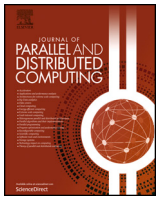
Journal of Parallel and Distributed Computing 196 (2025) 105010

Contents lists available at ScienceDirect

Journal of Parallel and Distributed Computing

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



Enabling semi-supervised learning in intrusion detection systems

Panagis Sarantos [a], John Violos [b,*], Aris Leivadeas [b]

[a] Damen Naval, Vlissingen, Netherlands
[b] Dept. of Software and IT Engineering, École de technologie supérieure, Montreal, Canada

A B S T R A C T

Intrusion Detection systems (IDS) are alerting cybersecurity tools that analyze network traffic in order to identify suspicious activity and known threats. State of the art IDS rely on supervised machine learning models which are trained to categorize the network flow with a historical labeled dataset. Nonetheless, next-generation networks are characterized as heterogeneous and dynamic. The heterogeneity can make every network environment to be significantly different and the dynamicity means that new threats are constantly emerging. These two factors raise the research question if a supervised machine learning based IDS can work efficiently in a network environment different from the one that generated its labeled training data. In this paper, we first give an answer to this research question and next try to propose a semi-supervised learning approach that can be generalized sufficiently in a different network environment using unlabeled data, taking into consideration that unlabeled data are much easier and cheap to be collected compared to labeled ones. In order to have a proof of concept we made experiments with two labeled datasets CIC-IDS2017, CIC-IDS2018 which are publicly available and one unlabeled dataset PS-Azure2023 which we constructed for this work and make it also publicly available. The results confirm our assumption and the applicability of the semi-supervised learning paradigm for the design of IDS.

1. Introduction

Cybersecurity is considered as a crucial part of distributed networks and computing systems. Attackers from all over the world are constantly pursuing to have unauthorized access in multiple and different network environments in order to steal, alter, expose, disable, or destroy data and applications. Cybersecurity experts design tools and apply practices to protect the users and reduce the network vulnerabilities. Even though they manage to keep secure the network environments for which they are responsible, their solutions can become fast outdated and inapplicable when the network infrastructure changes. This makes a necessity to design adaptive solutions that can be generalized sufficiently when new threats are emerged and when they are applied in new network environments. In this context a network environment is defined by the infrastructure including physical and virtual computing, storage, and communication devices. In addition, it includes the various services, protocols and applications it runs and also the users' behavior.

Cybersecurity can be defined as the software and the processes designed to protect computers including hardware, software, network and data from unauthorized access [1]. The unauthorized access is mostly connected with vulnerabilities exploited by cybercriminals, terrorist groups and malicious users (malicious hackers) over the Internet. It is important to note that hackers are not synonymous with cybercriminals. Hackers can be classified as ethical hackers, or "white hat" hackers, who conduct their activities legally. Conversely, there are malicious hackers, also known as "black hat" hackers, who operate with malicious intent. Furthermore, Cybersecurity extends beyond safeguarding computer systems to also encompass the protection of Internet-based digital equipment (or networks in general) and the associated data against unauthorized access and modification.

Accordingly, the rapid expansion of network technologies constantly makes cybersecurity more challenging as current solutions are far from being considered permanent. Hence, even though there is a continuous struggle to introduce various mechanisms and software tools to protect the network and the information exchanged, all of them provide a short-term protection [2]. This creates the need for a better understanding of how security mechanisms and appropriate strategies can aid in protecting computer systems, networks, programs, devices and data from cyberattacks on a longer term basis.

Cybersecurity mostly includes three processes: threat prevention, detection, and response [3]. In this paper we deal with threat detection and specifically with the design of a methodology to detect network attacks in-progress which is called intrusion detection. National Institute of Standards and Technology defines intrusion as an attempt to breach the confidentiality, integrity and availability of information or bypass the security processes in order to control a computer or network [4]. The process of capturing and analyzing events occurring on a computer system or network in order to find signs of intrusions is defined as in-

* Corresponding author.
E-mail addresses: p.sarantos@damennaval.com (P. Sarantos), ioannis.violos.1@ens.etsmtl.ca (J. Violos), aris.leivadeas@etsmtl.ca (A. Leivadeas).

https://doi.org/10.1016/j.jpdc.2024.105010
Received 11 October 2023; Received in revised form 25 September 2024; Accepted 7 November 2024
Available online 12 November 2024

Fig. 1. Workflow of our Research.

trusion detection. Similarly, an Intrusion Detection System (IDS) can be defined as the software or hardware system capable of detecting an intrusion in an automated way, facilitating and supporting the work of security professionals and community emergency response teams [5].

Specifically, IDS leverage machine learning (ML) models and more precisely classifiers that categorize the network activity into benign or malicious. In case the activity is flagged as malicious, the classifier will try to distinguish to which type it belongs, such as Denial-of-Service (DoS) attacks, SQL injections, etc. However, the particular ML models are trained on a specific network environment and cannot always be generalized. Hence, in this article, we examine a specific limitation of IDS that is related with its inability to generalize well in different network environments. This limitation is related to the difficulty to obtain a sufficient labeled dataset in order to train a machine learning based IDS. In order to fill this research gap we follow a methodology that collects unlabeled data from a vulnerable network setup also known as honeypot, we apply a semi-supervised machine learning technique to label them, and finally improve the performance of the IDS when working in new network environments.

1.1. Problem description

The focal point of this study is to explore the implications of shifting network environments on the performance of IDS. Traditionally, IDS are trained using supervised machine learning methods within a specific network context, leveraging local historical data [6]. However, the critical question arises: can these models maintain their effectiveness when deployed in different network environments? This inquiry is particularly significant as it delves into the challenge of generalization across diverse network domains [7], a key concern in both industry and cybersecurity.

Moreover, the endeavor to construct labeled datasets for each domain is not only resource-intensive but also time-consuming, posing a hurdle in the era of autonomous networks where efficient knowledge transfer is paramount [8]. In light of this, the research seeks to investigate how IDS models can adapt seamlessly across varied network landscapes with minimal retraining and label annotation efforts.

The empirical findings, as will be detailed in the experimental evaluation section, underscore the inadequacy of current IDS models in maintaining accuracy amidst changing network environments. This realization prompts the formulation of a secondary research question: how can the performance of IDS models be enhanced in response to changing network environments? To answer this question, our article explores how we can integrate a semi-supervised learning approach [9] into an IDS to enable adaptation across diverse domains [10].

1.2. Motivation & driving ideas

The above challenges and problems identified inspired us to propose a novel IDS system that leverages a semi-supervised learning instead of the traditional supervised ones used so far in the pertinent literature. The motivation behind enabling semi-supervised learning in IDS stems from the increasingly complex and dynamic nature of cybersecurity threats. Traditional supervised learning approaches in IDS heavily rely on labeled data for training, which can be scarce, expensive, and time-consuming to obtain. Additionally, labeled data may not adequately represent the evolving landscape of cyber threats, leading to limitations in the detection capabilities of IDS. Semi-supervised learning offers a promising solution by leveraging both labeled and unlabeled data, thereby enhancing the robustness and adaptability of intrusion detection mechanisms. By harnessing the abundance of unlabeled data typically available in network traffic, semi-supervised learning algorithms can effectively identify anomalous patterns indicative of potential security breaches. This approach aligns with the need for IDS to continuously evolve and adapt to emerging threats in real-time, ensuring proactive defense against sophisticated cyberattacks. Furthermore, another driving idea is the generalization of our approach, by allowing to create a highly dynamic and adaptive IDS model that can be generalized in different network environments and settings.

