TU Delft

Delft University of Technology

Document Version
Accepted author manuscript

Citation (APA)
Rimmer, V., Nadeem, A., Verwer, S., Preuveneers, D., & Joosen, W. (2022). Open-World Network Intrusion Detection. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (pp. 254-283). (Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Vol. 13049 LNCS). Springer. https://doi.org/10.1007/978-3-030-98795-4_11

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
In case the licence states “Dutch Copyright Act (Article 25fa)”, this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.

Open-World Network Intrusion Detection

Vera Rimmer[1](✉), Azqa Nadeem[2], Sicco Verwer[2], Davy Preuveneers[1], and Wouter Joosen[1]

[1] imec-DistriNet, KU Leuven, Leuven, Belgium
{vera.rimmer,davy.preuveneers,wouter.joosen}@cs.kuleuven.be
[2] Delft University of Technology, Delft, The Netherlands
{azqa.nadeem,s.e.verwer}@tudelft.nl

Abstract

This chapter contributes to the ongoing discussion on the integration of AI and security in the scope of intrusion detection. The focus is set on detection of network attacks through traffic analysis. We provide an outline of machine learning empowered solutions with a special emphasis on open-world detection. By analyzing research trends and key challenges of integrating machine learning techniques in open dynamic network environments, we reflect on the prospects for a reliable, practical, and secure adoption of open-world network intrusion detection.

1 Introduction

Intrusion detection is an integral part of securing information systems. Detection tools stem from the early realization of the computer security community that full and provable protection of an ICT infrastructure is practically infeasible, if not impossible. Attempts to compromise the system can emerge from within the infrastructure as well as from its adversarial environment. An intrusion detection system (IDS) therefore aims at detecting exploitation attempts and active misuse by internal and external attackers.

Continuous monitoring of the system and accurate detection of malicious behavior constitute the first step of the incident response process. For this fundamental step, an IDS performs acquisition of relevant data streams that describe operation of the system and its internal and external communication. Through an in-depth real-time analysis of these monitored data, an IDS searches for any signs of a potential misuse of the system. If such evidence is detected with high enough confidence, an alert is raised which is then propagated to security analysts for further investigation.

In essence, an IDS is responsible for intelligent automated decision making that can, depending on the correctness of those decisions, either safeguard or disrupt normal operation of the whole system. Reliable intrusion detection is indispensable, but despite the world-wide efforts of the last 40 years, IDSs regularly generate false alarms, at times fail to prevent intrusions, and thus end up jeopardizing system and data security with extremely high recovery costs. A vast amount of research and development has gone into creating dedicated tools and algorithms in order to bring intelligent and reliable intrusion detection into reality. Among the possible solutions, artificial intelligence (AI)

has always been a compelling component for automated knowledge retrieval that aids in efficient detection of ever-changing attacks, with a varying level of complexity and involvement. Especially in light of recent advances in the machine learning and deep learning domains, the solution space is evolving so rapidly that it has become challenging to keep track of major changes.

With this chapter, we aim to revisit the foundations of machine learning based network intrusion detection and, in light of recent advances in the field, discuss the intrinsic factors that continue to pose a challenge for research. First, in Section 2, we introduce the domain and its core concepts. After reviewing the problem statement and the threat model of a network-based IDS, we move on to outlining the principal machine learning techniques and their underlying assumptions in Section 3. Last but not least, in Section 4, we analyze open challenges, specifically in relation to the usage of machine learning for open-world detection on network traffic. As these challenges were revealed through a long and potent line of research, an important question remains as to how adequately or completely they have been addressed thus far.

2 Network Intrusion Detection

Malicious attempts to invade an ICT infrastructure must be detected and localized while in progress. This objective implies performing real-time analysis of continuous streams of data from various sources and locations in search for indicators of compromise. Traditionally, the process of intrusion detection has been split between network-based (NIDS) and host-based (HIDS) systems, that serve complementary purposes by monitoring malicious activities at different levels. HIDS runs on internal nodes, carrying the ability to closely monitor their individual behavior. It primarily relies on host-specific data sources, such as system and kernel-level activity traces, application logs, files and documents. Maintaining a fine-grained access to individual hosts' activity enables HIDS to precisely localize misuse, specifically active malware.

Naturally, however, the holy grail of intrusion detection is to recognize threats as early as possible before the system gets compromised. Since the most common way for intruders to enter an infrastructure is through the network, a NIDS – being placed at the edge of the network – analyses inbound and outbound network traffic and thus acts as the first line of defense. As opposed to the host-based level of monitoring, a network-based approach gives a more expansive view over the network, allowing for an early detection of attacks that target multiple hosts at once. Today, NIDSs can be also freely deployed within the perimeter, enabling them to detect internal network attacks.

AI algorithms are ubiquitously used across all possible data sources and points of IDS deployment. For the purpose of a more contained discussion, this chapter focuses on one particularly prominent and highly representative application of ML in network security – open-world network intrusion detection. Specifically, we discuss ML-based NIDS research and explore network traffic as the primary data source for detection. NIDSs can detect attacks either through *misuse detection*, i.e. matching observed traffic to a known malicious traffic signature, or through *anomaly detection*, i.e. comparing traffic to a previously established benign baseline. A ML-enabled NIDS employs a wide

range of learning algorithms for both approaches, and in this chapter we highlight traffic anomaly detection with ML as a particularly challenging but promising line of research.

2.1 Network threats

Before diving into particular detection algorithms, it is important to establish a threat model under which a NIDS operates. Cyberthreats targeted by a NIDS are either executed over a network, or communicate with external parties over a network. A NIDS mirrors the entire network traffic that is permitted by a firewall, being able to distinguish attacks across the OSI stack, from Layer 2 (Data Link) potentially all the way up to Layer 7 (Application). Very broadly, these threats can be categorized into passive and active attacks.

Passive attacks do not involve any meaningful interaction with nodes in a target network and do not alter any data. Instead, the purpose is to *probe* the system – obtain useful information which can be efficiently collected through, for instance, *network scanning* or *port scanning*. Network scanning allows to detect accessible nodes, while probing open ports allows to identify services running on these nodes, exposing their vulnerabilities. As such, a passive attack does not leave any traces in the system and is by itself a preparatory step before a more aggressive intervention. Detecting a passive attack in real-time gives a defender an opportunity to proactively identify an adversary and prevent their intrusion.

The types of *active attacks*, however, vary greatly. These are the network attacks that aim at compromising integrity, confidentiality and availability of target systems. The most basic way to *penetrate* the network would be by brute forcing credentials of a legitimate user, which is characterized by an overwhelming series of unsuccessful logins rather noticeable in the inbound network traffic. More advanced attackers penetrate a protected network by carefully exploiting vulnerabilities found in its perimeter and thus gaining unauthorized, potentially privileged access to the system in a more stealthy manner. These can be misconfigurations or vulnerabilities in firmware of entry-level network devices, or web and software vulnerabilities of publicly accessible hosts, such as buffer overflows, cross-site scripting (XSS) and SQL injections. Similar to brute force attacks, these exploits can also be launched over the network, carried in incoming network packets' payloads. However, due to traffic encryption and a number of other considerations to be discussed further, malicious payloads even of well-known exploits are not guaranteed to be detected.

Apart from penetration attacks, another large family of network threats is *denial of service* (DoS) attacks, which aim at disrupting normal functioning of target hosts and deny their availability to legitimate users. This goal can be achieved through flooding the victim node or resource with superfluous requests in an attempt to overload both the network bandwidth and the system, possibly also targeting its IDS. There are many variations of a DoS attack, including SYN flooding, ICMP flooding, smurf and others, which differ in mechanics and final effects. A distributed DoS attack (DDoS) is launched from numerous sources at once, often automated by a whole network of compromised computers – bots. Today, *botnets* are seen as the largest network security threat and remain one of the key research topics in intrusion detection. Bots become disguised as

legitimate actors through infecting privately owned systems. Their automated illegitimate activities can have various malicious impact, ranging from spam and click fraud campaigns to identity theft, DoS attacks and malware infections. Remote command-and-control (C&C) servers send instructions to the compromised cluster of computers and receive back reports and leaked information. Even though botnets generate a lot of communication, detecting them through traffic analysis or other means is a serious challenge for any type of IDS, as bots continue to evolve and find better disguises.

While botnets are rather stealthy and can cause extremely damaging consequences, there is an attack class that surpasses others in evasiveness, sophistication and severity – *advanced persistent threats* (APTs). These threats are human-driven attacks targeted against a specific infrastructure and aimed at gaining an ongoing access for a long period of time in order to exfiltrate sensitive valuable data – this could be, for instance, intellectual property of organizations, trade secrets, or customer information. It is especially difficult to detect an APT at the moment of perimeter penetration or privilege escalation, since advanced attackers may use unknown exploits or social engineering tricks to infiltrate an infrastructure. Afterwards, an APT only infects a few chosen hosts to get closer to valuable resources of the network, effectively staying under the radar. They rarely contact remote C&C servers, and when they do, they use encryption or obfuscation techniques, complicating traffic analysis by a NIDS. However, the *data exfiltration* process – an inevitable goal of an APT – may be well observable in outbound network traffic, which grants a crucial defensive role to a NIDS.

