.* the identification of the attacker, we should consider if the attack or parts of it are automated. For instance, in most cases DDoS attacks are automated either through software or botnets. In this case the use of IP as a correlation criterion is not helpful. On the other hand, if an attack is manually accomplished, there are more possibilities of a direct connection back to its origin. However, even in this case the attribution may require multiple stages of correlation. For example, in the case of pivoting from one infected host to the final target, the connections detected on the final host may be misleading. In this case, correlating the similar behaviour of the two hosts and the network traffic between them could reveal the pivoting. Then, evidence from the initially infected host could be used for the identification of the attacker.

The evident problem that arises is the complexity and the diversity of the correlations to be performed in such a massive amount of heterogeneous and rapid data. Rules and predetermined signatures cannot be used, as they would identify only known patterns and thus limit the effectiveness of such a system. Conversely, a method that is able to identify the natural connections and correlations by using constructed sets of features is needed in intrusion detection. For these reasons, the digital forensics field could benefit from big data analytics and data mining techniques, such as feature-based attribution. Moreover, as the field of machine learning and data mining is evolving, new algorithms and optimizations are proposed, which can benefit the security domain in many ways. Several security related tasks, such as intrusion detection and attack reconstruction, can be automated or semi-automated. Also, the performance of such tasks in terms of time and resources can be significantly improved.

B. IDS attack classes

All IDS-related literature divides the indicators produced by the system in four classes, namely DoS, U2R, R2L and probe (see Section III). As attacks evolve these classes should have evolved along with them and, thus, we consider that they are not sufficient anymore. The open issues regarding the attack classes of intrusion detection systems that have been identified in this work are as follows:

1) An intrusion detection system produces attack indicators and divides them in the aforementioned 4 classes. According to Akamai [1] and OWASP TOP 10 [104], the three most frequent web application attacks are SQL injections, Local file inclusion and XSS. All three of them belong to the U2R and R2L classes, which are currently the least detected attack classes. This fact supports our previous argument that it is necessary to reinforce the IDS's ability to detect these classes by using feature selection, extraction and transformation techniques and manually selected features (see Section III.B).
2) As stated by Kaspersky [105], cyber espionage through data exfiltration is one of the worst fears of every business. In this attack, the transmission of the targeted data to the attacker takes place after host infection with malware and in most cases uses encrypted traffic. This malicious network activity does not fall under the traditional four attack classes that are used in the literature, as it is the action that follows a successful intrusion. Nevertheless, creating an indicator for this malicious network traffic could identify an intrusion and reveal the infected host, even if the infection was not detected.
3) Similarly as mentioned in [106], the number of new ransomware families has steadily increased since 2011 and in 2016 almost 43% of ransomware victims were employees in organizations. This attack falls under the malware infection category, which does not necessarily require network traffic (*e.g.*, infection with an USB drive). However, it possesses a unique characteristic: after the first execution the malware has to communicate with a C&C server to receive the encryption key. By selecting a proper feature subset we might create indicators for this type of communication (see Section III.B) and stop the realization of the attack, even if the infection was successful.
4) Botnets facilitate DDoS attacks, regardless if they are hired or owned by the attackers [5]. Thus, botnet infections have increased the last years and large networks with high speeds, such as corporate or university networks are the primary targets. Again, botnet infections and communication do not fall under any of the four known attack categories. Detecting the infection stage could be based mostly on features related to the host as in any other malware infection instance. On the contrary, detecting botnet communication can be based on network features regarding a) the type of communication: HTTP, IRC, P2P or a social network communication method [107] and b) the occurrence of hiding and evasion techniques, such as Fast-Flux [108]. Finally, produced indicators could be combined to identify infected hosts as members of a botnet through spatial and temporal correlation. For instance, if a group of hosts is observed to be performing similar activities in response to similar messages from the same server, then they are likely part of the same botnet.

As easily observed, points two, three and four are associated with the outgoing network traffic. Traditionally, computer security uses the notion of the perimeter, considers everything out of it enemy territory and everything inside as safe zone. Consequently, malicious traffic used to be considered as always part of the incoming traffic. This fact however is no longer valid, with the emergence of insider threats, data exfiltration and C&C communication. To conclude, the existing categorization of attacks needs to be extended.

Following the evolution of the attacks and the current threats, the outgoing traffic needs to be considered as well. Based on these, we propose the addition of the following attack classes: data exfiltration, botnet C&C communication and ransomware communication. Ransomware communication is referred to the acquisition of the encryption key but utilizes C&C methods for

it. Consequently, this class could be considered a subclass of C&C. However, ransomware network communication is more specific and its messages are limited in number whereas botnet C&C communication is extended and continuous through time. Moreover, if we consider host evidence as well, a ransomware malware reveals itself to the user whereas a botnet malware utilizes different techniques to keep itself hidden. Finally, botnet communication often includes multiple hosts exhibiting the same suspicious behaviour, which constitutes another unique characteristic.

V. Related Work

Bhuyan *et al.* [32] provide a comprehensive background and a thorough review of anomaly detection papers under the categories of statistical, classification-based, knowledge-based, soft computing, clustering-based and combination learners. Moreover, the authors present capturing methods, different metrics, attack types and a review of the most used datasets. Finally, they discuss open issues and provide practical recommendations for designing an IDS. In [109], Catania *et al.* compare known signature and anomaly based methods not only according to their detection rate, but also based on the automation level they can achieve without human interaction. Signature-based, anomaly-based and stateful protocol analysis methods are included in the review but also the authors attempt to discuss the gap between theoretical data mining methods for intrusion detection and their deployment in real networks [110]. Buczak *et al.* [111] compared representative machine learning (ML) and data mining (DM) methods, based on the number of citations and the relevancy. Their survey includes a comprehensive background of the DM/ML field and compares different methods, such as artificial neural networks, Bayesian network, clustering, decision trees, etc. The authors also present some known cyber-security datasets for ML and DT. Finally, they highlight some of the most significant problems in intrusion detection, such as the collection of labelled data for re-training.