1.3. Research contributions

The latter driving idea provides our first contribution, which answers to the research question of whether a machine learning IDS model experiences performance degradation when deployed in a different testbed from the one its training data were generated. Through rigorous experimentation and evaluation across different network environments, we empirically demonstrate the impact of the deployment context on the IDS performance. Our findings reveal nuanced insights into the generalization capabilities of machine learning-based IDS, shedding light on the challenges and limitations inherent in traditional supervised learning approaches. Building upon these insights, we propose a methodology to improve the ability of IDS to generalize in a different network environment based on the semi-supervised learning paradigm. By leveraging both labeled data from the original training environment and unlabeled data from a second environment, our approach enhances the adaptability and robustness of IDS into a third deployment environment.

Furthermore, our research contributes to the advancement of IDS by addressing key challenges in adaptability to emerging threats and improved detection accuracy, through the use of a label propagation algorithm. Firstly, by integrating semi-supervised learning techniques into IDS, we empower security practitioners with more effective tools for proactive threat detection and response. Secondly, the ability of semi-supervised learning-based IDS to continuously learn from different network environments, even if the data instances are unlabeled, using a label propagation algorithm can identify the novel attack patterns and timely mitigate emerging cyber threats.

Finally, our last contribution facilitates further research and development in the field of IDS, by providing the research community with a new and real intrusion detection dataset. This dataset encompasses a diverse range of network traffic scenarios and attack vectors, serving as a valuable resource for benchmarking and evaluating the efficacy of intrusion detection algorithms across different network environments. Last but not least we should mention that the creation of datasets hold significant importance, particularly within the realm of cybersecurity. Datasets in this domain are often costly to generate and are seldom made publicly accessible. Our study underscores automatic data generation and annotation with human intervention. This approach essentially involves the process of labeling datasets, thereby expanding their utility and applicability. By demonstrating the viability of this method, we contribute to enhancing the accessibility and diversity of datasets available for cybersecurity research and analysis.

1.4. Structure of the article

Our research workflow is depicted in Fig. 1. In the first step, we identify the research problem that there is no model generalization in IDS

between different network environments. In the second step, we create the basic model suggested by the research [11] which is also trained with the dataset CIC-IDS2017 generated on the testbed of the authors of the article. We continue making experiments to verify the hypothesis that the IDS models do not generalize sufficiently by evaluating the basic model with the dataset CIC-IDS2018 provided by [12]. After confirming the hypothesis, we then propose a semi-supervised methodology in order to tackle the limitation of IDS to be sufficiently generalized. In order to apply the semi-supervise methodology we also construct the PS-Azure2023 dataset. In the next step, we train the baseline model using the semi-supervised methodology with the CIC-IDS2017 and PS-Azure2023 datasets. In the last step, by making the evaluation in the CIC-IDS2018 dataset, a significant improvement is observed during the cyberattack detection by the IDS.

The rest of the paper is structured as follows: Section 2 exposes the research gap identified within the existing literature concerning domain generalization in IDS, juxtaposed with our driving idea to employ a semi-supervised learning approach. In Section 3, we offer a detailed presentation of the datasets employed in our research, encompassing both the two publicly available datasets and a bespoke dataset constructed specifically for the purposes of this study. This section elucidates the rationale behind dataset construction in IDS and provides insights on how to tackle the problem of human label annotation. Section 4 delineates our proposed methodology based on our motivation and driving ideas, detailing the intricacies of our approach to integrating semi-supervised learning techniques into IDS and outlining the key steps involved in our methodology. Subsequently, Section 5 presents the comprehensive results of our performance evaluation, shedding light into the outcomes of applying our methodology to the datasets discussed in Section 3. This section not only stands as a proof of concept for our motivation and driving ideas but also offers a nuanced discussion and interpretation of the results. Finally, Section 6 offers a summative reflection on the merits of our work, while also identifying research avenues of alternative data driven solutions within the field of IDS such as the generation of synthetic data.

2. Related work & background

This section first introduces the state-of-the-art methods in subsection 2.1, followed by a discussion of how our work surpasses these methods in subsection 2.2. Next, we describe the background of our work in subsection 2.3, covering the topics of semi-supervised learning, the label propagation algorithm, decision trees, random forests, and the differences between decision trees and random forests in subsections 2.3.1, 2.3.2, 2.3.3, 2.3.4, and 2.3.5 respectively.

2.1. State-of-the-art methods

As stated in the previous section, IDSs are being used to protect servers, networks and users from any attack that could harm the confidentiality, integrity and the availability of them [13]. An IDS can be deployed as a hardware device or a software-based and virtualized function [14] with the purpose to identify threats that a normal firewall cannot. The three main categories that IDS fall in are the Signature-based Intrusion Detection System (SIDS), the Anomaly-based Intrusion Detection System (AIDS) and the data-driven intelligent IDS also named machine learning based IDS or ML-based IDS [6].

The SIDS, which is also known as Knowledge-based Detection or Misuse Detection, uses predefined signatures to raise an alert that something malicious has been spotted [15]. Signatures are, basically, human-crafted rules that identify a threat in a host or a network. These rules are elicited based on historical network activities and data related with a threat, or a family of threats. On the other hand, AIDS is designed to automatically identify anomalies on a network or host-based traffic [16]. AIDS tries to overcome the limitations that SIDS has, by establishing a model for the normal network behavior, using machine learning, statistical-based or knowledge-based methods and gauging the degree of irregularity for a new event. So, any major deviation from the normal behavior that is being established in the model, is considered as anomaly and possibly a threat or incident.

Statistical-based AIDS methods are using statistics to build a model for the normal behavior of the network or host. The statistical techniques are further categorized into univariate, multivariate and time-series based on the number of features and whether the values are time ordered [5]. On the other hand, knowledge-based methods establish their activity profiles based on human knowledge, in a set or rules that identify the normal behavior for the system. This method is good for reducing false positives [17], where false positives are security alerts that are misclassified as threat indications when none exist.

Machine learning techniques have been extensively used for data-driven intelligent IDS [18]. In order to extract knowledge from intrusion datasets, a variety of feature engineering techniques [19] algorithms and methodologies have been used, including decision trees - based [20], support vector machines [21], ensemble methods [22] and deep learning models [23]. Deep learning is one of the exciting techniques which recently are vastly employed by the IDS. Different types of networks such as Recurrent Neural Networks [24], Deep Convolutional Neural Network Bidirectional-Long-Short Term Memory [25] and the more advanced architecture of Transformers [26] have been employed.

Above mentioned Machine learning methods are mostly categorized in three main learning paradigms to build a model: supervised, unsupervised and semi-supervised learning [9]. Supervised learning-based AIDS methods identify threats by using a dataset that consists of labeled records [27]. In the dataset, each record is a row containing a number of data features that describe the record and the binary label “Benign” or “Malicious” in case the record concerns normal traffic activity or a cyberattack respectively. On the other side, unsupervised learning techniques do not require predefined labels for the records [28]. Instead, they discover hidden patterns, find structure and categorize the records into various groups using similarities and closeness feature metrics.

The semi-supervised learning paradigm uses a dataset with a mix of labeled and unlabeled records in order to build a descriptive or predictive model. This paradigm falls in-between the previous two and has many applications in today’s industry. Especially, in the cybersecurity field the labeled datasets are rare for two reasons [29]. Firstly, they are held by private corporations that do not make them publicly available since they may hold sensitive information. Secondly, the labelling of data records is an expensive task that requires meticulous work from expert annotators. Hence, it is difficult to find and build realistic datasets, since companies and enterprises do not often permit the collection of traffic samples from their network due to confidentiality and privacy issues. Additionally, national or international laws may be in effect hindering the data collection and thus publishing the actual data in public domains. To overcome such issues, usually, user related information is removed from the dataset to address the privacy issues [30].