2.2 Network traffic monitoring

Network traffic became a universal data source for intrusion detection thanks to standardization and ubiquity of network protocols, which makes NIDSs adaptable to a wide range of platforms and applications. A NIDS collects and inspects network traffic in different modes, mostly either on a packet level or on a flow level. These approaches to traffic monitoring differ in informational content and practicality, and selecting one, or a combination, depends on the environment and the threat model for a NIDS.

Packet-level inspection. Capturing traffic from the network by a NIDS for monitoring and analysis purposes is achieved with promiscuous access to *copies* of network packets, and therefore without interference in communication. Full packet captures are usually made in the *pcap* file format, a widely used and portable format for packet inspection. Pcap files can be processed with Deep Packet Inspection (DPI), which performs both packet header and payload analysis. DPI can provide extensive information about communication, exposing malicious payloads to a NIDS. One of the earliest IDS tools that performs application-level DPI is Snort [1], an open-source signature-based detector. Snort matches observed packets with known malicious patterns using regular expressions (e.g., for a linux web server, a pattern could be an HTTP request containing 'etc/passwd'). A more recent open-source NIDS called Suricata [2] improves scalability of Snort [3]. Such signature-based NIDS strongly rely on a rich database of malicious payloads. For anomaly-based NIDS, one prominent example is the Bro tool [4]. Bro constructs benign baseline profiles for an application's usage based on predefined policy scripts and flags deviations from these profiles. For instance, it was shown to effectively detect web attacks, such as reflected XSS injection and SQL injection, by inspecting

strings in the HTTP-request parameters [5]. With access to payloads, these attacks are straightforwardly detected due to presence of unusual characters in the request body.

Despite all these advantages, processing full packet captures comes with some considerable practical issues. The sheer volume of packets in modern high speed communication networks is overwhelming, making DPI inefficient or even infeasible in real-time. Moreover, storing full captures for further network forensics is a highly limited resource. Captures of very large packets are often incomplete or even limited to header information, largely omitting the most informative parts. And finally, two fundamental limitations to performing DPI are (i) invasion of privacy through accessing and storing benign packets' content, and (ii) traffic encryption. Packet-level inspection on TLS-encrypted traffic can be realized through man-in-the-middle solutions that decrypt and re-encrypt payloads, thereby violating end-to-end security guarantees, which can be both unsafe and computationally intensive for a particular environment. An ongoing line of research explores DPI over encrypted traffic through matching encrypted tokens with encrypted rules [6,7]. Currently this approach requires computationally intensive setup phases for every network connection, and, without the aid of decryption, supports only a limited number of IDS rules.

In view of the above circumstances – high data rates, computationally demanding processing, privacy and encryption concerns – DPI for intrusion detection is arguably becoming increasingly obsolete in modern environments. Nevertheless, there exist numerous network intrusion detection datasets with full packet captures, as packet-level analysis has proven to be highly beneficial for research purposes.

Flow-level inspection. Rather than inspecting and storing all individual incoming and outgoing packets, a NIDS may group relevant packets together in a *flow* and collect their aggregated information on a flow level. Traffic flows are commonly defined as bidirectional packet streams between two hosts that correspond to one complete network connection. Namely, each flow is constructed as a series of packets collected over a predefined period of time (normally over one network connection) that share a five-tuple: source and destination IP addresses, source and destination ports, and a protocol. Flow-level information is aggregated across all the packets belonging to one flow – typically this includes packet header counts, arrival times, and counts of certain header attributes. These traffic flow meta-data provide a high-level description of communication between source and destination hosts which can be very telling about its benign or malicious nature [8]. As a result, flow-based inspection does not take into account traffic content, but instead reveals informative high-level communication patterns, while greatly reducing the size of data to be analyzed.

Packet counts are recorded when packets cross network router interfaces. Most often, flow data aggregation is performed through Netflow[9] – a network monitoring protocol that is well integrated in modern network environments. Since Netflow counters are mostly generated directly on the network equipment, performance of the network may be affected. The overhead can be limited by performing traffic aggregation at the hardware level or even by decoupling traffic routing and flow computation by passively copying traffic data, similar to the case of packet inspection. As opposed to the packet-level inspection, which mostly provides signature-based analysis, an advantage of flow-based inspection is that it supports anomaly detection approach, as we describe in Section3.

Furthermore, network aggregates are applicable in the context of end-to-end encryption or privacy constraints of a particular environment, because they omit packet payloads from analysis. This property facilitates public availability of real traffic data aggregated in the form of flows, which is extremely valuable for open network intrusion detection research. A NIDS can also implement a hybrid approach that combines DPI and traffic flow analysis in a number of variations, mostly relying on the flow-based analysis complemented with an occasional payload inspection of suspicious traces.

3 A data analysis approach

Monitored traffic traces – in the form of full captures or aggregated flows – are analyzed to find indicators of potential attacks. In the early days, review of monitored activities for intrusion detection was performed manually by security analysts or system administrators. They used to devise and manually adjust rules and heuristics that would help to find harmful packets and identify suspicious behavior. The volume and increasing complexity of monitored data has long deemed any such manual efforts insufficient and prompted the community to introduce automation. Already in the 1990s, progress in AI research enabled investigation of ML techniques[3] in application to intrusion detection. The power of data analysis is in interpreting large amounts of data and automatically discovering new relevant knowledge – a highly valuable capability in the ever-growing and ever-changing security landscape.

A ML-based IDS employs a data-driven approach to intrusion detection – it uses machine learning methods to autonomously learn characteristic rules and patterns from previously observed data. For the case of network intrusion detection, the abundance of network traffic data creates an opportunity to apply data-driven techniques. A ML-based NIDS configured for *misuse detection* can detect variants of known attacks by finding patterns sufficiently similar to previously seen malicious traffic. On the other hand, a ML-based NIDS that performs *anomaly detection* can model 'normal' behavior of the system by learning from benign network traffic, and catches anomalous patterns that deviate significantly from the baseline. In the context of high non-stationarity and strong heterogeneity of network traffic, another strength of ML is its ability to dynamically adapt to changes in the network when exposed to new data.

The prolific use of AI cannot be merely attributed to impressive automation capabilities of ML algorithms, but also heavily relies on expert involvement. In order to benefit from the advantages offered by ML, the designer of a ML-based NIDS applies their domain expertise to create an appropriate learning system. First and foremost, an in-depth understanding of the network environment and the threat model are required in the *data representation* phase that converts monitored traffic data into a suitable format for ML. Applying ML methods directly to raw monitoring data, such as full pcap files, is not only hardly computationally feasible, but also does not usually yield useful results. The reason is that the numerous values in their original form are not equally relevant to the learning problem, which is especially true for data of such complex structure and

[3]Machine learning is defined as a subfield of AI that focuses on data-driven modeling of concepts, while deep learning is a subfield of machine learning that uses a particular family of techniques – artificial neural networks with representation learning.

overwhelming volume as network traffic. Therefore, an IDS designer leverages expert knowledge to find a compact representation of raw traffic which conveys characteristics that are most relevant to the task of misuse or anomaly detection. This step is known as *feature extraction* – a transformation that converts high-dimensional input data into low-dimensional *features* that capture its underlying structure. As a result, information-rich raw data is represented as a *feature vector* that is appropriate for training and testing a ML model. Today there exist two major general approaches to extracting features: (i) *feature engineering* – hand-crafting most informative features with the use of practical experience and intuition about the problem, and (ii) *feature learning* – automated feature extraction with the use of learning algorithms. While the former has traditionally played a predominant role, the latter is receiving increasing attention lately in light of recent progress in deep learning (DL) research.

Feature engineering & selection. The quality of features constructed from input data is one of the most influential factors that define the effectiveness of a ML algorithm. The resulting feature vectors are expected to be compact and most informative, and for manually extracted features, each dimension of the vectors usually has clear interpretable semantics. A common choice of features is a statistical description of data, such as distributions of attributes or groups of attributes. For a NIDS, the most widespread input data format is traffic flow data, and the latest versions of NetFlow already compute very basic *statistical features* of traffic flows on-the-fly, including bytes per packet and packets per flow. A lot of research and engineering effort has gone into finding the most meaningful and optimal statistical flow features. Most approaches extract simple aggregated features, e.g., average packet lengths, the variance in payload size, bytes transmitted per second, maximum packet inter-arrival time, the proportion of TCP flags, and more. Despite the simplicity, statistical features turn out to be highly informative of the nature of a particular network connection, as they are quite effective in revealing traffic anomalies and particular known malicious patterns. For instance, a DoS attack is characterized by sending many packets in one direction within a short time period, making flow statistics well-suited for detection. The recent examples of datasets with flow statistics are CICIDS2017 and CSE-CIC-IDS2018 [10]. The latter is an enhanced and expanded version that goes beyond one-flow statistics: CSE-CIC-IDS2018 introduces aggregated measurements in relation to other observed flows that could be relevant to a particular flow, e.g., number of recent connections from the same source. Such information derived across multiple flows is highly instrumental, e.g., in detecting attacks executed over multiple connections, i.e. from different sources or against multiple victims in the network. Probe attacks, DDoS attacks and bot communication are the obvious examples, where one flow in isolation might appear completely harmless, while the overall behavior is more indicative.