Readers interested in cloud intrusion detection may refer to [112]. The authors present an exhaustive review of Virtual Machine Introspection (VMI) and Hypervisor Introspection (HVI) based techniques for intrusion detection along with their advantages and disadvantages. That includes misuses, anomaly based and hybrid methods. Furthermore, a threat model is proposed especially for highlighting vulnerabilities in cloud environments, which includes both attacks launched from within and outside the cloud. Attacks are classified target-component wise in five categories: attacks in virtual machines (VMA), attacks on virtual machine monitor (VMMA), attacks in hardware (HA), attacks on virtual storage (VSA) and attacks in tenant network (TNA).

Contrary to other surveys, Vasilomanolakis *et al.* [43] focus on Collaborative IDSs (CIDSs) and divide them in three categories based in their architecture: centralized, decentralized and distributed. After disassembling the CIDS into five basic building blocks, each block is discussed. For each one of the three categories, the most known CIDSs where compared in block level based on the capabilities they offer. Authors define the requirements for a CIDS and present relationship between them and the attacks against them. The paper concludes that no CIDS is able to provide the necessary capabilities in a large scale network.

In [113] authors present different intrusion detection approaches for Mobile Ad-Hoc Networks (MANET) and Wireless Sensor Networks (WSN) taking in consideration their architecture: distributed, centralized, hierarchical or standalone. Their survey included statistical, clustering, game theory and genetic algorithms. Likewise, Mitchell *et al.* in [114] classify existing anomaly, signature based and hybrid IDS techniques according to the type of system they will be deployed to. Their classification includes Wireless Local Area Networks (WLAN), Wireless Sensor Networks (WSN), Wireless Personal Area Networks (WPAN), Ad-hoc networks, mobile telephony, Cyber Physical Systems (CPS) and Wireless Mesh Networks (WMN). Luong *et al.* in [115] provide a comprehensive review of economic and pricing approaches for detecting security attacks against wireless networks, such as eavesdropping, jamming and black hole attacks. Furthermore, the paper concludes by providing research directions for both existing challenges from an attack and an economic tool perspective, as well as for new research problems regarding 5G HetNets. Similarly, in [117] a review of machine learning methods used to address a variety of problems in WSNs is presented. The authors do not focus only on the security aspect but also on other functional and non-functional characteristics of a wireless network. Xu *et al.* in [116] discuss a major threat against WSNs, node forgery or impersonation, and propose device fingerprinting solutions against it. The authors review fingerprinting methods for WSN security and propose their usage for generating non-forgeable signatures in order to distinguish malicious users from legitimate ones. Zhu *et al.* [118] provided a taxonomy of SCADA-specific intrusion detection systems and tried to highlight the requirements of an IDS designed to protect a control system. Except from a review on IDS techniques, the authors define a set of metrics and attacks specifically for SCADA systems.

VI. Conclusion

The topic of intrusion detection has been researched extensively over the last two decades. Through the years different methods, signature and anomaly based, have been deployed. This paper surveys anomaly based IDSs of the last few years that use unsupervised techniques, as they exhibit the ability to detect unknown attacks, based on the features that describe each attack class without any prior knowledge or the need of labelled data for regular training. Moreover, anomaly based hybrid IDSs have been reviewed, as they aim to combine the advantages of both unsupervised and supervised methods.

Nowadays forensic investigations include a significant amount of data from diverse sources, such as network traffic, host machines, logs from different devices, etc. Unsupervised techniques are more malleable to the addition of new features extracted from different evidence sources and do not require regular re-training. For this reason, we present and compare feature selection methods for intrusion detection. Through our survey we highlight that finding and using an optimal subset of features for each class decreases the computational time and

complexity, boosts the detection accuracy and decreases the false positive rate. This concept is well known in the data analytics field, but it has not yet been extensively used in intrusion detection.

Furthermore, we identify the limitations of the current IDSs and discuss directions for future work to effectively cope with the current threat landscape. For this reason, we argue that the gap between the IDS and the forensic instigation needs to be filled. IDS outputs can be introduced to the forensic investigation processes in order to construct the timeline of an attack and correlate the reconstructed attacks aiming to identify the perpetrator. Finally, we propose the extension of the existing attack classes by adding three new categories regarding the outgoing network communication.

ACKNOWLEDGMENT

The authors wish to thank Evangelos Karagiannis for his valuable contribution.

REFERENCES

[1] Akamai's State of the Internet, Q1, Report (2017). https://www.akamai.com/StateOfTheInternet

[2] Kaspersky Security Bulletin (2015). https://securelist.com/analysis/kaspersky-security-bulletin/73038/kaspersky-security-bulletin-2015-overall-statistics-for-2015/

[3] Caballero, J., Grier, C., Kreibich, C., & Paxson, V. (2011, August). Measuring Pay-per-Install: The Commoditization of Malware Distribution. In *Usenix security symposium* (p. 15).

[4] Sood, A. K., Bansal, R., & Enbody, R. J. (2013). Cybercrime: Dissecting the state of underground enterprise. *IEEE internet computing*, *17*(1), 60-68.

[5] Symantec Internet Security Threat Report, VOLUME 21, APRIL 2016 : https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf

[6] DETECTION, I. (2002). Intrusion detection: a brief history and overview.

[7] Lazarevic, A., Kumar, V., & Srivastava, J. (2005). Intrusion detection: A survey. In *Managing Cyber Threats* (pp. 19-78). Springer US.