In contrast, the gathering of unlabeled data is a cheap and automated process [31]. Accordingly, the semi-supervised learning paradigm can leverage a small labeled dataset and by propagating the known labels it can categorize the unlabeled records [32]. Aggregating the records from the labeled dataset and the output dataset of the semi-supervised method a larger labeled dataset can be produced. The only concern regarding the quality of the new dataset is that a percentage of the records that comes from the unlabeled dataset may be wrong. Thus, the evaluation of this percentage is not an easy task, since there is not a ground truth to compare against. Conversely, limitations such as missing or duplicate values and data outliers can be identified in both labeled and unlabeled datasets. This underscores the importance of data preprocessing for both types of datasets, a topic we will explore further in subsection 5.1. Generally, the primary difference between the two is the presence of label annotations on each observation.

Traditional SIDS bring the limitation of being dependant on the predefined signatures and for not having the capability of detecting new

and zero-days attacks [33]. On the other hand, AIDS, tries to mitigate this disadvantage but tend to trigger lots of false positive alarms [34]. This makes them too noisy and inconvenient for the security professionals. To tackle most of the above flaw, in today's industry, data-driven intelligent IDS has become the mainstream approach to follow.

2.2. Going beyond the state-of-the-art

It is obvious that outdated IDS models are unable to meet modern requirements in a world where cyberattacks are constantly evolving. Additionally, there are few labeled datasets available for cybersecurity needs, and the industry is unable to address this issue because the process of creating a dataset and the label annotation is very costly and time-consuming. Consequently, data-driven IDSs lag behind and are unable to adjust to the most recent and dynamic situations. To the best of our knowledge our work is the first-of-its-kind that tries to address and to provide a solution for each of these issues in such a holistic manner. Specifically, we show for the first time that an IDS's performance might degrade when its network environment is changed, and we propose a semi-supervised method that enables us to address the fundamental issue with today's data-driven IDSs by utilizing more, richer datasets that can improve the precision and effectiveness of these systems. In addition to resolving issues with older IDS systems, such as false positives for AIDS and SIDS's static nature, we are also able to sustain a potent data-driven IDS and ensure its reliable operation across various network settings and infrastructures. Furthermore, we also tackle the issue of labeled dataset scarcity in the cybersecurity domain. Our semi-supervised method allows us to create new labeled records from unlabeled data, expanding the size of the existing labeled datasets.

Additionally, our work not only addresses significant issues like the lack of publicly available cybersecurity datasets or the effectiveness of data-driver intrusion detection systems, but it also highlights critical elements in these AI models' performance that should be taken into account while conducting experiments. Finally, we firmly believe that the creative idea we developed for this work is easily expandable and adaptable to a variety of industries as it will be showcased in the rest of the paper.

2.3. Background

Previous research works have also made experimental comparisons among different machine learning models and concluded that random forest has the best performance and decision trees has the second best performance [11]. Both models are used as basic models to verify the applicability of the semi-supervised learning paradigm to tackle the generalization limitation of IDS. These models will be briefly presented in subsection 4.3. Additionally, our proposed methodology makes use of a specific semi-supervised machine learning method called Label Propagation Algorithm (LPA) which will be described in 2.3.2. The label propagation algorithm automatically assigns labels following an iterative process that propagates and updates labels of records based on the majority labels that are observed in other labeled datasets.

2.3.1. Semi-supervised learning

In machine learning, there is a distinction between supervised and unsupervised learning. In supervised learning, the set of data records are represented with feature vectors and categorized based on predefined labels. So the goal is to build an inference model that estimates the output label for previously unseen data records. In unsupervised learning, on the other hand, no specific output labels are provided. Instead, one tries to infer some underlying structures based on the feature vectors.

Semi-supervised learning is a branch of machine learning that aims to combine these two tasks. Typically, semi-supervised algorithms attempt to improve performance on one of these two tasks by using information generally related to the other. To address a classification problem, additional data with no labels can be used to aid in the classification process. For clustering methods, on the other hand, the learning process can be done from the knowledge that some records with the same label may share.

As with machine learning in general, the vast majority of research on semi-supervised learning focuses on classification. Semi-supervised classification methods are particularly relevant to scenarios where there is a lack of labeled data. IDS belongs to this type of scenarios since it is difficult to have labels regarding the cyberattacks. Additionally, as mentioned before, gathering unlabeled data is an easier process than labeled ones. However, if we make assumptions regarding the distribution of their labels, we can use them to improve the performance of a machine learning based IDS.

2.3.2. Label propagation algorithm

LPA is a fast algorithm for estimating coherent groups of data records using a graph representation model. The estimation of the groups takes place using only the graph structure as a guide and does not require a predefined objective function or prior information about the groups and their labels.

LPA works by propagating labels throughout the graph and forming groups based on the label propagation process. In our case, we represented chunks of the dataset as graphs with nodes to stand as the network flow records and edges to join the most similar records. The similarity is defined based on the features of the records and the radial basis function [35].

2.3.3. Decision trees

A decision tree [36], is a classifier expressed as a recursive partition of the occurrence space. It comprises nodes forming a rooted tree with a node known as the "root" and others as internal, and leaves, also termed as decision nodes. Each internal node divides the feature space based on a discrete function, typically considering single categorical features to partition the occurrence space. In the case of numerical features, the condition refers to a range. Leaves are assigned classes or probability vectors, and records are sorted by navigating from the root of the tree to a leaf, according to the result of the tests along the path. The complexity of the tree is measured by the total number of nodes, the total number of leaves, the depth of the tree, and the number of features used. In order to construct a decision tree from a given data set we use decision tree inducers which are mostly divided into top-down and bottom-up approaches with a clear preference in the literature for the first approach. There are various top-down decision tree inducers such as ID3, C4.5, CART [37]. Some consist of two conceptual phases: growth and pruning (C4.5 and CART). Other inducers perform only the growth phase. In our implementation we used CART since it has better results as we will see in the experimental evaluation.

2.3.4. Random forests

Random forests [38] is an ensemble modelling technique that consists of a series of tree classifiers. Each tree gives a unit vote for the most popular category, and then by combining these votes, the final classification result is produced. This results in a wide variety of decisions trees that generally leads to a better model. Therefore, in random forest, only a random subset of features is considered by the algorithm to split a node. In general, random forests have high classification accuracy, perform well in noisy data/outliers and show strong resistance against overfitting. Additionally, random forests have the advantage of adding extra randomness to the model while growing the trees. Instead of looking for the most important feature when splitting a node, they look for the best feature among a random subset of features. This is achieved by looking at how much the tree nodes using that feature reduce the impurity across all trees in the forest. The main limitation of random forest is that it involves a large number of trees which demands a greater allocation of computational resources.

2.3.5. Differences between decision trees & random forests

Random forests, unlike single decision trees, utilize a collection of decision trees, offering distinct characteristics. While decision trees establish rules based on training data to make predictions, random forest randomly selects records and features to build multiple trees, then averages the results. Notably, decision trees are prone to overfitting, which random forest mitigates by generating random feature subsets for smaller tree construction. However, this approach can slow calculations depending on the number of trees. Random forests have the advantage of higher accuracy and robustness. On the other hand, decision trees sometimes are preferable due to their simplicity, explainability, and interpretable decisions.

3. Datasets in IDS

Supervised machine learning based IDS require the training with a dataset in order to identify the cyberattacks. In our research we use some well-known datasets that have been extensively used for cybersecurity purposes. Specifically, we used three different datasets. Two of them, CIC-IDS2017 & CIC-IDS2018 (formally CSE-CIC-IDS2018), were created and disposed by the Canadian Institute for Cybersecurity (CIC) [39]. The CIC is based at the university of New Brunswick in Fredericton and brings together researchers and practitioners from across the academic spectrum to share innovative ideas, create disruptive technologies and conduct ground-breaking research on the most pressing cyber challenges. The third dataset is called PS-Azure2023. This dataset was created for the purposes of our research and specifically to serve the experimental requirements. The name came from the initial letters of the creator's and first author's name and because we monitored the Azure servers of Microsoft to acquire it.