Temporal statistical features aim to solve the problem of a narrow one-flow view by aggregating traffic information over time across multiple flows, thereby respecting temporal dependencies between them. One of the latest examples is temporal statistics introduced by Mirsky et al. [11].

The resulting set of extracted features is often further optimized through automatic *feature selection* in order to discard redundant or irrelevant features and reduce dimensionality of data. Machine learning models may benefit in both effectiveness and

efficiency from compact but sufficient representations of data. For different approaches to composing and reducing the feature set, we refer the reader to the corresponding surveys [12,13].

Feature learning. As ML approaches struggle with high-dimensional inputs, deep neural networks have recently been embraced for feature learning and dimensionality reduction. DL approaches are able to automatically extract discriminative internal representations of the input through a series of non-linear transformations. Several studies in the literature applied *deep belief networks* for misuse and anomaly detection in order to obviate manual engineering of traffic flow features [14,15,16,17]. As a result, automatically learned representations have proven to be more robust to irrelevant deviations in data and thus contributed to higher generalizability of ML models to earlier unseen patterns. The immediate drawback of automatically learned abstract features is that they do not provide clear semantics and are hardly interpretable. This calls for additional methods to verify what the DL-based feature extractor has learned and to explain the model's decisions.

3.1 Machine learning for NIDS

Good quality features directly impact performance of a learning algorithm. When selecting an appropriate machine learning model, it is crucial to understand how to leverage the properties of the features and relationship between them. For instance, streaming data represented as a time series consists of temporal features, which are best interpreted with a model capable of recognizing temporal dependencies. The choice of a learning algorithm, however, starts with defining a concrete ML problem statement that most accurately represents the task of intrusion detection, be it misuse- or anomaly-based. This encapsulates at the very least such influential factors as (i) expected input and output of the system, (ii) assumption about the knowledge of all existing data categories, and (iii) availability of annotated training data.

Input and output. The expected input implies the data representation, i.e. the types and dimensionality of extracted and selected features. The expected output of the model is a design choice of how to present the inferred information about the event for further analysis and response. In general, for a given test instance, the output can be a *label*: malicious vs. benign or anomaly vs. benign. For anomaly detection in particular, it can also be an anomaly score that indicates significance of the detected anomaly for further investigation. Optionally, the model can also provide its confidence score for each decision made.

Closed-world and open-world assumptions. The assumption about the knowledge of all data categories is what largely drives the choice between misuse and anomaly detection in the first place. A misuse detector is typically deployed under a *closed-world* assumption, which implies that all possible data categories, i.e. types of intrusions, have been seen at the training stage of the model. A common closed-world NIDS employs a ML *classifier* that learns to recognize a traffic instance as benign or belonging to one or another attack class, thus performing *intrusion recognition*. Closed-world detection has been thoroughly researched in the network security domain, and has been traditionally favored by industry due to predictability and high detection rates. However, in operation,

such a model can only detect known malicious behaviors and assumes that no unexpected attack type may appear.

In reality, a network environment operates under a much more challenging threat model that includes known attacks, new variants of known attacks and completely novel, earlier unseen cyber-threats, comprising an *open world* of possibilities (hence the title of this chapter). In order to enable open-world detection, a ML problem statement has to change from standard classification to either *open-world recognition* [18] (also called open-set recognition/classification), or *anomaly detection* [19] (also called outlier detection). An open-world classifier performs its originally intended task, but also leverages additional mechanisms to be able to identify novel patterns as instances that cannot be confidently classified as one of the learned attack types. Anomaly detection algorithms, however, are inherently open-world: as was explained earlier, anomaly detection exclusively relies on knowledge of benign data (normal, background traffic), and flags any sufficiently deviant pattern as a potential intrusion attempt of unknown nature. Therefore, in operation, an anomaly-based NIDS is an open-world detector as it targets both known and unknown attacks, although it can similarly use patterns of earlier seen attacks for model evaluation at the design stage. The main drawback of anomaly detection is its dependence on the notion of 'normality' – a pattern that deviates from normal data for benign reasons is also flagged as potentially malicious, usually causing a high number of false alerts. Further in Section 4, we zoom in on the challenges of traffic anomaly detection.

There are many studies in the literature that join anomaly detection with misuse detection in an attempt to combine the strengths of both paradigms: improve the detection rate and minimize the rate of false alerts. In fact, commercial platforms very rarely use anomaly detection in isolation, but rather adopt the *hybrid* approach [20]. The mismatch between the promises of anomaly detection and its actual adoption in industry is what demands a more explicit academic focus on anomaly-based IDS research.

Supervised and unsupervised learning. Another fundamental distinction between ML approaches relates to availability of annotated/*labeled* data. *Supervised* ML, such as classification, is a learning mode that relies on labeled training data. Namely, a classifier requires a significant number of representative labeled training examples from all the considered classes. Therefore, a classifier-based misuse detector works with a continuously updated database of known malicious patterns that need to be well represented in training data for a NIDS. Acquiring labeled malicious data is expensive, as it requires either manual investigation by network experts, or development of automated labeling algorithms, which essentially create a chicken and egg problem. Moreover, supervised learning is in general highly sensitive to *class imbalance* in data, demanding equal representation of every class. Otherwise, a classifier trained with imbalanced data becomes biased towards the majority class and largely ignores instances belonging to the minority class of interest. Since in network environments benign traffic is predominant, the benign class outweighs malicious traffic classes in labeled training data, causing a much lower representation of attacks. This undermines the sole purpose of intrusion detection, since the most interesting and inherently rare intrusions become overlooked. The issue can be addressed by using specialized techniques to increase importance of the minority attack classes [21]. Another solution is attack simulation performed to generate

more malicious traffic for training under an assumption of its representativeness of real intrusions. However, for simulated benchmark datasets, the class imbalance problem is not unheard of, either.

Unsupervised ML paradigm obviates the need for labeled data altogether. A general example is *clustering*, which performs exploratory data analysis to draw inferences and find hidden patterns and correlations in unlabeled data. Clusters are automatically formed with the use of a similarity measure between instances. Unsupervised approaches generally do not assume any a priori knowledge on the data distribution and labels, which corresponds to a realistic NIDS scenario. In practice, though, *semi-supervised* anomaly detection is often applicable under the assumption of availability of labeled normal data. As the shortage of malicious labeled traffic is the main issue, both supervised and semi-supervised approaches pose the biggest interest for NIDS research.

While supervised NIDS approaches are widely utilized and thoroughly studied in the literature, they either violate the open-world context of network security, or extensively rely on manual data labeling, or both. In recent years, unsupervised and semi-supervised techniques in application to NIDS are gaining more traction; however, the research is largely ongoing. The community have composed a number of excellent surveys on the topic that provide detailed taxonomies and analyses of existing ML-based approaches to NIDS. For the closed-world misuse detection research, we refer the reader to the corresponding expansive literature that surveys supervised classification methods [22,23]. Further in our discussion, we elaborate on the open-world NIDS research, specifically, unsupervised or semi-supervised anomaly-based ML paradigms, which we believe deserve more attention in the field of network security. Hence, our aim is to complement the existing surveys [24,25,26,27], which provide in-depth analyses of individual techniques, with a broad overview of the current solution space and the key remaining challenges.

3.2 Anomaly detection for open-world NIDS

An *anomaly* is commonly defined as a rare pattern that does not conform to expected behavior. In machine learning, an anomaly is detected as an outlier with respect to the region representing normal data. In intrusion detection, different types of anomalies are typically mapped to different types of malicious behavior:

1. *Point anomalies* – individual data observations that lie outside of the normal behavioral boundaries (relative to the rest of the data). For instance, sophisticated network exploits that aim to gain unauthorized access or escalate privileges, such as buffer overflow attacks or web attacks, can be carefully deployed through one packet payload, or one traffic flow. Simple probing attacks, launched through malformed packets, incomplete connections or with incorrect combinations of header attributes, also form a point anomaly.
2. *Contextual anomalies* – individual data observations that are anomalous in a given context. One example of a contextual anomaly are stealthy probing attack [28], where each individual packet and the whole connection may correspond to normal traffic. However, given the context of systematic information collection without meaningful interaction, the connection becomes anomalous. Some botnet traffic

can also arguably be considered a contextual anomaly: while communication with the C&C server can by itself form a benign connection, its timestamp may point to suspicious behavior.