[8] Yeung, D. Y., & Ding, Y. (2003). Host-based intrusion detection using dynamic and static behavioral models. *Pattern recognition*, *36*(1), 229-243.

[9] Smaha, S. E. (1988, December). Haystack: An intrusion detection system. In *Aerospace Computer Security Applications Conference, 1988., Fourth* (pp. 37-44). IEEE.

[10] Lichodzijewski, P., Zincir-Heywood, A. N., & Heywood, M. I. (2002, May). Host-based intrusion detection using self-organizing maps. In *IEEE international joint conference on neural networks* (pp. 1714-1719).

[11] Hu, J., Yu, X., Qiu, D., & Chen, H. H. (2009). A simple and efficient hidden Markov model scheme for host-based anomaly intrusion detection. *IEEE network*, *23*(1), 42-47.

[12] Creech, G., & Hu, J. (2014). A semantic approach to host-based intrusion detection systems using contiguous and discontiguous system call patterns. *IEEE Transactions on Computers*, *63*(4), 807-819

[13] Hoglund, A. J., Hatonen, K., & Sorvari, A. S. (2000). A computer host-based user anomaly detection system using the self-organizing map. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on* (Vol. 5, pp. 411-416). IEEE.

[14] Debar, H., Dacier, M., & Wespi, A. (1999). Towards a taxonomy of intrusion-detection systems. *Computer Networks*, *31*(8), 805-822.

[15] Biddle, P., England, P., Peinado, M., & Willman, B. (2002, November). The darknet and the future of content distribution. In *ACM Workshop on Digital Rights Management* (Vol. 6, p. 54).

[16] Sperotto, A., Schaffrath, G., Sadre, R., Morariu, C., Pras, A., & Stiller, B. (2010). An overview of IP flow-based intrusion detection. *IEEE communications surveys & tutorials*, *12*(3), 343-356.

[17] Lee, W., & Stolfo, S. J. (1998, January). Data Mining Approaches for Intr Roesch, M. (1999, November). Snort: Lightweight Intrusion Detection for Networks. In LISA (Vol. 99, No. 1, pp. 229-238).

[18] Paxson, V. (1999). Bro: a system for detecting network intruders in real-time. Computer networks, 31(23), 2435-2463.usion Detection. In *Usenix security*.

[19] Suricata-IDS: https://suricata-ids.org/

[20] AlienVault IDS Unified Security Management (USM) : www.alienvault.com/ids

[21] CISCO Firepower Next-Generation IPS (NGIPS): https://www.cisco.com/c/en_uk/products/security/ngips/index.html

[22] FireEye Security Solutions: https://www.fireeye.com/products/nx-network-security-products.html

[23] Dreger, H., Feldmann, A., Paxson, V., & Sommer, R. (2004, October). Operational experiences with high-volume network intrusion detection. In *Proceedings of the 11th ACM conference on Computer and communications security* (pp. 2-11). ACM.

[24] Cheng, T. H., Lin, Y. D., Lai, Y. C., & Lin, P. C. (2012). Evasion techniques: Sneaking through your intrusion detection/prevention systems. *IEEE Communications Surveys & Tutorials*, *14*(4), 1011-1020.

[25] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, *41*(3), 15.

[26] Barbara, D., Couto, J., Jajodia, S., Popyack, L., & Wu, N. (2001). ADAM: Detecting intrusions by data mining. In *In Proceedings of the IEEE Workshop on Information Assurance and Security*.

[27] Kim, Gisung, Seungmin Lee, and Sehun Kim. "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection." *Expert Systems with Applications* 41.4 (2014): 1690-1700.

[28] Depren, Ozgur, et al. "An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks." *Expert systems with Applications* 29.4 (2005): 713-722.

[29] Zhang, Jiong, and Mohammad Zulkernine. "A hybrid network intrusion detection technique using random forests." *First International Conference on Availability, Reliability and Security (ARES'06)*. IEEE, 2006.

[30] Hwang, K., Cai, M., Chen, Y., & Qin, M. (2007). Hybrid intrusion detection with weighted signature generation over anomalous internet episodes. *IEEE Transactions on Dependable and Secure Computing*, *4*(1), 41-55.

[31] Anderson, D., Frivold, T., & Valdes, A. (1995). Next-generation intrusion detection expert system (NIDES): A summary.

[32] Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2014). Network anomaly detection: methods, systems and tools. *IEEE Communications Surveys & Tutorials*, *16*(1), 303-336.

[33] Catania, C. A., & Garino, C. G. (2012). Automatic network intrusion detection: Current techniques and open issues. *Computers & Electrical Engineering*, *38*(5), 1062-1072.

[34] Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security*, *28*(1), 18-28.

[35] Laskov, P., Düssel, P., Schäfer, C., & Rieck, K. (2005, September). Learning intrusion detection: supervised or unsupervised?. In *International Conference on Image Analysis and Processing* (pp. 50-57). Springer Berlin Heidelberg.

[36] Pattewar, T. M., & Sonawane, H. A. (2015, November). Neural network based intrusion detection using Bayesian with PCA and KPCA feature extraction. In *2015 IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS)* (pp. 83-88). IEEE.

[37] Qadeer, M. A., Iqbal, A., Zahid, M., & Siddiqui, M. R. (2010, February). Network traffic analysis and intrusion detection using packet sniffer. In *Communication Software and Networks, 2010. ICCSN'10. Second International Conference on* (pp. 313-317). IEEE.

[38] Kang, I., Jeong, M. K., & Kong, D. (2012). A differentiated one-class classification method with applications to intrusion detection. *Expert Systems with Applications*, *39*(4), 3899-3905.