We choiced CIC-IDS2017 and CIC-IDS2018 datasets because they have been established as benchmarks from CIC. These datasets can be used in our research due to their two crucial characteristics: both of them possess high standards of quality in terms of development methods and tools, despite being constructed in completely distinct network environments/testbeds. Both datasets encompass a wide variety of attack types and provide a robust network environment for data generation. Additionally, they effectively cover the vast majority of common protocols and services found in real-world scenarios, utilizing identical network data capture techniques and feature creation tools. In contrast, other publicly available IDS datasets exhibit significantly greater differences from one another in terms of attack types, including uncorrelated attacks and more pronounced distribution disparities. Utilizing CIC-IDS2017 and CIC-IDS2018 ensures the scientific integrity of our experiments and results. This is because the features of one dataset can be easily correlated with those of the other, enabling accurate evaluation of the performance differences of our machine learning models.

Regarding the labels of the datasets we distinguish two approaches: the binary-class dataset and the multiclass dataset. In the binary-class dataset we have two labels for the network flow the benign and the malicious for the normal traffic activity and the cyberattacks respectively. In case we have a multiclass dataset we have the benign label and seven different malicious labels based on the list of common attack types, which are described below [11]:

- **Brute Force Attack**: It is one of the most popular password cracking attacks. It is an exhaustive search of all possible combinations for password cracking. It can be used also for unveiling hidden content in a web page.
- **Heartbleed Attack**: This attack originates from a bug in the OpenSSL cryptography library, which is a widely used implementation of the Transport Layer Security (TLS) protocol. It is typically exploited by sending a malformed "heartbeat" request with small size disguised as a big size payload to the vulnerable server. The server needs to send back the exact size of the big payload and, thus memory data are added by mistake to the response that may contain sensitive information.
- **Botnet**: It contains a group of Internet-connected devices that the owner of the botnet uses for a variety of purposes and has complete control over. It can be used to send spam, steal data, and give an attacker access to the connected device.
- **DoS Attack**: With this type of attack, an attacker seeks to make a host or network resource temporarily unavailable. This can be accomplished by flooding the target with unnecessary requests in an attempt to overload the system, make it unavailable, and thus prevent normal traffic from being serviced.
- **DDoS Attack**: It often happens when numerous systems (such as botnets) overload a victim's bandwidth or resources saturating thus the targeted system with excessive amounts of network traffic.
- **Web Attack**: Web attacks cover a wide range of sub-attacks and involves everything that targets web pages and applications. The most popular ones are cross-site scripting (XSS), which occurs when developers fail to properly test their code to find the possibility of script injection, SQL Injection, which allows an attacker to create a string of SQL commands, and brute force over HTTP, which can try a list of passwords to find the administrator's password.
- **Infiltration attack**: Network infiltration usually exploits vulnerable software. A successful exploitation leads to installing a backdoor on the victim's computer that can perform different attacks on the victim's network, such as IP scanning, full port scanning, and service enumerations using Nmap. Nmap which stands for Network Mapper [40], is a free and open source popular tool used for network discovery and security purposes.

For both datasets, CIC reported that they used a profiling approach to generate the network flow in a systematic way including the following two separate categories of profiles:

- **B-Profiles**: This particular category of user profiles simulates the smooth network traffic that some users would have. The separation between benign and malicious traffic lies in the embedded features extracted from the network traffic. These are the distributions of a protocol's packet sizes, number of packets per flow, some patterns in the payload, payload size, and request time distribution of a protocol. The following protocols will be used for this simulation: HTTPS, HTTP, SMTP, POP3, IMAP, SSH and FTP.
- **M-Profiles**: On the other hand, this category refers to user profiles that simulate malicious traffic on the network and therefore the "user" performs malicious actions, attacks and general actions with the aim of damaging another machine.

3.1. CIC-IDS2017 dataset

The CIC-IDS2017 dataset [12] contains both benign traffic and the seven types of cyberattacks described before.

The CICFlowMeter [41] tool has been used to extract the features of network traffic reading the PCAP files that capture the packets of data transferred in the testbed. CICFlowMeter is a network traffic flow generator and analyzer tool. More than 80 features were extracted to represent the network flow records. These records as we will describe in the next section will be used to train a machine learning model to recognize the actual attacks and normal user behavior.

The testbed architectures consist of two completely separate networks, namely the victim-network and the attack-network. In the victim-network, the domain controller and the active directory connect three servers, one firewall, two switches, and ten desktops. Additionally, the victim-network's primary switch contains one port that has been set up as the mirror port, which captures all the traffic being sent and received by the network. The attack-network includes one router, one switch, and four workstations running the Kali and Windows 8.1 operating systems.

Table 1
CIC-IDS2017 Label Distribution.

Labels	#Records
BENIGN	2273097
DDoS	286957
Bot	1966
DoS GoldenEye	10304
DoS Hulk	231073
DoS Slowhttptest	5499
DoS slowloris	5796
SSH-Patator	5897
FTP-Patator	7938
WA-Brute Force	1507
WA-Sql Injection	21
WA-XSS	652
Infiltration	36

Table 2
CIC-IDS2018 Label Distribution.

Labels	Records
BENIGN	1685585
DDoS	157991
Bot	35773
DoS GoldenEye	5189
DoS Hulk	57739
DoS Slowhttptest	17486
DoS slowloris	1373
SSH-Patator	23448
FTP-Patator	24170
WA-Brute Force	76
WA-Sql Injection	10
WA-XSS	29
Infiltration	20242

CIC-IDS2017 has 2,830,743 records including 77 features, and one additional feature that is the label of the cyberattack. From the 13 labels in total, only one refers to 'BENIGN', which indicates the benign network activity, while the remaining fourteen are the various types of malicious activities, as shown in Table 1. In addition, a key observation drawn from Table 1 is that the distribution of the labels is unbalanced. This is evident from the fact that the class labeled 'BENIGN' occupies approximately 80% of the data set. This particular observation is important for our work as special treatment is required in datasets that are unbalanced.

3.2. CIC-IDS2018 dataset

The second dataset is the CIC-IDS2018. We used the particular dataset in order to assess the performance of the IDS. As we will explain in the next section we can assess how well a machine learning IDS model can generalize in a different network environment when training with CIC-IDS2017 and testing with CIC-IDS2018. We chose the CIC-IDS2018 because it has the same labels and it is consistent with the features of CIC-IDS2017. Table 2 summarizes the number of records for each type of attack.

Similar to the first dataset, the network environment that generated the CIC-IDS2018 dataset consists of an attack infrastructure and a victim organization. The attack infrastructure generates malicious traffic using 50 computers, whereas the victim organization has 5 departments and involves 420 computers and 30 servers. The dataset includes each machine's network traffic and system log files. The same raw packet information extraction tool, CICFlowMeter, has been used and the same 77 features have been extracted from the recorded traffic.

Table 3
"Honeypot" Machine Specifications.

OS	Ubuntu Focal - Linux
SKU	20_04-LTS
Type	Virtual Machine
Location	West Europe
Hostname	"honeymachine"
Admin Username	"azureuser"
Virtual CPUs	2
RAM	4GB
FQDN	honey.westeurope.cloudapp.azure.com
Private IP address	10.0.0.4

Table 4
"Honeypot" data captured.