3. *Collective anomalies* – multiple data observations occurring together that differ from normal behavior. The key here is the collective occurrence of those observations, as each single instance is not anomalous by itself. A common example is a DoS attack, where only one connection is legitimate, but the abundance of similar connections becomes anomalous as they overwhelm the target system. Another example is the brute force network attacks, where a single incorrect log-in attempt is not yet suspicious, but a sequence of frequent attempts makes them collectively anomalous.

For anomaly-based network intrusion detection to be effective, the following assumptions have to hold:

- *Benign data assumption* – there exists a region with well-defined boundaries that encompasses all the normal traffic data.
- *Clean training data* (for semi-supervised approaches) – benign training data acquired by collecting live background traffic is attack-free.
- *All attacks are rare and anomalous* – traffic generated by malicious actions related to network intrusions deviates sufficiently from the normal traffic and will only constitute a small fraction of monitored data.
- *All anomalies are malicious* – whenever a deviant pattern is observed, it presents evidence of a potential intrusion.
- *Attacks are universal* – given correct modeling of normal data, all types of attacks are detected equally well.

Naturally, the extent to which these properties can be safely assumed differs from one environment to another and strongly depends on the threat model of a NIDS. For instance, it is already clear that a system tailed to detection of one of the three types of anomalies is not a universal detector for all attack types. We elaborate more on the implications of these assumptions in the next section.

Major anomaly detection techniques explored in application to intrusion detection can be grouped in four categories.

Statistical approaches. Statistical anomaly-detection works based on the principles of the statistical theory to model the distinction between normal and anomalous. A common solution for anomaly-based NIDS is Principle Component Analysis (PCA) – a dimensionality reduction approach that projects high-dimensional data onto a normal and anomalous subspace. PCA does not assume any statistical distribution and is known for low computational complexity. Lakhina et al. explored the use of PCA on network traffic [29]; however, further studies revealed sensitivity of PCA to such aspects as the level of traffic aggregation and small noises in the normal subspace [30], which the state-of-the-art solutions aim to overcome [31,32].

Clustering approaches. Clustering groups unlabeled traffic based on a chosen similarity metric, e.g. a Euclidean distance, and flag outliers as potential intrusions. Plenty of clustering algorithms have been applied to NIDS. More recent works utilize k-means with optimizations [33], Gaussian mixture model [34], incremental grid clustering [35] and novel affinity propagation clustering [36]. The advantages of clustering usually are

stable performance and a possibility of incremental updates. On the other hand, clustering is not intrinsically optimized for anomaly detection, can be time-consuming and heavily depends on distance measures and tuning.

One-class classification. A semi-supervised adaptation of classification is called one-class classification, as it only utilizes negative examples in training, i.e. benign data. A data instance that falls outside of the learned class, depending on the chosen threshold, is considered anomalous. One-class Naive Bayes [37] and one-class Support Vector Machine (SVM) [20] are recent examples of traditional ML approaches used for anomaly-based NIDS. While we already discussed deep learning approaches for feature learning and dimensionality reduction, deep neural networks are also being employed as sole anomaly detectors. Deep belief networks [38], variational autoencoders [39] and ensembles of light-weight shallow autoencoders [11] have been successfully used for anomaly detection on network data, demonstrating good generalization abilities and self-adaptive nature of neural networks. A lingering issue of DL-based anomaly detectors is that by themselves, they are not optimized for anomaly detection, therefore selecting appropriate thresholds and tuning the architecture is challenging.

Time-series forecasting. Forecasting is a semi-supervised predictive ano- maly detection approach specifically tailed for sequential inputs (including data with high seasonality), as they are capable of detecting temporal anomalies in complex scenarios [40]. The idea is to perform rolling predictions based on observed normal data and compare them with new observations. Strong deviation from predictions thus indicates an anomaly. While there exist numerous advanced time-series modeling and forecasting techniques, from traditional exponential smoothing [41] to more modern ones such as recurrent neural networks [42], their application to network traffic anomaly detection has thus far been limited. This approach does not only heavily rely on unpolluted training benign data and clear observable trends, but also struggles with high-dimensionality and categorical inputs. In light of remarkable performance by recurrent neural networks in anomaly detection on multi-dimensional time-series, we expect new forecasting NIDS approaches to appear in near future.

In the remainder of this chapter, we give a fresh look on the state of open-world NIDS research in terms of main challenges and recent contributions.

4 Challenges and advances in open-world NIDS research

Machine learning algorithms, and anomaly detection in particular, have gained a lot of attention in network intrusion detection research because of its compelling potential in detecting novel attacks. A decade ago, the community brought into the spotlight the intrinsic challenges of open-world network intrusion detection [43,44,25,24]. It turned out that most of the conducted research explored ML-based IDS solutions under numerous unrealistic assumptions. In reality, with these wishful assumptions dropped, the effectiveness of ML-based solutions in detecting novel and known attacks falls way down below the estimated performance. In the context of a NIDS, ML algorithms are tasked with search for the unknown, while costs for mistakes in a security-critical environment are painfully high. A fundamental question was raised as to how appropriate

ML algorithms are to such defensive applications, and which guarantees they can give for operation in sensitive environments.

Since then, the security domain grew significantly, with attacks becoming more sophisticated and resourceful. A wide spectrum of cutting-edge machine learning techniques, including deep learning and big data analytics, have been proposed for a variety of applications. New benchmark NIDS datasets have been jointly developed and evaluated. In general, today we observe a closer collaboration between the AI and the security community. In light of the new developments, we revisit the primary conceptual issues of ML-based NIDS.

4.1 Original premise of anomaly detection

The underlying assumptions of machine learning underpin open-world ML-based NIDS solutions. To enable the full potential of ML, these assumptions have to align with domain-specific characteristics, which in the case of securing dynamic and modern network environments is not a trivial question. For anomaly detection specifically, the community is actively attempting to address some of the following fundamental questions:

Can normal data be modeled? Most of the studies attempt to model benign traffic; however, not all benign behaviors follow a common distribution. It is overwhelmingly hard to completely capture the notion of 'normality', so the safest assumption to make is that the model cannot describe all the possible benign instances. Hence, false alerts and missed attacks are unavoidable, and adjustment to novel benign patterns is necessary, which we discuss further in the section.

Is it possible to acquire clean training data? The current consensus is that normal traffic collected in a live environment is never attack-free without additional (manual) sanitization.

Are attacks rare? Certain illegitimate activities in the network (e.g., scanning) have become so common that they comprise a large fraction of background traffic [45]. Durumeric et al. [46] revealed that DDoS cannot be considered anomalous in most networks. However, even though large-scale attacks are not rare, these are not of the biggest interest for detection. More sophisticated intrusions such as APTs are still manifested in rare events.

Are attacks anomalous? The answer directly relates to the vague definition of traffic 'normality'. Due to the noisy and highly varied nature of traffic, attack features may in practice appear as variations of benign traffic. Iglesias et al. [47] have recently conducted an analytical study to assess the 'outlierness' of malicious traffic. They confirmed that network attacks have higher global distance-based outlierness averages; however, attack and normal traffic distributions strongly overlap. One can choose the feature space that maximizes the separation of benign and malicious traffic, which indicates that understanding the nature of target anomalies in a certain scenario is instrumental for anomaly detection. Another known issue is that attackers may attempt to make traffic features indistinguishable from normal traffic. We elaborate on the associated risks further in the section.

Are attacks universal and equally detectable? Taking everything into account, there is little ground in assuming that different types of intrusions can be detected in

one common manner. Moreover, the very definition of what is malicious differs across environments. Indeed, we observe the trend of developing NIDSs tailored to specific threat models. This includes, e.g., works that focus on botnet detection [48,49], DDoS detection [50], and especially APT detection [51], where data exfiltration through the network can be a target anomaly. It is quite unlikely that such targeted detectors generalize to other types of intrusions, but perhaps that should not be the initial goal. We advocate for deeper insight in target malicious activities even for open-world anomaly detectors, in order to adopt the most suitable strategies.

Is a detected anomaly an attack? Nowadays, it is commonly acknowledged that an anomaly detected by a NIDS is most probably a false alert. Even correctly detected anomalies are not always malicious: sometimes, deviations happen due to noise, changes in the underlying infrastructure or changes in the benign data distribution. Therefore, additional processing is required to investigate the issue, as monitoring and detection is just the earliest stage in the complex process of incident management. Additional analysis, attack correlation and response planning is a prerogative of Security Information and Event Management (SIEM) platforms [52]. While researchers have mainly focused on developing effective solutions for detection, studies on automatic intrusion response are still limited. The main challenge is in providing an accurate and informative description of the detected anomaly, including *interpretation* of the ML model's decision to raise an alert.