[39] Kuang, F., Xu, W., & Zhang, S. (2014). A novel hybrid KPCA and SVM with GA model for intrusion detection. *Applied Soft Computing*, *18*, 178-184.

[40] Pattewar, T. M., & Sonawane, H. A. (2015, November). Neural network based intrusion detection using Bayesian with PCA and KPCA feature extraction. In *2015 IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS)* (pp. 83-88). IEEE.

[41] Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on* neural networks, 16(3), 645-678.

[42] Portnoy, L., Eskin, E., & Stolfo, S. (2001). Intrusion detection with unlabeled data using clustering. In *In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001*.

[43] Vasilomanolakis, E., Karuppayah, S., Mühlhäuser, M., & Fischer, M. (2015). Taxonomy and survey of collaborative intrusion detection. *ACM Computing Surveys (CSUR)*, *47*(4), 55.

[44] Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation

[45] Fawcett, T. (2006). An introduction to ROC analysis. Pattern recognition letters, 27(8), 861-874.

[46] Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods.

[47] Roth, V., & Lange, T. (2003). Feature selection in clustering problems. In *Advances in neural information processing systems* (p. None).

[48] Fahad, A., Tari, Z., Khalil, I., Habib, I., & Alnuweiri, H. (2013). Toward an efficient and scalable feature selection approach for internet traffic classification. *Computer Networks*, *57*(9), 2040-2057.

[49] Fahad, A., Tari, Z., Khalil, I., Almalawi, A., & Zomaya, A. Y. (2014). An optimal and stable feature selection approach for traffic classification based on multi-criterion fusion. *Future Generation Computer Systems*, *36*, 156-169.

[50] Liu, Z., Wang, R., Tao, M., & Cai, X. (2015). A class-oriented feature selection approach for multi-class imbalanced network traffic datasets based on local and global metrics fusion. *Neurocomputing*, *168*, 365-381.

[51] Zhang, H., Lu, G., Qassrawi, M. T., Zhang, Y., & Yu, X. (2012). Feature selection for optimizing traffic classification. *Computer Communications*, *35*(12), 1457-1471.

[52] De la Hoz, E., de la Hoz, E., Ortiz, A., Ortega, J., & Martínez-Álvarez, A. (2014). Feature selection by multi-objective optimisation: Application to network anomaly detection by hierarchical self-organising maps. *Knowledge-Based Systems*, *71*, 322-338.

[53] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, *6(2)*, 182-197.

[54] Zhu, Y., Liang, J., Chen, J., & Ming, Z. (2017). An improved NSGA-III algorithm for feature selection used in intrusion detection. Knowledge-Based Systems, 116, 74-85.

[55] Li, Y., Wang, J. L., Tian, Z. H., Lu, T. B., & Young, C. (2009). Building lightweight intrusion detection system using wrapper-based feature selection mechanisms. *Computers & Security*, *28(6)*, 466-475.

[56] Amiri, F., Yousefi, M. R., Lucas, C., Shakery, A., & Yazdani, N. (2011). Mutual in-formation-based feature selection for intrusion detection systems. *Journal of Net-work and Computer Applications*, *34(4)*, 1184-1199.

[57] Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2016). A multi-step outlier-based anomaly detection approach to network-wide traffic. *Information Sciences*, *348*, 243-271.

[58] Casas, P., Mazel, J., & Owezarski, P. (2012). Knowledge-independent traffic monitoring: Unsupervised detection of network attacks. *IEEE Network*, *26(1)*, 13-21.

[59] Casas, P., Mazel, J., & Owezarski, P. (2012). Unsupervised network intrusion detection systems: Detecting the unknown without knowledge. *Computer Communications*, *35(7)*, 772-783.

[60] Amoli, P. V., Hamalainen, T., David, G., Zolotukhin, M., & Mirzamohammad, M. (2016). Unsupervised Network Intrusion Detection Systems for Zero-Day Fast-Spreading Attacks and Botnets. *JDCTA (International Journal of Digital Content Technology and its Applications*, *Volume 10 Issue 2*, 1-13.

[61] Bohara, A., Thakore, U., & Sanders, W. H. (2016, April). Intrusion detection in enterprise systems by combining and clustering diverse monitor data. *In Proceedings of the Symposium and Bootcamp on the Science of Security* (pp. 7-16). ACM.

[62] Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2011, February). NADO: network anomaly detection using outlier approach. *In Proceedings of the 2011 International Conference on Communication, Computing & Security* (pp. 531-536). ACM.

[63] Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2012, August). An effective unsupervised network anomaly detection method. *In Proceedings of the International Conference on Advances in Computing, Communications and Informatics* (pp. 533-539). ACM.

[64] Costa, K. A., Pereira, L. A., Nakamura, R. Y., Pereira, C. R., Papa, J. P., & Falcão, A. X. (2015). A nature-inspired approach to speed up optimum-path forest clustering and its application to intrusion detection in computer networks. *Information Sciences*, *294*, 95-108.

[65] Bostani, H., & Sheikhan, M. (2017). Modification of supervised OPF-based intrusion detection sys-tems using unsupervised learning and social network concept. Pattern Recognition, 62, 56-72.

[66] Lin, W. C., Ke, S. W., & Tsai, C. F. (2015). CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowledge-based systems*, *78*, 13-21.

[67] Hosseinpour, F., Amoli, P. V., Farahnakian, F., Plosila, J., & Hämäläinen, T. (2014). Artificial immune system based intrusion detection: Innate immunity using an unsupervised learning approach. *International Journal of Digital Content Technology and its Applications*, *8(5*), 1.

[68] Jha, M., & Acharya, R. (2016, September). An immune inspired unsupervised intrusion detection system for detection of novel attacks. In Intelligence and Security Informatics (ISI), 2016 IEEE Confer-ence on (pp. 292-297). IEEE.