Time Period	File (PCAP) Size	PCAP Lines	CSV Records
23/11/2022	92 MB	861.618	28561
24/11/2022	81 MB	1.206.501	52303
25/11/2022	67 MB	1.098.968	49500

3.3. Differences in the distribution of the datasets

There is a significant imbalance in both CIC-IDS2017 and CIC-IDS2018, with the "BENIGN" label having more than 80% of the observations. This is a significant problem since it complicates our job and affects how well the ML models function. On the other hand, this ratio is expected in real-world network flows, which can help the models become more resilient and ready for real-world situations.

Among the various attack types and labels presented, it is evident that the proportion of identical attack types across different datasets can fluctuate significantly. For instance, the label "Bot" appears 1,966 times in the CIC-IDS2017 dataset, whereas in the CIC-IDS2018 dataset, it exceeds 35,000 occurrences. These disparities are indicative of the class imbalance problem previously discussed and may influence the performance of our models. However, we address this issue with innovative techniques and new algorithms, as elaborated in subsection 5.2.1.

3.4. PS-Azure2023 dataset

Additionally, we created the PS-Azure2023 dataset for the needs of our research collecting network traffic data from Azure cloud servers. The specification of the servers are presented in Table 3, while the properties of the PCAP files are given in Table 4. In order to collect the data we used the honeypot mechanism. A cyber honeypot works by baiting a trap for malicious users. It is a server that sacrifices its resources and is intended to attract cyberattacks. It impersonates a target for malicious users and uses their intrusion attempts to gain information about cybercriminals and how they operate, or to distract them from other targets. Finally, we used the same feature set as the previous two papers. The dataset is available in the first author's GitHub repository.[1]

3.5. PS-Azure2023 dataset construction

The process of building an intrusion dataset is based on the collection of traffic samples from a real-world network. Both legitimate and malicious network flow are included in the PS-Azure2023 dataset. Normal user activity is the subject of normal traffic, but malicious activity includes various types of attacks such as DoS attacks and password cracking attempts on systems like SSH that need active user authentication. Because these machines have public IP addresses from a well-known address pool that Microsoft has reserved, we chose to set up a Virtual Machine (VM) on the well-known Azure cloud platform. Compared to a specific host from a much smaller provider that might have unique

[1] https://github.com/Panagiss/PS-Azure2023.

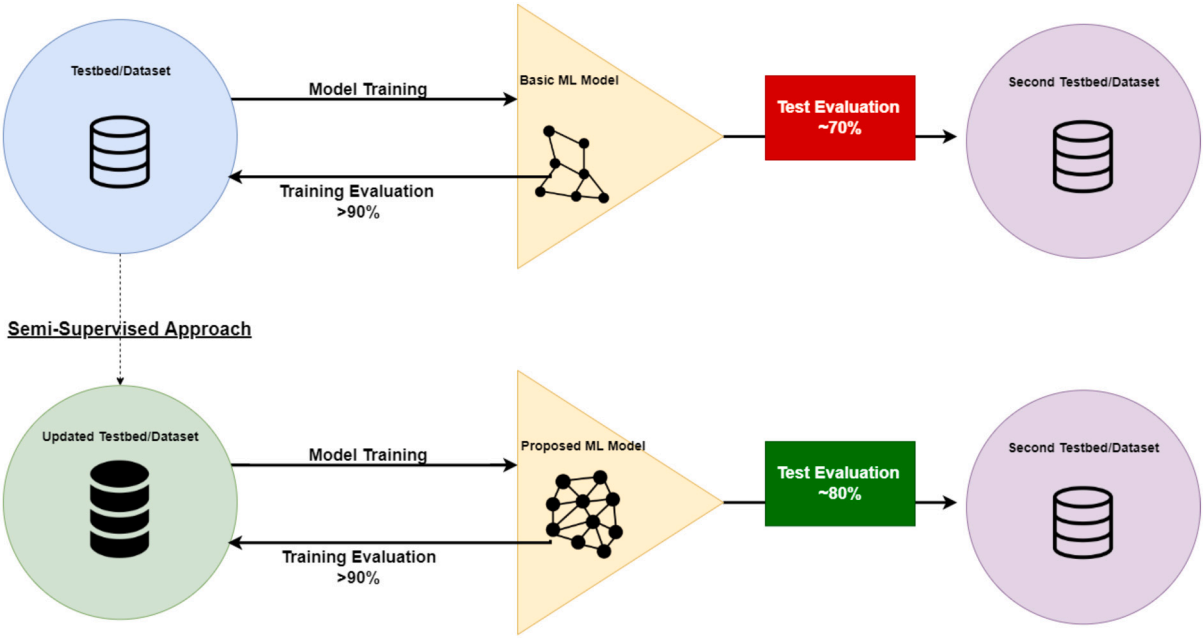


Fig. 2. Semi-Supervised Approach to Improve the Performance of Machine Learning based IDS.

perimeter defenses and architecture, many automated programs, bots, but also attackers, are far more likely to try harmful operations in VMs of major cloud providers with large and known address pool.

To entice further malicious users to attack the honeypot, we run many services on the side of these machines, with the ports of each service exposed and open for connections. Some of them are ssh, telnet, http web server, mysql database, among others. Any restriction that existed from the Azure firewalls was disabled so that the services were exposed and visible to malicious users.

Additionally, few services, including the MySQL database, were installed in our honeypot with vulnerable versions to entice attackers. Nonetheless, we tried to maintain an equilibrium in this situation since an attacker may become aware that their target host is a honeypot if too many susceptible programs and services are left running, and a honeypot should not reveal its true identity.

In order not to be solely dependant on automated attacks and malicious users, we generated our own malicious traffic by mimicking popular attack categories such as brute force attacks on ssh service and different variants of it such as account enumeration and password spraying. Additionally we performed advanced port scanning using NMAP [40] and different port scanning techniques. Although port scans may appear benign and insignificant, they often conceal underlying motives or potential security threats, necessitating thorough scrutiny and proactive measures. That is the reason why port scans represent precursors of a cyber attack. In cybersecurity and cyber-defense, precursors are elements of the incident identification and response process that allow both an attacker and a security researcher or professional to determine the existence of flaws and/or vulnerabilities within a host.

For the recording of the network flow we used tcpdump and CICFlowMeter tools on the remote servers. Tcpdump is a network data capture and protocol analysis tool. This program is based on the libpcap interface, a portable system-independent interface for capturing user-level network packets. Additionally, by using the CICFlowMeter we created bidirectional flows, where the first packet specifies the forward (source to destination) and the second specifies the reverse (destination to source) direction. Thus, more than 80 network traffic statistics such as duration, number of packets, number of bytes, the length of packets, etc. are recorded in the forward and backward directions. These records are stored in the PCAP files presented in Table 4.

As we have explained before, PS-Azure2023 is an unlabeled dataset in contrast to the CIC-IDS2017 and the CIC-IDS2018, which include labels regarding whether they are benign or malicious and the type of the IDS attacks. The construction of an unlabeled IDS dataset is a much easier process that can take place in any infrastructure but without knowing whether an attack takes place or identifying its type.

4. Proposed methodology

Previous works [11,12] have shown that having a labeled training dataset from a network environment-A, a ML based IDS can be built that can achieve very high performance when it runs on the same environment. In the rest of this article, such an IDS will be called as baseline model. However, when the baseline model is deployed and evaluated in a different network environment-B, a significant performance degradation in the detection of cyberattacks is noticed. This is depicted in the upper part of Fig. 2. Thus, our proposed methodology uses a third unlabeled dataset from an environment-C, different from the environment-A (used for training) and environment-B (used for testing). The unlabeled dataset-C in combination with the labeled dataset-A is leveraged in order to train an enhanced IDS machine learning model that performs better than the baseline model when it runs in the network environment-B. We can see this process in the bottom part of the Fig. 2.