4.2 High error rates & performance estimation

Among the main problems with adoption of anomaly detection in mainstream security systems, a high *false positive rate* (FPR) is an immediate candidate. For an enterprise IDS, manual investigation and interpretation of alerts consumes expensive analyst time. Given the large volumes of processed data and a low *base rate* of attacks of interest, even a very small fraction of false alerts generated by a nearly-perfect model yields an unacceptably large absolute number, effectively rendering a NIDS unusable in the operational setting. This issue of *base rate fallacy* was raised two decades ago [53], and is seen today as an inevitable pitfall of open-world detection: *precision of an IDS will always be determined by both the base rate of different attacks and the FPR*. Regretfully, however, we lack historical statistics for the base rates of attacks in real computer infrastructures, and measuring them reliably is still considered beyond present capabilities [54].

As the tolerance for errors in the application domain is critically low, researchers started advocating for placing more emphasis on constraining the FPR while preserving high detection rates [44]. Since then, more studies have targeted this specific problem. We observe that the solution space can be mainly branched into five complementary directions: (i) further developing more precise learning algorithms to lower the FPR [26,27]; (ii) post-processing alerts with the use of context or prior knowledge in the system [55], in order to aid in manual diagnostics and potentially understand the nature of an anomaly; (iii) employing a *hybrid* approach by combining anomaly detectors with misuse detectors [56], which cannot detect novel attacks but are considered less prone to mispredictions. (iv) tuning model parameters and detection thresholds in order to obtain

optimal trade-offs in success rates and false alerts [57]; (v) modeling a realistic network environment in a structured manner to correctly estimate the FPR.

While the first four objectives are gradually unfolding in present research, the last one is fundamental and largely remains an open question. It relates to the inherent difficulties with evaluating an open-world detector, which started being actively discussed more than ten years ago [43,24,44] and still hold today. With more progress in this direction, future NIDS studies should adopt an appropriate evaluation methodology and correct metrics that correspond to an actual operational usage of the target system. This requirement encapsulates such a crucial issue as validating and testing the model on data that resembles real-world ratios of benign vs. attack data – which again relates to the base rate fallacy. Without satisfying these goals, performance numbers and errors rates achieved in lab conditions will remain hardly reliable or comparable. The issue is especially pronounced for unsupervised methods, which learn from distributions and spaces drawn from the observed data. Note that even modern benchmark datasets are not said to be representative of an actual ratio of normal and attack traffic, therefore they are most often not directly applicable for (unsupervised) open-world evaluation schemes. We detail on the representativeness of existing datasets further below.

All in all, the research on decreasing the FPR while preserving performance in a general NIDS setting is still unfolding. Despite some studies emphasizing the post-processing stage of predictions, there is generally not enough investigation being made on the nature of false alerts, while most of the works solely focus on increasing the detection rates instead. Even though anomaly detectors with manageable error rates are allegedly becoming more widely adopted in industry, these solutions are often designed for specific scenarios and their internals are rarely publicly available [24,26,58], preventing direct comparison. The field appears to be in the urgent need of a common comprehensive methodology for estimating and comparing performance and error rates of an open-world NIDS.

4.3 Representative datasets & ground truth

In IDS research, evaluation on benchmark datasets primarily serves a two-fold purpose: (i) *real-world performance estimation* of a particular algorithm, and (ii) *consistent comparison* between different approaches. In this respect, quality of data has a decisive influence on valid outcome of both objectives. Several critical studies have shown that many benchmark datasets do not adequately represent the real problem of network intrusion detection, discrediting performance numbers achieved in laboratory conditions. As a response, over the last 10 years the community has collectively devised the criteria that reliable research traffic data should meet [25,59,60,61,62], which encapsulate the following dataset properties: (i) realistic w.r.t. real production environments; (ii) valid w.r.t. completeness of traces; (iii) labeled; (iv) correctly labeled w.r.t. benign training data for anomaly-detection; (v) highly variant and diverse w.r.t. used services, protocols, benign behaviors and attacks; (vi) correctly implemented w.r.t. real attack scenarios; (vii) easily updatable with new services and attacks; (viii) reproducible for periodical updates and performance comparisons; (ix) shareable/non-sensitive; (x) well-documented. Despite this recently achieved consensus and clarity in guidelines, a lot of fundamental limitations of the task hamper both creation and publication of a corresponding proper dataset.

Consequentially, many researchers have kept using the existing suboptimal datasets for the sake of comparison with prior work. Nevertheless, the research community is making tangible progress in this direction by exploring both possibilities to contribute a new dataset: (i) generate synthetic traffic, and (ii) collect real traffic in a production environment.

Generation of synthetic datasets provides the luxury of a controlled environment, clean labels and no privacy concerns. The main challenge, however, is in simulation of realistic background traffic, lately attempted through statistically modeling user behavior [60,63]. Even though creators of modern synthetic datasets strive to satisfy the requirements and minimize occurrence of *simulation artifacts*, a practice of evaluating a novel IDS on a synthetic dataset solely, however, is often criticized as insufficient. While it can be reasonable to compare different frameworks on synthetic data, evaluation on diverse network traffic collected in a live environment over a lengthy period of time is becoming the desired norm in NIDS research. Real traffic, on the other hand, should be stripped of confidential data, carefully labeled and rigorously sanitized in order to meet the established criteria. Several studies contributed approaches to sanitization of traffic [64,65,66,67] in order to not only label embedded attacks and benign traces, but also to pre-select the most representative instances. Automated sanitization uses such methods as entropy analysis and signature-based attack labeling, which may result in erroneous ground-truth. Manual sanitization hardly scales and is prone to human bias, which threatens reliability and representativeness of the dataset, respectively. However, manual supervision in labeling seems unavoidable when it comes to zero-day network attacks.

All in all, it is unclear whether a perfectly sanitized real traffic-based dataset can be obtained. Hence, learning algorithms that are robust to the inevitably occurring noise in labels would give a strong advantage from the operational point of view. Promising examples for anomaly detection on imperfectly labeled traffic include, e.g., robust PCA algorithms [68,69] and a convex combination of anomaly detectors' outputs [70].

Another suggestion for creation of an open, real NIDS dataset was voiced by Gates et al. [43], who promoted a community-based approach. One prominent example is the MAWILab dataset [71] – a public repository for automated labeling and performance estimation that has since been continuously updated and collectively labeled with the use of state-of-the-art anomaly detectors. While anomaly detection solutions on these data are still scarce [72], we believe that such collective efforts establish a strong foundation for open-world detection research.

4.4 Concept drift

In dynamic environments, events undergo gradual and abrupt changes over time, which cause a shift in data distribution known as the *concept drift* [73]. When developing data-driven real-time defensive solutions such as ML-based IDSs, it is crucial to account for concept drift, otherwise the model's performance is unpredictably and heavily impacted. For an anomaly detector, this implies the need to track drift in the data in order to continuously adjust to the new definition of normal behavior, instead of erroneously flagging these changes as anomalies. A direct way to re-adapt the system accordingly is to re-train the model on new data, as is strongly recommended in the literature [44,24,74].

In anomaly detection literature, the problem of detecting newly emerging patterns is referred to as *novelty detection*, when previously unobserved detected patterns in data are incorporated into the normal model. In time-series analysis, a similar idea is defined as *change point detection* [75] that aims to detect points in a time-series from which the data distribution changes. Conventional approaches often suggested for ML-based defenses aim to detect concept drift by recognizing model's performance degradation on streaming data and identifying an appropriate moment for a model update. Naturally, the crucial trade-off emerges between the detection delay and the detection quality. Some families of ML algorithms such as neural networks can adapt through a continuous retraining mechanism known as *online/incremental learning*, exemplified by a DL-based IDS that analyzes log data [76]. By incorporating the most recent changes in system logs into the DNN model, a DL-based IDS can adjust to the newly emerging patterns in a timely manner. Incremental learning can also be applied for traditional ML algorithms, albeit with high computational complexity. For instance, Rassam et al. [77] utilize an adaptive principal component classifier-based anomaly detector that tracks dynamic normal changes in real sensor data. However, effectiveness and practicality of incremental learning or re-training for an anomaly IDS on network traffic – non-stationary streaming data – largely remain unexplored. Raza et al. [78] developed a theoretical approach that addresses detection of covariate shifts in generic non-stationary environments and can potentially aid in IDS concept drift. One promising approach to an autonomic anomaly NIDS was proposed by Wang et al. [36], who use novel clustering algorithms to label new data and dynamically adapt to normal behavior. They show efficacy of their algorithm on a private dataset of real HTTP traffic streams. Zhang et al. [79] employ a competing approach specifically tailored to high-dimensional streaming data. To account for concept drift, they perform adaptive subspace analysis that fully relies on human feedback to prune away irrelevant subspaces of anomalies. As this novel algorithm is only evaluated on the KDD'99 dataset, its generalizability to real traffic and scalability to live environments are unknown. Dong et al. [80] developed a batch-based adaptation approach that utilizes an SVM classifier and incorporates human feedback to determine when re-training is necessary. Their evaluation is limited to malicious web requests, and they use a public dataset with HTTP traffic.