[69] Elbasiony, R. M., Sallam, E. A., Eltobely, T. E., & Fahmy, M. M. (2013). A hybrid network intrusion detection framework based on random forests and weighted k-means. *Ain Shams Engineering Journal*, *4(4)*, 753-762.

[70] Aljarah, I., & Ludwig, S. A. (2013, June). MapReduce intrusion detection system based on a particle swarm optimization clustering algorithm. *In 2013 IEEE Congress on Evolutionary Computation* (pp. 955-962). IEEE.

[71] Song, J., Takakura, H., Okabe, Y., & Nakao, K. (2013). Toward a more practical unsupervised anomaly detection system. *Information Sciences*, *231*, 4-14.

[72] Ashfaq, R. A. R., Wang, X. Z., Huang, J. Z., Abbas, H., & He, Y. L. (2017). Fuzziness based semi-supervised learning approach for intrusion detection system. Information Sciences, 378, 484-497.

[73] Chandrasekhar, A. M., & Raghuveer, K. (2013, January). Intrusion detection technique by using k-means, fuzzy neural network and SVM classifiers. *In Computer Communication and Informatics (ICCCI), 2013 International Conference on* (pp. 1-7). IEEE.

[74] Gogoi, P., Bhattacharyya, D. K., Borah, B., & Kalita, J. K. (2014). MLH-IDS: a multi-level hybrid intrusion detection method. *The Computer Journal*, *57(4)*, 602-623.

[75] Om, H., & Kundu, A. (2012, March). A hybrid system for reducing the false alarm rate of anomaly intrusion detection system. *In Recent Advances in Information Technology (RAIT), 2012 1st International Conference on* (pp. 131-136). IEEE.

[76] Mingqiang, Z., Hui, H., & Qian, W. (2012, July). A graph-based clustering algorithm for anomaly intrusion detection. *In Computer Science & Education (ICCSE), 2012 7th International Conference on* (pp. 1311-1314). IEEE.

[77] Chen, M., Challita, U., Saad, W., Yin, C., & Debbah, M. (2017). Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks. arXiv preprint arXiv:1710.02913.

[78] Shiravi, A., Shiravi, H., Tavallaee, M., & Ghorbani, A. A. (2012). Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Computers & Security*, *31(3)*, 357-374.

[79] Saad, S., Traore, I., Ghorbani, A., Sayed, B., Zhao, D., Lu, W., ... & Hakimian, P. (2011, July). Detecting P2P botnets through network behavior analysis and machine learning. *In Privacy, Security and Trust (PST), 2011 Ninth Annual International Conference on* (pp. 174-180). IEEE.

[80] French Chapter of Honenynet : http://www.honeynet.org/chapters/france

[81] Szabó, G., Orincsay, D., Malomsoky, S., & Szabó, I. (2008, April). On the validation of traffic classification algorithms. *In International Conference on Passive and Active Network Measurement* (pp. 72-81). Springer Berlin Heidelberg.

[82] The CAIDA UCSD Anonymised Internet Traces 2016: http://www.caida.org/data/passive/passive_2016_dataset.xml

[83] Sony, C. S. L. (2000). Traffic data repository at the WIDE project. *In Proceedings of USENIX 2000 Annual Technical Conference: FREENIX Track* (pp. 263-270).

[84] LBNL Enterprise Trace Repository. : http://www.icir.org/enterprise-tracing.

[85] UNIBS Data Sharing: http://netweb.ing.unibs.it/~ntw/tools/traces/

[86] MIT LINCOLN Laboratory: DARPA Intrusion Detection Evaluation: https://www.ll.mit.edu/ideval/data/2000data.html

[87] KDDcup99, "Knowledge discovery in databases DARPA archive,"http://www.kdd.ics.uci.edu/databases/kddcup99/task.html, 1999

[88] ISCX NSL-KDD dataset: http://www.unb.ca/research/iscx/dataset/iscx-NSL-KDD-dataset.html

[89] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In *Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009*.