For the initial evaluation of the baseline model we employed two state of the art methods, the decision trees and random forest, which we already described in the subsections 2.3.3 and 2.3.4 respectively. Previous research studies have also done experimental comparisons among different machine learning models and determined that random forest has the highest performance and decision trees has the second best performance [12]. The objective of this research is not to introduce new custom algorithms, although there may be some in the future. Instead, the focus is on highlighting the significant issue of domain generalization, which is often overlooked or unknown to many professionals. Additionally, this research aims to propose a solution to this problem by utilizing a Semi-supervised Machine Learning model as described in subsection 2.3.1.

4.1. The workflow

The pipeline of the semi-supervised methodology is depicted in Fig. 3. In step 1 of the pipeline, we monitor the network environment

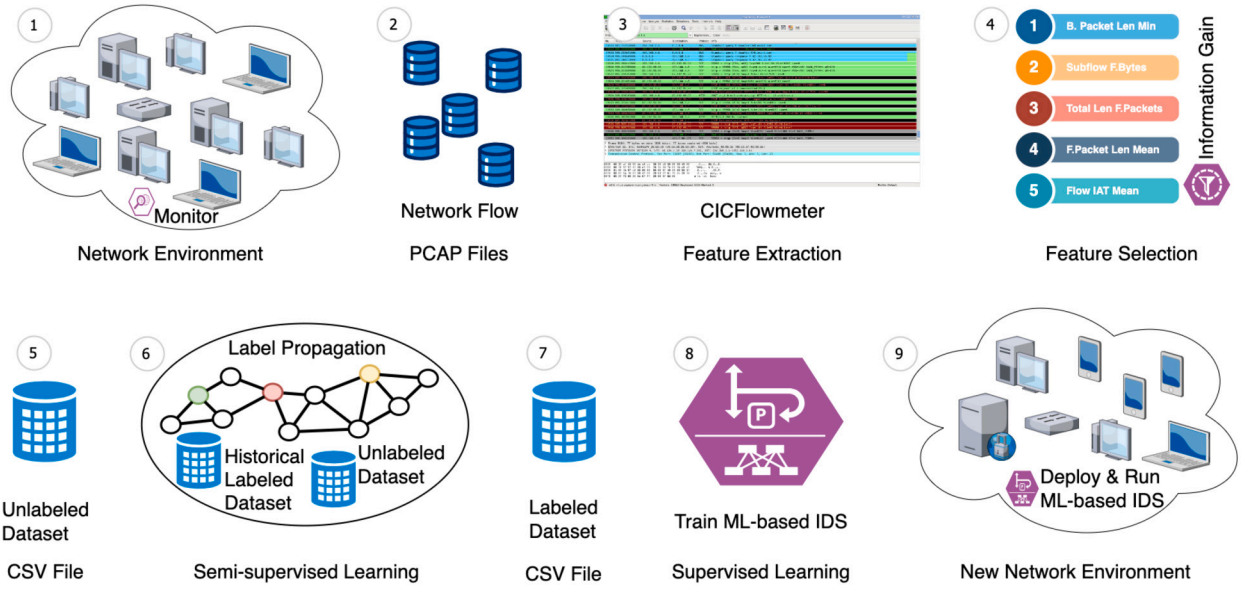


Fig. 3. Pipeline of nine steps for building a ML-based IDS following the Semi-supervised approach.

and save the network flow in the PCAP files of step 2. These steps are described in subsection 3.5. In step 3 we extract the network flow features using the CICFlowmeter and in step 4 we apply the information gain feature selection algorithm to select the most important features. These steps are described in subsection 4.2. The output is an unlabeled dataset stored in a CSV file as depicted in step 5. The step 6 is the core of the semi-supervised methodology, described in subsection 4.4. In this step, using the LPA and a historical labeled dataset we insert labels into the unlabeled dataset. The outcome of the LPA is the labeled dataset depicted in step 7. Using this new labeled dataset we can train a decision tree or random forest model as it is described in subsection 4.3 and depicted in step 8. Eventually in step 9, we deploy the machine learning based IDS in a new network environment. It is important to mention that the semi-supervised learning paradigm does not replace the supervised learning. Instead it adds the extra steps in order to enrich the historical labeled dataset with data records from different network environments.

4.2. Data features & feature selection

A data-driven methodology that works with different datasets should keep the same data model. In this way, we designed the data records, also known as feature vectors of PS-Azure2023 to have the same feature names, types and units with the CIC-IDS2017 and CIC-IDS2018 datasets. It should be also taken into consideration that machine learning algorithms work with numerical values. Thus, the features that represent string categorical variables are converted into numerical variables.

The papers that introduced the CIC-IDS2017 and CIC-IDS2018 datasets had used feature selection techniques in order to select the most critical features, eliminate the redundant ones and increase the prediction power of the algorithms. In PS-Azure2023, we also applied the results of the feature selection techniques in order to be consistent with the previous datasets. Generally, a dataset with many dimensions poses several difficulties in the analysis and organization of the data by the machine learning models. This phenomenon is termed as curse of dimensionality [42] and can be mitigated using a feature selection technique.

At this point, it should be noted that the CICFlowMeter tool can extract more than 80 features, but in our work, as well as in the preceding two research works [11] [12], we utilized the identical set of 77 features to elucidate the network flow. The original 80 features along with their descriptions are located in the tool's public repository [43]. CICFlowMeter's extracted features give a broad statistical explanation of a network flow. Furthermore, from counting the different flags found in the packets (FIN, SYN, RST, ACK, etc.) we identify the min, max, std and mean sent times between two packets. Additionally, each of these features corresponds to a particular flow direction, so that a feature for both the forward and backward directions of network movement is always present. We also used an additional feature, the 78th feature, which indicates whether the particular record was malicious or not, and if so, what attack type has being carried out. The rest of the features beyond these 78 are more explanatory for human understanding, such as: the time, the IP and port of entry and exit, but we decided not to include them for consistency reasons.

Taking into consideration the large number of features that the CICFlowMeter outputs and the previous research works, we used the information gain feature selection metric [44] and we scored the importance of each feature with a weight. Eventually only the 22 features with the highest score are used for the machine learning - based IDS. Table 5 presents these 22 most important features to detect the different types of cyberattacks and their importance score.

4.3. Baseline model

As an initial stage in our research, we create the baseline model by employing supervised learning methods, specifically decision trees and random forest, utilizing the CIC-IDS2017 dataset. The classification of network activity into benign and malicious is a categorization problem. In this scenario, our model will be required to assign labels or classify the new records according to the knowledge and training it has acquired. The training of models is a crucial step in supervised learning. It allows us to assess the predictive capabilities of each model and provides an indicator of their performance.

However, as we initially observed and emphasized, it is essential to assess the accuracy of these models using new, unfamiliar data or in a distinct environment. The final assessment of these models will be conducted on the CIC-IDS2018 dataset, which our models have not encountered before. By observing the eventual decline in performance when transitioning from a familiar to an unfamiliar model environment, we may validate our assumption.

4.4. Proposed model

Our approach begins with the labeling of the PS-Azure2023 using the semi-supervised learning paradigm described in subsection 2.3.1. This

Table 5
Feature Selection from CIC-IDS2017.