Currently, a thorough investigation of concept drift detection and adaptation techniques for open-world NIDSs is pending, and the lack of public representative benchmark datasets that contain labeled shifts in traffic has been one of the largest roadblocks. A notable recent contribution is the UGR'16 dataset [81] – real anonymized Netflow data for adaptive NIDS research that includes long-term traffic evolution and periodicity.

4.5 Real-time detection

In the era of growing risk and severity of cyber-attacks, an effective NIDS is expected to detect potential threats immediately as they occur in the network. An ideal *real-time* detector processes and analyzes a continuous stream of data in its natural sequential form and makes immediate decisions online [82]. Anomaly detection is regarded as indispensable in early open-world detection of novel, unusual behaviors, and yet the existing approaches are not effective enough in real-time detection [26] and still largely resort to offline analysis, or batch processing at best, allowing some intrusions to go

unnoticed for days. In the meantime, the bar for real-time processing capabilities is only increasing: not only does the internet traffic double each year, but in addition to that, the growth of the Internet of Things (IoT), sensors, smart cities, mobile clouds, autonomous vehicles, and other emerging technologies has unleashed enormous amounts of generated network data. Cisco has reported [83] that by 2022, the omnipresent non-PC devices are estimated to drive 81% of global internet traffic, opening the gate for more large-scale network attacks against small connected devices. The network data of today is already characterized by huge volume, velocity, variety and veracity, fulfilling the definition of *big data*. Traditional ML-enabled NIDS have not been developed to handle big data, but largely aimed at enhancing learning algorithms, which mostly results in increasing the computational complexity and processing time [26,22,27], further hindering real-time analysis.

As the demand for uninterrupted security monitoring raises higher by day, novel solutions are required to facilitate large-scale, real-time detection. Hoplaros et al. [84] explored *data summarization* techniques that mine patterns in summaries of network traffic to approximate final decisions and improve efficiency of detection. Since this approach effectively allows to cut offline detection runtimes, the authors propose to develop stream data summarization and distributed summarization methods for online detection. The downside is that complex summarization on big data contributes to opacity of model predictions, while threat analysis benefits from more granularity and transparency in decision-making.

Collaborative intrusion detection systems that employ several distributed monitors for collection and analysis of traffic pose an alternative to the bottleneck stand-alone anomaly detectors. A collaborative NIDS is considered to be much more efficient in analysis of numerous data streams traversing through large networks and IT ecosystems. Vasilomanolakis et al. [85] provide a taxonomy and a detailed survey on the topic, including possible topologies and threat models for a collaborative NIDS. Zarpelao et al. [86] presented a survey of stand-alone and collaborative IDS solutions specifically for IoT infrastructures. Crucially, most of the modern NIDS research on large-scale networks, IoT in particular, does not provide enough details for reproducibility and use private specifically chosen testbeds or simulation tools. Moreover, the internal mechanisms of existing commercial products are also hardly available. All in all, a thorough investigation on public data with a standardized evaluation strategy is required to assess effectiveness of a collaborative NIDS in real-time detection of sophisticated attacks in modern network environments.

Suthaharan et al. [87] were among the first to highlight the challenging big data properties associated with network monitoring for security. They advocated incorporating known big data frameworks, e.g. Hadoop [88], into a ML-based NIDS framework, in order to combine *big data processing* tailored to real-time analytics with supervised ML classifiers and representation-learning techniques. This integration requires to rethink implementations of ML algorithms in general and introduce *parallelization* by either dividing data into separately processed subsets, or dividing a ML algorithm into concurrently performed steps. Later on, the discussion was extended to anomaly-based NIDS in order to enable real-time open-world detection on large streaming data [89,90,11,91,72]. Recently, Habeeb et al. [58] have thoroughly reviewed real-time network anomaly de-

tection algorithms and discussed the aspects and challenges of their application to big network traffic data. Despite the promise of big data frameworks widely deployed in other domains, in network security we still observe a premature state of big data processing capabilities. Efficient model and parameter selection, automation of data filtering and curation, dynamic resource allocation, reduction of power and memory consumption are only a few associated future research directions. With these enhancements, advance anomaly detection in combination with modern big data tools should be adequate to handle large-scale real-time detection, feature extraction and selection, labeling, and model retraining.

4.6 Adversarial robustness

Attackers have always had great incentives and tools to evade detection by a NIDS. Knowing the blind spots of the detector, an adaptive attacker chooses the optimal strategy that fools a NIDS into thinking their traffic is legitimate. Misuse-based detectors that inspect traffic on the packet level are traditionally evaded through such means as encryption, obfuscation and packet fragmentation, which make sure that malicious traffic does not match a known signature. A general anomaly-based detector is vulnerable to mimicry attacks, which modify malicious traffic in such a way that it corresponds to normal traffic patterns [92,93]. Besides, any type of a NIDS is susceptible to various DoS attacks, which can overload the detector with meaningless connections to create a bottleneck, so that the actual malicious connection comes through unnoticed. The security analysis of novel NIDS solutions with respect to adaptive attackers is regarded as a crucial research angle.

In the last decade, more attention was brought specifically to evasion of ML-enabled NIDS. When placing a ML model at the core of a defensive system, one will always involuntarily introduce a new attack vector of undetermined severity. *Adversarial machine learning* is a set of techniques that exploit specific vulnerabilities of ML algorithms in order to trigger an incorrect output. Corona et al. [94] describe a general adaptive threat model for a NIDS and review studies of one particular category of adversarial ML – *poisoning attacks*. Automatic adaptability of ML to changes in normal traffic allows attackers to poisons the model's decision boundary by inserting adversarial noise in benign traffic that is consequently used for training. Kloft and Laskov [95] explored poisoning attacks against centroid anomaly detection and confirmed its effectiveness on feature vectors representing real HTTP traffic. The second type of adversarial ML attacks is *evasion*, where a malicious *adversarial example* evades detection by introducing carefully crafted minor perturbations in the network communication. Apruzzese and Colajanni have recently demonstrated the effectiveness of evasion on a closed-world botnet classifier [96]. They generated adversarial examples on Netflow features of real botnet traffic [97].

Crucially, the prevailing majority of existing works make unrealistic assumptions about attacker capabilities, including direct access to the extracted features and knowledge of training data. Adversarial examples constructed in the *feature space* still have to be mapped back to the *problem space* – actual network traffic – on-the-fly, as that is the level of attacker's access [98]. The existing NIDS research does not explicitly investigate the practical feasibility and effectiveness of adversarial ML strategies in the context of

real-world constraints. As the impact of adversarial learning in the operational scenario of a NIDS is still unknown, hardening detectors against adversarial perturbations or incorporating additional defenses are currently not considered strongly motivated.

5 Conclusion

Machine learning for network intrusion detection is an extremely intriguing and potent research direction, which – despite its strong theoretical base – is still lacking devoted attention in defensive security. Today, the community acknowledges the non-stationarity and adversarial nature of security applications, promoting thoughtful and realistic evaluation of effective and adaptable ML-based defenses. We had to face the hard truths about domain-specific properties and limitations of ML in open dynamic environments. While we cannot create a silver-bullet solution to network intrusion detection, we can deepen our understanding of the underlying issues and provide fundamentally sound ML techniques for NIDS.

In this chapter, we reviewed the wide spectrum of impressive research efforts in the area of anomaly-based NIDS and highlighted the main challenges that should become the focus of the future research. Our analysis encompasses the domain misalignment with the original assumptions of anomaly detection, high error rates, the problem of performance estimation and comparison, availability of realistic datasets and reliable ground truth, adaptability to concept drift, feasibility of real-time detection, and adversarial impact. From our discussion, it is evident that there is no clear-cut separation between various challenges or desired properties of open-world detection systems.

We hope that the future research will reason about network intrusion detection in a more principled way that considers all important aspects in conjunction and allows to systematically assess how they affect each other. To achieve that, we encourage the community to collectively devise appropriate ML methodologies to develop and evaluate realistic open-world network intrusion detectors in different environments and threat models. As this is a tough task for the years to come, we need to scale research by composing benchmarking scenarios under a common set of assumptions to fairly compare novel methods. To this end, open realistic datasets and open-source implementations are of the highest priority.

Acknowledgments This research is partially funded by the Research Fund KU Leuven, and by the Flemish Research Programme Cybersecurity.

References

1. M. Roesch *et al.*, "Snort: Lightweight intrusion detection for networks.," in *Lisa*, vol. 99, pp. 229–238, 1999.
2. "Suricata ids [online]," 2010. Accessed: Jun. 1, 2020.
3. E. Albin and N. C. Rowe, "A realistic experimental comparison of the suricata and snort intrusion-detection systems," in *2012 26th International Conference on Advanced Information Networking and Applications Workshops*, pp. 122–127, IEEE, 2012.