[90] METROSEC: http://projects.laas.fr/METROSEC/
[91] Cheung, S., Lindqvist, U. and Fong, M.W., 2003, April. Modeling multistep cyber attacks for scenario recognition. In *DARPA information survivability conference and exposition, 2003. Proceedings* (Vol. 1, pp. 284-292). IEEE.
[92] Limmer, T., & Dressler, F. (2008). Survey of Event Correlation Techniques for Attack Detection in Early Warning Systems. *University of Erlangen, Dept. of Computer Science, Technical Report/, April.*
[93] Noel, S., Robertson, E., & Jajodia, S. (2004, December). Correlating intrusion events and building attack scenarios through attack graph distances. In *Computer Security Applications Conference, 2004. 20th Annual* (pp. 350-359). IEEE.
[94] Dwivedi, N., & Tripathi, A. (2015, February). Event Correlation for Intrusion Detection Systems. In *Computational Intelligence & Communication Technology (CICT), 2015 IEEE International Conference on* (pp. 133-139). IEEE.
[95] Jiang, G., & Cybenko, G. (2004, June). Temporal and spatial distributed event correlation for network security. In *American Control Conference, 2004. Proceedings of the 2004* (Vol. 2, pp. 996-1001). IEEE.
[96] Morin, B., & Debar, H. (2003, September). Correlation of intrusion symptoms: an application of chronicles. In *International Workshop on Recent Advances in Intrusion Detection* (pp. 94-112). Springer Berlin Heidelberg.
[97] Krügel, C., Toth, T., & Kerer, C. (2001, December). Decentralized event correlation for intrusion detection. In *International Conference on Information Security and Cryptology* (pp. 114-131). Springer Berlin Heidelberg.
[98] Virvilis, N., & Gritzalis, D. (2013, September). The big four-what we did wrong in advanced persistent threat detection?. In *Availability, Reliability and Security (ARES), 2013 Eighth International Conference on* (pp. 248-254). IEEE.
[99] Hutchins, E. M., Cloppert, M. J., & Amin, R. M. (2011). Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. *Leading Issues in Information Warfare & Security Research*, *1*, 80.
[100] Mylonas, A., Meletiadis, V., Mitrou, L., & Gritzalis, D. (2013). Smartphone sensor data as digital evidence. Computers & Security, 38, 51-75.
[101] Operation Payback Twitter: https://twitter.com/payback_op?lang=en
[102] Anonymous Op Twitter: https://twitter.com/anon_operation?lang=en
[103] Macdonald, M., Frank, R., Mei, J., & Monk, B. (2015, August). Identifying digital threats in a hacker web forum. In Advances in Social Networks Analysis and Mining (ASONAM), 2015 IEEE/ACM International Conference on (pp. 926-933). IEEE.
[104] OWASP TOP 10 List: https://www.owasp.org/index.php/Top_10_2013-Top_10
[105] Kaspersky Lab: Damage Control: The Cost of Security Breaches, in Kaspersky Labs (IT Security Risks Special Report Series) (2015). https://media.kaspersky.com/pdf/it-risks-survey-report-cost-of-security-breaches.pdf
[106] Symantec, An ISTR Special Report: Ransomware and Businesses 2016 http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/ISTR2016_Ransomware_and_Businesses.pdf
[107] Thomas, K., & Nicol, D. M. (2010, October). The Koobface botnet and the rise of social malware. In *Malicious and Unwanted Software (MALWARE), 2010 5th International Conference on* (pp. 63-70). IEEE.
[108] Nazario, J., & Holz, T. (2008, October). As the net churns: Fast-flux botnet observations. In *Malicious and Unwanted Software, 2008. MALWARE 2008. 3rd International Conference on* (pp. 24-31). IEEE.
[109] Catania, C. A., & Garino, C. G. (2012). Automatic network intrusion detection: Current techniques and open issues. *Computers & Electrical Engineering*, *38(5)*, 1062-1072.
[110] Liao, H. J., Lin, C. H. R., Lin, Y. C., & Tung, K. Y. (2013). Intrusion detection sys-tem: A comprehensive review. *Journal of Network and Computer Applications*, *36(1)*, 16-24.
[111] Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 18(2), 1153-1176.
[112] Mishra, P., Pilli, E. S., Varadharajan, V., & Tupakula, U. (2016). Intrusion Detection Techniques in Cloud Environment: A Survey. *Journal of Network and Computer Applications.*
[113] Butun, I., Morgera, S. D., & Sankar, R. (2014). A survey of intrusion detection systems in wireless sensor networks. IEEE Communications Surveys & Tutorials, 16(1), 266-282.
[114] Mitchell, R., & Chen, R. (2014). A survey of intrusion detection in wireless network applications. *Computer Communications*, *42*, 1-23.
[115] Luong, N. C., Hoang, D. T., Wang, P., Niyato, D., & Han, Z. (2017). Applications of economic and pricing models for wireless network security: A survey. *IEEE Communications Surveys & Tutorials*, 19(4), 2735-2767.
[116] Xu, Q., Zheng, R., Saad, W., & Han, Z. (2016). Device fingerprinting in wireless networks: Challenges and opportunities. *IEEE Communications Surveys & Tutorials*, 18(1), 94-104.
[117] Alsheikh, M. A., Lin, S., Niyato, D., & Tan, H. P. (2014). Machine learning in wireless sensor networks: Algorithms, strategies, and applications. IEEE Communications Surveys & Tutorials, 16(4), 1996-2018.
[118] Zhu, B., & Sastry, S. (2010, April). SCADA-specific intrusion detection/prevention systems: a survey and taxonomy. In *Proceedings of the 1st Workshop on Secure Control Systems* (SCS).
[119] A. Moore, D. Zuev, M. Crogan, Discriminators for use in flow-based classification, Technical Report, University of Cambridge, Computer Laboratory.
[120] Moore, A., Hall, J., Kreibich, C., Harris, E., & Pratt, I. (2003, April). Architecture of a network monitor. In *Passive & Active Measurement Workshop* (Vol. 2003).

Antonia Nisioti received her Diploma and M.Eng. in Electrical and Computer Engineering from the Democritus University of Thrace in Greece. She is currently pursuing a PhD in Bournemouth University (BU), UK on the field of data driven intrusion detection and network forensics. Her research interests include memory, network and Android forensics, SIEM technology, intrusion detection and big data analytics.

Moreover, she has developed digital forensic scenarios and exercises for training purposes for the European Union Agency for Network and Information Security (ENISA) and participated in many cyber defence exercises. Antonia has also worked in the digital forensics industry, working as a lab analyst of the Hellenic Data Protection Authority. She is a Member of IEEE and ACM.

Alexios Mylonas is a Lecturer at Bournemouth University. He holds a PhD in Information and Communication Security and a BSc (Hons) in Computer Science from the Athens University of Economics and Business, as well as an MSc in Information Security from Royal Holloway, University of London. In the past, he was a Lecturer at Staffordshire University and before that he worked as a security consultant within VeriSign's PKI Trust Network.

Alexios is an expert in cybersecurity and his current research interests include cybersecurity, threat intelligence, and web security. Alexios has more than 20 publications, which are well referenced and appear in esteemed conference and journal publications. He has served as technical committee member of conferences and journals. He is a Member of IEEE and ACM.