Label	Feature	Weight
Benign	B.Packet Len Min	0,0479
	Subflow F.Bytes	0,0007
	Total Len F.Packets	0,0004
	F.Packet Len Mean	0,0002
DoS GoldenEye	B.Packet Len Std	0,1585
	Flow IAT Min	0,0317
	Fwd IAT Min	0,0257
	Flow IAT Mean	0,0214
Heartbleed(DoS)	B.Packet Len Std	0,2028
	Subflow F.Bytes	0,1367
	Flow Duration	0,0991
	Total Len F.Packets	0,0903
DoS Hulk	B.Packet Len Std	0,2028
	B.Packet Len Std	0,1277
	Flow Duration	0,0437
	Flow IAT Std	0,0227
DoS Slowhttp	Flow Duration	0,0443
	Active Min	0,0228
	Active Mean	0,0219
	Flow IAT Std	0,0200
DoS slowloris	Flow Duration	0,0431
	F.IAT Min	0,0378
	B.IAT Mean	0,0300
	F.IAT Mean	0,0265
SSH-Patator(Brute Force)	Init Win F.Bytes	0,0079
	Subflow F.Bytes	0,0052
	Total Len F.Packets	0,0034
	ACK Flag Count	0,0007
FTP-Patator(Brute Force)	Init Win F.Bytes	0,0077
	F.PSH Flags	0,0062
	SYN Flag Count	0,0061
	F.Packets/s	0,0014
Web Attack	Init Win F.Bytes	0,0200
	Subflow F.Bytes	0,0145
	Init Win B.Bytes	0,0129
	Total Len F.Packets	0,0096
Infiltration	Subflow F.Bytes	4,3012
	Total Len F.Packets	2,8427
	Flow Duration	0,0657
	Active Mean	0,0227
Bot	Subflow F.Bytes	0,0239
	Total Len F.Packets	0,0158
	F.Packet Len Mean	0,0025
	B.Packets/s	0,0021
PortScan	Init Win F.Bytes	0,0083
	B.Packets/s	0,0032
	PSH Flag Count	0,0009
DDoS	B.Packet Len Std	0,1728
	Avg Packet Size	0,0162
	Flow Duration	0,0137
	Flow IAT Std	0,0086

will develop a new model that follows identical training and evaluation processes as the baseline one. We utilized the LPA as described in the subsection 2.3.2 to assign labels in PS-Azure2023.

To apply the LPA for labeling the PS-Azure2023 dataset, we initially partition the CIC-IDS2017 and PS-Azure2023 datasets into smaller segments as shown in Fig. 4. These segments are incrementally aggregated, and LPA is applied iteratively. Initially, the data segments include 80% labeled records from CIC-IDS2017 and 20% unlabeled records from PS-Azure2023. As the labeling process advanced, the proportion of data from PS-Azure2023 is progressively increased. Following each addition of PS-Azure2023 segments, LPA is reapplied to label the newly introduced unlabeled records. This iterative process continues until the entire PS-Azure2023 dataset is labeled and integrated with CIC-IDS2017. The outcome is a newly labeled dataset that amalgamates the two original datasets, which can subsequently be utilized to train machine learning-based IDS. The percentages we used for labeled and unlabeled records in the LPA model can be 80% labeled and 20% unlabeled. We concluded these numbers based on our experiments. In addition, we have seen that a higher proportion of unlabeled records make the convergence of the LPA model more challenging and less effective. Furthermore, it is critical to ensure that the segments from CIC-IDS2017, constituting 80% of the data used in the LPA model, are evenly distributed across different attack types, thereby maintaining the scientific rigor of the experiment.

Using the LPA, a single label can rapidly become predominant within a densely connected cluster of nodes, but it will have a low likelihood of traversing a sparsely connected region. Consequently, labels tend to become confined within densely connected clusters, and nodes that ultimately share the same label upon the algorithm's completion can be considered part of the same group. This process is illustrated in Fig. 4, where it is evident that the groups have converged, resulting in consistent labeling of coherent records. Coherence in this context is determined based on the proximity of feature vectors. Another significant characteristic of LPA is its ability to assign preliminary labels to nodes, which can constrain the range of solutions produced and facilitate the algorithm's convergence. This capability enables LPA to function as a semi-supervised method for identifying groups, utilizing knowledge derived from previously labeled datasets.

At this point we should clarify that the goal of our research is to propose and evaluate the semi-supervised learning paradigm in order to tackle the limitation that the machine learning based IDS does not generalize sufficiently in new network environments. It is not our goal to make an extended research on the performance of the various available machine learning models for IDS. In case the readers choose a different machine learning model such as a new flavor of deep learning model, they can apply the same proposed methodology in a similar way.

4.5. Effectiveness of proposed semi-supervised approach

Numerous studies have demonstrated that tree-based classification methods are highly effective in detecting intrusions and malicious activities [20]. This effectiveness stems from the ability of machine learning models to identify network usage patterns associated with cyberattacks, based on the historical data they are trained on. While these supervised models perform with high accuracy when evaluated within the same network environment as their training data, they tend to fail when applied in different environments. We experimentally demonstrate this issue in Section 5. To address this issue, the semi-supervised approach enhances the training labeled dataset by incorporating unlabeled data from various network environments. Collecting such unlabeled data is straightforward, as it only requires basic network monitoring and does not necessitate expert annotation. By incorporating data from diverse network environments, our approach improves the generalizability of IDS to new environments, even when annotated historical data is unavailable for training.

5. Experimental evaluation

The proposed methodology has been tested and evaluated with the datasets described in section 3. The CIC-IDS2017 was used in the training of the supervised learning approach, the combination of CIC-IDS2017 and PS-Azure2023 in the training of the semi-supervised and the CIC-IDS2018 for testing both approaches. Furthermore, the random forests and decision trees based IDS were built using the features described in subsection 4.2. Finally, all models were developed with Python 3 using the frameworks NumPy, Pandas and Scikit-learn, while the experiments were executed in the notebook Jupyter by Google Collaboratory.

5.1. Preprocessing

Data prepossessing took place in order to transform the records as described in subsection 4.2. The first step was to correct the names of

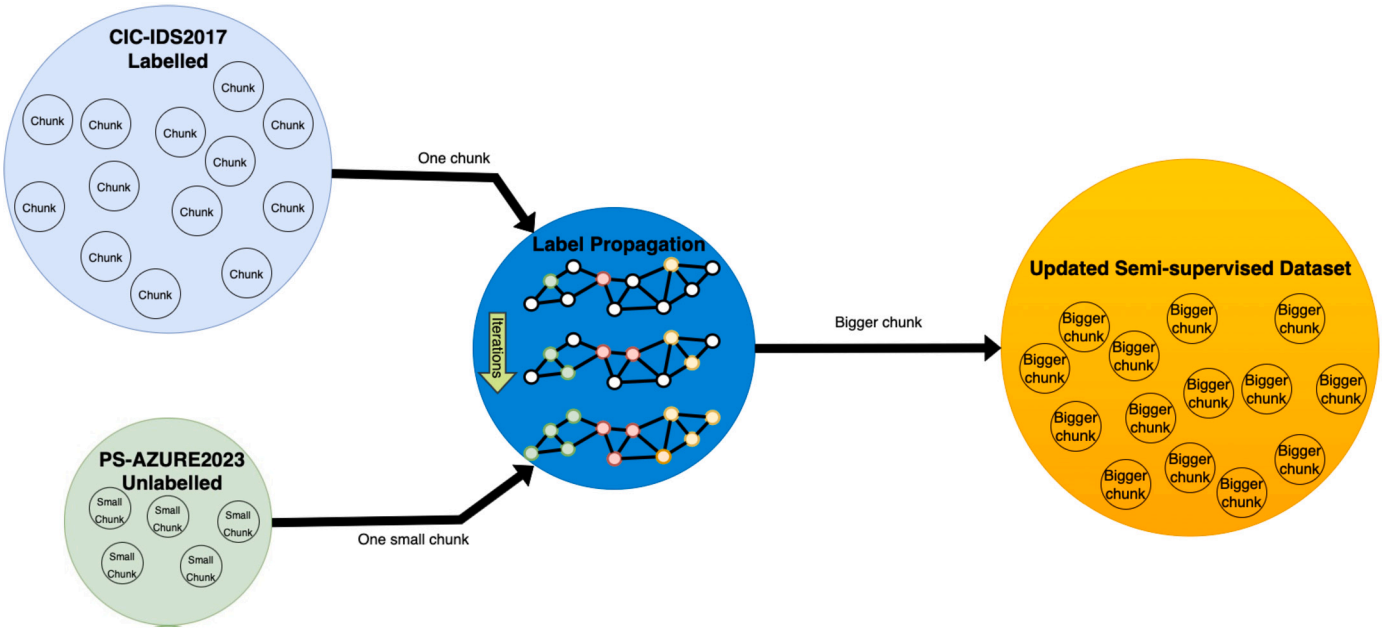


Fig. 4. Leveraging Label Propagation Algorithm for Semi-supervised Dataset.