4. V. Paxson, "Bro: a system for detecting network intruders in real-time," *Computer networks*, vol. 31, no. 23-24, pp. 2435–2463, 1999.
5. G. K. Varadarajan and M. Santander Peláez, "Web application attack analysis using bro ids," *SANS Institute*, vol. 90, 2012.
6. J. Sherry, C. Lan, R. A. Popa, and S. Ratnasamy, "Blindbox: Deep packet inspection over encrypted traffic," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, pp. 213–226, 2015.
7. J. Ning, G. S. Poh, J.-C. Loh, J. Chia, and E.-C. Chang, "Privdpi: Privacy-preserving encrypted traffic inspection with reusable obfuscated rules," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1657–1670, 2019.
8. A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, "An overview of ip flow-based intrusion detection," *IEEE communications surveys & tutorials*, vol. 12, no. 3, pp. 343–356, 2010.
9. B. Claise, G. Sadasivan, V. Valluri, and M. Djernaes, "Cisco systems netflow services export version 9," 2004.
10. "Datasets by canadian institute for cybersecurity [online]," 2018. Accessed: Apr. 2, 2020.
11. Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: an ensemble of autoencoders for online network intrusion detection," *arXiv preprint arXiv:1802.09089*, 2018.
12. J. J. Davis and A. J. Clark, "Data preprocessing for anomaly based network intrusion detection: A review," *computers & security*, vol. 30, no. 6-7, pp. 353–375, 2011.
13. H. T. Nguyen, K. Franke, and S. Petrovic, "Feature extraction methods for intrusion detection systems," in *Threats, Countermeasures, and Advances in Applied Information Security*, pp. 23–52, IGI Global, 2012.
14. N. Gao, L. Gao, Q. Gao, and H. Wang, "An intrusion detection model based on deep belief networks," in *2014 Second International Conference on Advanced Cloud and Big Data*, pp. 247–252, IEEE, 2014.
15. S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning," *Pattern Recognition*, vol. 58, pp. 121–134, 2016.
16. K. Alrawashdeh and C. Purdy, "Toward an online anomaly intrusion detection system based on deep learning," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 195–200, IEEE, 2016.
17. N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018.
18. A. Bendale and T. Boult, "Towards open world recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1893–1902, 2015.
19. V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.
20. G. Kim, S. Lee, and S. Kim, "A novel hybrid intrusion detection method integrating anomaly detection with misuse detection," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1690–1700, 2014.
21. C. Thomas, "Improving intrusion detection for imbalanced network traffic," *Security and Communication Networks*, vol. 6, no. 3, pp. 309–324, 2013.
22. A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications surveys & tutorials*, vol. 18, no. 2, pp. 1153–1176, 2015.
23. D. S. Berman, A. L. Buczak, J. S. Chavis, and C. L. Corbett, "A survey of deep learning methods for cyber security," *Information*, vol. 10, no. 4, p. 122, 2019.

24. P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *computers & security*, vol. 28, no. 1-2, pp. 18–28, 2009.
25. M. Tavallaee, N. Stakhanova, and A. A. Ghorbani, "Toward credible evaluation of anomaly-based intrusion-detection methods," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 5, pp. 516–524, 2010.
26. M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016.
27. G. Fernandes, J. J. Rodrigues, L. F. Carvalho, J. F. Al-Muhtadi, and M. L. Proença, "A comprehensive survey on network anomaly detection," *Telecommunication Systems*, vol. 70, no. 3, pp. 447–489, 2019.
28. S. Staniford, J. A. Hoagland, and J. M. McAlerney, "Practical automated detection of stealthy portscans," *Journal of Computer Security*, vol. 10, no. 1-2, pp. 105–136, 2002.
29. A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," *ACM SIGCOMM computer communication review*, vol. 34, no. 4, pp. 219–230, 2004.
30. H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of pca for traffic anomaly detection," in *Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pp. 109–120, 2007.
31. G. Fernandes Jr, J. J. Rodrigues, and M. L. Proenca Jr, "Autonomous profile-based anomaly detection system using principal component analysis and flow analysis," *Applied Soft Computing*, vol. 34, pp. 513–525, 2015.
32. G. Fernandes Jr, L. F. Carvalho, J. J. Rodrigues, and M. L. Proença Jr, "Network anomaly detection using ip flows with principal component analysis and ant colony optimization," *Journal of Network and Computer Applications*, vol. 64, pp. 1–11, 2016.
33. A. Karami and M. Guerrero-Zapata, "A fuzzy anomaly detection system based on hybrid pso-kmeans algorithm in content-centric networks," *Neurocomputing*, vol. 149, pp. 1253–1269, 2015.
34. E. Bigdeli, M. Mohammadi, B. Raahemi, and S. Matwin, "Incremental anomaly detection using two-layer cluster-based structure," *Information Sciences*, vol. 429, pp. 315–331, 2018.
35. J. Dromard, G. Roudière, and P. Owezarski, "Online and scalable unsupervised network anomaly detection method," *IEEE Transactions on Network and Service Management*, vol. 14, no. 1, pp. 34–47, 2016.
36. W. Wang, T. Guyet, R. Quiniou, M.-O. Cordier, F. Masseglia, and X. Zhang, "Autonomic intrusion detection: Adaptively detecting anomalies over unlabeled audit data streams in computer networks," *Knowledge-Based Systems*, vol. 70, pp. 103–117, 2014.
37. M. Swarnkar and N. Hubballi, "Ocpad: One class naive bayes classifier for payload based anomaly detection," *Expert Systems with Applications*, vol. 64, pp. 330–339, 2016.
38. U. Fiore, F. Palmieri, A. Castiglione, and A. De Santis, "Network anomaly detection with the restricted boltzmann machine," *Neurocomputing*, vol. 122, pp. 13–23, 2013.
39. Q. P. Nguyen, K. W. Lim, D. M. Divakaran, K. H. Low, and M. C. Chan, "Gee: A gradient-based explainable variational autoencoder for network anomaly detection," in *2019 IEEE Conference on Communications and Network Security (CNS)*, pp. 91–99, IEEE, 2019.
40. S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, 2017.
41. M. Szmit and A. Szmit, "Usage of modified holt-winters method in the anomaly detection of network traffic: Case studies," *Journal of Computer Networks and Communications*, vol. 2012, 2012.
42. P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proceedings*, vol. 89, Presses universitaires de Louvain, 2015.

43. C. Gates and C. Taylor, "Challenging the anomaly detection paradigm: a provocative discussion," in *Proceedings of the 2006 workshop on New security paradigms*, pp. 21–29, 2006.
44. R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *2010 IEEE symposium on security and privacy*, pp. 305–316, IEEE, 2010.
45. K. Xu, Z.-L. Zhang, and S. Bhattacharyya, "Profiling internet backbone traffic: behavior models and applications," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, pp. 169–180, 2005.
46. Z. Durumeric, M. Bailey, and J. A. Halderman, "An internet-wide view of internet-wide scanning," in *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pp. 65–78, 2014.
47. F. Iglesias, A. Hartl, T. Zseby, and A. Zimek, "Are network attacks outliers? a study of space representations and unsupervised algorithms," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 159–175, Springer, 2019.
48. M. N. Sakib and C.-T. Huang, "Using anomaly detection based techniques to detect http-based botnet c&c traffic," in *2016 IEEE International Conference on Communications (ICC)*, pp. 1–6, IEEE, 2016.
49. D. C. Le, A. N. Zincir-Heywood, and M. I. Heywood, "Data analytics on network traffic flows for botnet behaviour detection," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–7, IEEE, 2016.
50. M.-Y. Su, "Real-time anomaly detection systems for denial-of-service attacks by weighted k-nearest-neighbor classifiers," *Expert Systems with Applications*, vol. 38, no. 4, pp. 3492–3498, 2011.
51. M. Marchetti, F. Pierazzi, M. Colajanni, and A. Guido, "Analysis of high volumes of network traffic for advanced persistent threat detection," *Computer Networks*, vol. 109, pp. 127–141, 2016.
52. S. Bhatt, P. K. Manadhata, and L. Zomlot, "The operational role of security information and event management systems," *IEEE security & Privacy*, vol. 12, no. 5, pp. 35–41, 2014.
53. S. Axelsson, "The base-rate fallacy and its implications for the difficulty of intrusion detection," in *Proceedings of the 6th ACM Conference on Computer and Communications Security*, pp. 1–7, 1999.
54. P. C. van Oorschot, "Intrusion detection and network-based attacks," in *Computer Security and the Internet*, pp. 309–338, Springer, 2020.
55. M. Grill, T. Pevnỳ, and M. Rehak, "Reducing false positives of network anomaly detection by local adaptive multivariate smoothing," *Journal of Computer and System Sciences*, vol. 83, no. 1, pp. 43–57, 2017.
56. C. Guo, Y. Ping, N. Liu, and S.-S. Luo, "A two-level hybrid approach for intrusion detection," *Neurocomputing*, vol. 214, pp. 391–400, 2016.
57. A. Sperotto, M. Mandjes, R. Sadre, P.-T. de Boer, and A. Pras, "Autonomic parameter tuning of anomaly-based idss: an ssh case study," *IEEE Transactions on Network and Service Management*, vol. 9, no. 2, pp. 128–141, 2012.
58. R. A. A. Habeeb, F. Nasaruddin, A. Gani, I. A. T. Hashem, E. Ahmed, and M. Imran, "Real-time big data processing for anomaly detection: A survey," *International Journal of Information Management*, vol. 45, pp. 289–307, 2019.
59. J. O. Nehinbe, "A critical evaluation of datasets for investigating idss and ipss researches," in *2011 IEEE 10th International Conference on Cybernetic Intelligent Systems (CIS)*, pp. 92–97, IEEE, 2011.
60. A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *computers & security*, vol. 31, no. 3, pp. 357–374, 2012.