Paul D. Yoo (M'11–SM'13) is currently with the Cranfield Defence and Security based at the United Kingdom's Ministry of Defence establishment on the Oxfordshire/Wiltshire borders. Prior to this, Dr Yoo held academic/research posts in Sydney, Bournemouth and the UAE. Dr Yoo serves as an Editor of IEEE Communications Letters and Journal of Big Data Research (Elsevier) and holds over 60 prestigious journal and conference publications in highly regarded IEEE/ACM journals. Dr Yoo is affiliated with University of Sydney and Korea Advanced Institute of Science and Technology (KAIST) as a Visiting Professor. He is a Senior Member of IEEE and a Member of BCS. His research addresses two broad topics: advanced data analytics (inc. machine learning) and cyber physical systems security.

Vasilis Katos obtained a Diploma in Electrical Engineering from Democritus University of Thrace in Greece, an MBA from Keele University in the UK and a PhD in Computer Science (network security and cryptography) from Aston University. He is a certified Computer Hacking Forensic Investigator (CHFI) and have worked in the Industry as Information Security Consultant. He has also served as an expert witness in Information Security for a criminal court in the UK and a misdemeanor court in Greece.

Vasilis' research falls in the area of digital forensics and incident response. He has participated in 2 FP7 and 3 nationally funded research projects and in a number of national and international cyberdefence exercises. He has over 80 publications in journals, book chapters and conference proceedings and serves as a referee on several reputable conferences and journals (for example, IEEE Communications Letters, Computers & Security, Information and Computer Security), has coordinated and delivered a number of workshops, both in an academic and a security professionals context.

TABLE I
FEATURE SELECTION METHODS FOR INTRUSION DETECTION

Author	Method	FS Algorithm	Datasets	Type	Evaluation Criterion
[57]	Mutual information for feature-class relevancy and generalized entropy for feature-feature non-redundancy	Mutual Information and Generalized Entropy FS (MIGE-FS)	KDD99, NSL-KDD, TUIDS, UCI	Filter	Mutual information, generalized entropy
[50]	Two phases: 1. evaluation of the local and global score of its feature and search for a relevant and discriminative subset, 2. removal of redundant features from all subsets	Class-Oriented Feature Selection (COFS)	Cambridge, UNIBS, SCUT	Filter	Maximum entropy
[52]	1. Select a feature subset 2. Train classifier 3. Classification on dataset 4. Evaluation with Jaccard coefficient 5. Evolution of the population	NSGA-II and classifier is GHSOM with probabilistic relabeling	NSL-KDD	Wrapper	Jaccard coefficient
[49]	Three stages: 1. combines multiple FS techniques to find the optimal subset 2. Adaptive threshold based on maximum entropy 3. Random Forest filtering	Global Optimization Algorithm(GOA)	Cambridge Lab [119], Dataset [119]	Filter	Stability, Optimality
[48]	1. Extract an optimal FS subset for each of the 5 FS techniques 2.Support is calculated for each feature in the optimal subsets 3. If support is higher than the threshold the feature is going to the final subset	Local Optimization Approach (LOA)	KDD99, MAWI, Dataset [119]	Filter	Goodness Rate, Stability, Similarity
[51]	1. Filtering with WSU metric 2.Select optimal features with wrapper based on AUC 3. Choose the robust features	Weighed Symmetrical Uncertainty_Area Under Roc(WSU_AUC), Selection Robust Stable Features(SRSF)	UNIBS, Cambridge (non ids), CAIDA	Wrapper	Weighted Symmetrical uncertainty
[56]	1. Select 1st feature the one with maximum MI to the output 2.Greedy selection: compute feature to feature MI and select the optimal 3. Repeat until the desired number of features	Modified mutual information-based feature selection algorithm (MMIFS)	KDD99	Filter	Mutual Information
[55]	1. Initial subset generation 2.Iterative procedure: generate subset with modified RMHC and compare with the previous one 3. Find optimal subset after maximum iterations or predefined criterion is satisfied	Modified Random Mutation Hill Climbing (RMHC)+multiple linear SVMs	KDD99	Wrapper	N/A
[54]	1. Initialization of the population 2. Training of the classifier 3. Classification 4. Evaluation of the results based on the Jaccard coefficient 5. If not converge, evolution of the population based on a) special domination method (bias-selection) or b) predefined multiple targeted search (fit selection) 6. Repeat steps 2-4 until converge.	I-NSGA-III+GHSOM	KDD99	Wrapper	Jaccard coefficient

TABLE II
UNSUPERVISED AND HYBRID IDS

Authors	Method	Algorithm	Probe	DoS	R2L	U2R	Datasets	Input
[61]	1. Feature selection for redundant info using Pearson correlation 2. Clustering on the host and network data determine attacks through cluster normalcy	K-means, DBSCAN	N/A	N/A	N/A	N/A	VAST 2011 Mini Challenge 2	z
[58]	1. Mutual information and generalized entropy based feature selection technique 2. Tree based sub-clustering 3. Outlier detection using ROS' score and a predefined threshold	Tree-based subspace clustering (TCLUS)	98.07	99.99	89.96	76.32	KDD99, NSL-KDD, TUIDS	y
[60]	Two engines: 1. Clustering and outlier ranking using a Dynamic Self-Adaptable Threshold 2. Botnet detection	DBSCAN	FP: 3.61% TN:96.39% ACC:98.39% RECALL:100% PRECISION:98.12%				DARPA, ISCX	z
[64]	Use nature inspired optimization techniques to set k parameter for Optimum Path Forest clustering of the data	Optimum Path Forest Clustering (OPF)	PURITY MEASURE: ISCX: 0.9637, KDD99:0.7166,NSL-KDD: 0.9988				ISCX, KDD99, NSL-KDD	z
[65]	Three modules: 1. Partitioning: uses k-means to create training subsets for detection module 2. Pruning: aims to prune the training subsets to speed up MOPF 3. Detection: uses MOPF to detect attacks	K-means, Modified Optimum Path Forest (MOPF)	85.92	96.89	77.98	81.13	NSL-KDD	z
[66]	1. Extraction of cluster centers and nearest neighbours 2. Calculation of dist1 and dist2 3. Sum dist1 and dist2 to create a new distance feature for each point in the dataset 3. k-NN classifier is used for the new dataset	Clustering, KNN	6 f: 99.99 ,19 f:87.61	6 f: 99.99,19 f: 99.68	6 f:0 ,19 f:57.02	6 f: 0,19 f:3.85	KDD99	z
[67]	1. Primary innate immunity: clustering into self and non-self 2. secondary adaptive immunity: from the results of the clustering detectors are generated and when they are mature they will be distributed to the hosts	DBSCAN	FPR:0.008, TNR:0.991, ACC:0.771, RECALL:0.589, PRECISION:0.987, F-1:0.738				KDD99	z
[68]	Two layered Immune system Inspired IDS (I3DS): T-cells layer: a Hidden Markov Model (HMM) is used to identify possible attacks, B-cells layer: a decision tree is used to confirm true attacks	Hidden Markov Model (HMM), Decision tree	Detection rate: 60.2 ,F-measure: 64.5 , I-measure: 55.4, Precision: 77.8				KDD99	z
[69]	Online module: Misuse signature comparison through Random forest. If no match the offline module is called for clustering and creation of new signatures	Random forest and weighted K-means	maximum detection rate: 98.3% with FPR 1.6%				KDD99	y
[70]	MapReduce is used to parallelize the Particle Swarm Optimization that clusters the data based on the global optimal centroids	PSO clustering	maximum AUC: 0.963				KDD99	x

[71]	Training: 1. Filtering to find normal 2. Clustering 3. Create one SVM model for each cluster Testing: 1. Compare traffic to SVM models 2. If no match the flow is considered anomalous	Clustering and one-class SVM	N/A	N/A	N/A	N/A	KDD99, Kyoto university	x
[72]	A divide-and-conquer model uses the magnitude of fuzziness to categorize unlabelled data. Then, a neural network with random weights (NNRw) model is used to identify the attacks.	Neural network with random weights (NNRw)	Accuracy ($KDDTest^{+}$): 84% Accuracy ($KDDTest^{-}$): 68%				KDD99	z
[73]	1. Clustering creates k clusters 2. One neuro-fuzzy model for each cluster 3. Previous step results to a SVM vector 4. Radial SVM classification for the detection	k-means, SVM and fuzzy NN	97.31	98.8	97.5	97.5	KDD99	z
[74]	1.Supervised Classifier detects DoS and Probe 2.Unsupervised Classifier detects Normal 3.Outlier based detection for R2L and U2R	CatSub+, K-point and GBBK	98.75	99.99	91.1	81.4	TUIDS, KDD99, NSL-KDD	y
[58]	1. Multi-resolution traffic flow 2. Time series criterion for detecting a potentially malicious flow 3. Sub-Space Clustering (SSC) 4. Evidence Accumulation Clustering (EAC) ranking	DBSCAN	N/A	N/A	N/A	N/A	KDD99, MAWI, METROSEC	z
[75]	Hybrid, 3 modules, module 1:entropy based feature selection, module 2:clustering(normal/attack), module 3:classification(types of attack)	K-Means, K-NN, NAÏVE BAYES	98.43	95.15	97.6	92	KDD99	y
[59]	Sub-Space Clustering and Evidence Accumulation: partition the feature space in N sub-spaces and perform clustering in a lower dimension space	DBSCAN	0.95	0.95	0.8-0.85	0.8-0.85	METROSEC	z
[63]	1. Unsupervised cluster formation 2. Stability analysis 3. Iteration until clusters are stable 4.CLUSLab: cluster labelling technique	Tree-based subspace clustering(TCLUS)	0.9645	0.9997	0.8652	0.6623	KDD99, TUIDS	y
[76]	1. Graph-based algorithm 2. Outlier detection based on the local deviation coefficient	LDCGB	N/A	N/A	N/A	N/A	KDD99	z

x- represents continuous *y*- represents mixed *z*- represents N/A

TABLE III
DATASETS FOR INTRUSION DETECTION

Citation	Year	Type	Attack	Publicly Available	Description	# of Features
UNB ISCX [77]	2012	RL	ALL	Y	Real packet traces were analysed to create profiles for agents that generate real traffic	19
ISOT Botnet[79]	2010	BM	ALL	Y	Combinations of several existing publicly	
CAIDA [82]	2008-2016	BM	ALL	Y	passive backbone anonymised traffic, no payload	N/A
MAWI [83]	2006-2016	RL	N/A	Y	daily trace at the transit link of WIDE to the upstream ISP, some longer traces for some years	N/A
LBNL [84]	2005	BM	DoS	Y	more than 100 hours of activity from a total of several thousand internal hosts, heavily anonymised, no payload	N/A
UNIBS [85]	2009	RL	ALL	Y	TCP (99%) and UDP traffic, 79000 flows, description table at the url	N/A
DARPA [86]	2000	BM	DoS	Y	two scenarios, LLDOS 1.0 and LLDOS 2.0.2	N/A
KDD99 [87]	1999	BM	ALL	Y	most widely used dataset	41
NSL-KDD [88]	1999	BM	ALL	Y	better version of KDD99 (url for reasons)	41
TUIBS	N/A	RL	ALL	N/A	N/A	50,24
METROSEC [90]	N/A	RL	N/A	N	N/A	N/A
DEFCON	N/A	BM	DoS	N/A	CTF traffic	N/A
RL- represents real-life BM- represents benchmark						