61. E. K. Viegas, A. O. Santin, and L. S. Oliveira, "Toward a reliable anomaly-based intrusion detection in real-world environments," *Computer Networks*, vol. 127, pp. 200–216, 2017.
62. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization.," in *ICISSP*, pp. 108–116, 2018.
63. I. Sharafaldin, A. Gharib, A. H. Lashkari, and A. A. Ghorbani, "Towards a reliable intrusion detection benchmark dataset," *Software Networking*, vol. 2018, no. 1, pp. 177–200, 2018.
64. M. Bermúdez-Edo, R. Salazar-Hernández, J. Díaz-Verdejo, and P. Garcia-Teodoro, "Proposals on assessment environments for anomaly-based network intrusion detection systems," in *International Workshop on Critical Information Infrastructures Security*, pp. 210–221, Springer, 2006.
65. G. F. Cretu, A. Stavrou, M. E. Locasto, S. J. Stolfo, and A. D. Keromytis, "Casting out demons: Sanitizing training data for anomaly sensors," in *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pp. 81–95, IEEE, 2008.
66. C. Guo, Y.-J. Zhou, Y. Ping, S.-S. Luo, Y.-P. Lai, and Z.-K. Zhang, "Efficient intrusion detection using representative instances," *computers & security*, vol. 39, pp. 255–267, 2013.
67. P. Velarde-Alvarado, C. Vargas-Rosales, R. Martinez-Pelaez, H. Toral-Cruz, and A. F. Martinez-Herrera, "An unsupervised approach for traffic trace sanitization based on the entropy spaces," *Telecommunication Systems*, vol. 61, no. 3, pp. 609–626, 2016.
68. C. Pascoal, M. R. De Oliveira, R. Valadas, P. Filzmoser, P. Salvador, and A. Pacheco, "Robust feature selection and robust pca for internet traffic anomaly detection," in *2012 Proceedings Ieee Infocom*, pp. 1755–1763, IEEE, 2012.
69. M. Mardani, G. Mateos, and G. B. Giannakis, "Dynamic anomalography: Tracking network anomalies via sparsity and low rank," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 1, pp. 50–66, 2012.
70. M. Grill and T. Pevnỳ, "Learning combination of anomaly detectors for security domain," *Computer Networks*, vol. 107, pp. 55–63, 2016.
71. R. Fontugne, P. Borgnat, P. Abry, and K. Fukuda, "MAWILab: Combining Diverse Anomaly Detectors for Automated Anomaly Labeling and Performance Benchmarking," in *ACM CoNEXT '10*, (Philadelphia, PA), December 2010.
72. P. Casas, F. Soro, J. Vanerio, G. Settanni, and A. D'Alconzo, "Network security and anomaly detection with big-dama, a big data analytics framework," in *2017 IEEE 6th International Conference on Cloud Networking (CloudNet)*, pp. 1–7, IEEE, 2017.
73. J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset shift in machine learning*. The MIT Press, 2009.
74. R. A. Maxion and K. M. Tan, "Benchmarking anomaly-based detection systems," in *Proceeding International Conference on Dependable Systems and Networks. DSN 2000*, pp. 623–630, IEEE, 2000.
75. M. Basseville, I. V. Nikiforov, *et al.*, *Detection of abrupt changes: theory and application*, vol. 104. prentice Hall Englewood Cliffs, 1993.
76. M. Du, F. Li, G. Zheng, and V. Srikumar, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1285–1298, 2017.
77. M. A. Rassam, M. A. Maarof, and A. Zainal, "Adaptive and online data anomaly detection for wireless sensor systems," *Knowledge-Based Systems*, vol. 60, pp. 44–57, 2014.
78. H. Raza, G. Prasad, and Y. Li, "Ewma model based shift-detection methods for detecting covariate shifts in non-stationary environments," *Pattern Recognition*, vol. 48, no. 3, pp. 659–669, 2015.
79. J. Zhang, H. Li, Q. Gao, H. Wang, and Y. Luo, "Detecting anomalies from big network traffic data using an adaptive detection approach," *Information Sciences*, vol. 318, pp. 91–110, 2015.

80. Y. Dong, Y. Zhang, H. Ma, Q. Wu, Q. Liu, K. Wang, and W. Wang, "An adaptive system for detecting malicious queries in web attacks," *Science China Information Sciences*, vol. 61, no. 3, p. 032114, 2018.
81. G. Maciá-Fernández, J. Camacho, R. Magán-Carrión, P. García-Teodoro, and R. Therón, "Ugr '16: A new dataset for the evaluation of cyclostationarity-based network idss," *Computers & Security*, vol. 73, pp. 411–424, 2018.
82. B. Mukherjee, L. T. Heberlein, and K. N. Levitt, "Network intrusion detection," *IEEE network*, vol. 8, no. 3, pp. 26–41, 1994.
83. G. M. D. T. Forecast, "Cisco visual networking index: Global mobile data traffic forecast update, 2017–2022," *Update*, vol. 2017, p. 2022, 2019.
84. D. Hoplaros, Z. Tari, and I. Khalil, "Data summarization for network traffic monitoring," *Journal of network and computer applications*, vol. 37, pp. 194–205, 2014.
85. E. Vasilomanolakis, S. Karuppayah, M. Mühlhäuser, and M. Fischer, "Taxonomy and survey of collaborative intrusion detection," *ACM Computing Surveys (CSUR)*, vol. 47, no. 4, pp. 1–33, 2015.
86. B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in internet of things," *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, 2017.
87. S. Suthaharan, "Big data classification: Problems and challenges in network intrusion prediction with machine learning," *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 4, pp. 70–73, 2014.
88. K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The hadoop distributed file system," in *2010 IEEE 26th symposium on mass storage systems and technologies (MSST)*, pp. 1–10, Ieee, 2010.
89. W. Feng, Q. Zhang, G. Hu, and J. X. Huang, "Mining network data for intrusion detection through combining svms with ant colony networks," *Future Generation Computer Systems*, vol. 37, pp. 127–140, 2014.
90. Y. Mirsky, A. Shabtai, B. Shapira, Y. Elovici, and L. Rokach, "Anomaly detection for smartphone data streams," *Pervasive and Mobile Computing*, vol. 35, pp. 83–107, 2017.
91. A. Ramamoorthi, T. Subbulakshmi, and S. M. Shalinie, "Real time detection and classification of ddos attacks using enhanced svm with string kernels," in *2011 International Conference on Recent Trends in Information Technology (ICRTIT)*, pp. 91–96, IEEE, 2011.
92. H. G. Kayacik and A. N. Zincir-Heywood, "Mimicry attacks demystified: What can attackers do to evade detection?," in *2008 Sixth Annual Conference on Privacy, Security and Trust*, pp. 213–223, IEEE, 2008.
93. P. Fogla, M. I. Sharif, R. Perdisci, O. M. Kolesnikov, and W. Lee, "Polymorphic blending attacks.," in *USENIX security symposium*, pp. 241–256, 2006.
94. I. Corona, G. Giacinto, and F. Roli, "Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues," *Information Sciences*, vol. 239, pp. 201–225, 2013.
95. M. Kloft and P. Laskov, "Online anomaly detection under adversarial impact," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 405–412, 2010.
96. G. Apruzzese and M. Colajanni, "Evading botnet detectors based on flows and random forest with adversarial samples," in *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*, pp. 1–8, IEEE, 2018.
97. S. Garcia, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *computers & security*, vol. 45, pp. 100–123, 2014.
98. F. Pierazzi, F. Pendlebury, J. Cortellazzi, and L. Cavallaro, "Intriguing properties of adversarial ml attacks in the problem space," *arXiv preprint arXiv:1911.02142*, 2019.