Table 6
Binary-Decision Trees: Evaluating on CIC-IDS2017.

Percentage used CIC-IDS2017	Number of Records	Supervised ML Trained: CIC-IDS2017				Semi-supervised ML Trained: CIC-IDS2017 & PS-Azure2023			
		Acc. %	Prec. %	Rec. %	F1 %	Acc. %	Prec. %	Rec. %	F1 %
100%	2.8m	99.87	99.87	99.87	99.87	99.87	99.87	99.87	99.87
0.0170%	50k	99.56	99.56	99.56	99.56	99.89	99.89	99.89	99.89
0.0140%	40k	99.53	99.53	99.53	99.53	99.89	99.89	99.89	99.89
0.0030%	10k	98.15	98.15	98.15	98.15	99.94	99.94	99.94	99.94

Table 7
Multiclass-Decision Trees: Evaluating on CIC-IDS2017.

Percentage used CIC-IDS2017	Number of Records	Supervised ML Trained: CIC-IDS2017				Semi-supervised ML Trained: CIC-IDS2017 & PS-Azure2023			
		Acc. %	Prec. %	Rec. %	F1 %	Acc. %	Prec. %	Rec. %	F1 %
100%	2.8m	99.83	99.83	99.83	99.83	99.84	99.84	99.84	99.84
0.0170%	50k	99.54	99.54	99.54	99.54	99.87	99.87	99.87	99.87
0.0140%	40k	99.44	99.44	99.44	99.44	99.87	99.88	99.87	99.87
0.0030%	10k	98.82	98.90	98.82	98.83	99.91	99.91	99.91	99.91

the features so that they would be in line with the CICFlowMeter's official list [41]. Next followed the deduplication of the records, whereas records with missing values and outliers were discarded. Furthermore, the feature selection was performed by verifying and cross-checking our outcomes using the information gain metric with the outputs of the article [45]. The attained output corroborates the results of the article and of Table 5.

After the preprocessing and feature selection the CIC-IDS2017 dataset had 2,830,743 records including 22 features and one more feature regarding the cyberattack label. Similarly, CIC-IDS2018 Dataset had 2,029,111 records, the same 22 features and one more for the cyberattack label. Lastly, PS-Azure2023 had 130,360 records also including the same 22 features, while the cyberattack label will be assigned by the label propagation algorithm.

5.2. Outcomes & discussion

In this subsection we present the experimental outcomes using the supervised learning paradigm which stands as the basic model compared to the semi-supervised learning paradigm which is the proposed model. The outcomes are summarized in Tables 6 - 13. The tables except from the comparison between the basic model and the proposed model also present the results of models that were created on different sizes of the labeled data. Given that the PS-Azure2023 dataset includes 130,360 records, it is important to see the results in the evaluation metrics when we combine it with different sizes of labeled data. In Tables 6 - 13, the first column presents the percentage used of CIC-IDS2017 and the second column the number of records in descending order and in a range from 10k to 2.8m. To construct these parts of the dataset, we sampled the CIC-IDS2017 records keeping the same distribution of benign and different types of malicious attacks.

Furthermore there is one table for each training method used, one for decision trees and one for random forest, and for each method there are two tables for the different classification problems, binary and multiclass, as discussed in the previous section. Tables 6 to 9 compare the proposed model against the baseline in the records of CIC-IDS2017. Tables 10 to 13 compare the proposed model against the baseline in the records of CIC-IDS2018. So we have eight different tables in total. In

Table 8
Binary-Random Forests: Evaluating on CIC-IDS2017.

Percentage used CIC-IDS2017	Number of Records	Supervised ML Trained: CIC-IDS2017				Semi-supervised ML Trained: CIC-IDS2017 & PS-Azure2023			
		Acc. %	Prec. %	Rec. %	F1 %	Acc. %	Prec. %	Rec. %	F1 %
100%	2.8m	99.88	99.88	99.88	99.88	99.88	99.88	99.88	99.88
0.0170%	50k	99.68	99.68	99.68	99.68	99.91	99.91	99.91	99.91
0.0140%	40k	99.65	99.65	99.65	99.65	99.91	99.91	99.91	99.91
0.0030%	10k	98.08	98.08	98.08	98.07	99.96	99.96	99.96	99.96

Table 9
Multiclass-Random Forests: Evaluating on CIC-IDS2017.

Percentage used CIC-IDS2017	Number of Records	Supervised ML Trained: CIC-IDS2017				Semi-supervised ML Trained: CIC-IDS2017 & PS-Azure2023			
		Acc. %	Prec. %	Rec. %	F1 %	Acc. %	Prec. %	Rec. %	F1 %
100%	2.8m	99.85	99.84	99.85	99.84	99.86	99.85	99.86	99.85
0.0170%	50k	99.65	99.64	99.65	99.64	99.90	99.90	99.90	99.90
0.0140%	40k	99.60	99.57	99.60	99.58	99.90	99.90	99.90	99.90
0.0030%	10k	98.23	98.12	98.23	98.13	99.94	99.93	99.94	99.93

Table 10
Binary-Decision Trees: Evaluating on CIC-IDS2018.

Percentage used CIC-IDS2017	Number of Records	Supervised ML Trained: CIC-IDS2017				Semi-supervised ML Trained: CIC-IDS2017 & PS-Azure2023			
		Acc. %	Prec. %	Rec. %	F1 %	Acc. %	Prec. %	Rec. %	F1 %
100%	2.8m	82.43	74.42	82.43	75.95	82.52	74.47	82.52	78.28
0.0170%	50k	82.17	78.31	82.17	79.24	85.65	84.56	85.65	84.94
0.0140%	40k	83.15	79.36	83.15	79.63	83.66	81.86	83.66	82.46
0.0030%	10k	82.10	77.35	82.10	78.31	82.49	77.75	82.49	78.39

Table 11
Multiclass-Decision Trees: Evaluating on CIC-IDS2018.

Percentage used CIC-IDS2017	Number of Records	Supervised ML Trained: CIC-IDS2017				Semi-supervised ML Trained: CIC-IDS2017 & PS-Azure2023			
		Acc. %	Prec. %	Rec. %	F1 %	Acc. %	Prec. %	Rec. %	F1 %
100%	2.8m	82.75	70.14	82.75	75.39	82.80	70.34	82.85	75.78
0.0170%	50k	79.13	71.10	79.13	74.78	82.15	74.77	82.15	78.25
0.0140%	40k	79.62	70.94	79.62	74.98	80.02	71.57	80.02	75.52
0.0030%	10k	79.27	71.57	79.27	75.21	81.27	79.27	81.27	78.21

Table 12
Binary-Random Forest: Evaluating on CIC-IDS2018.

Percentage used CIC-IDS2017	Number of Records	Supervised ML Trained: CIC-IDS2017				Semi-supervised ML Trained: CIC-IDS2017 & PS-Azure2023			
		Acc. %	Prec. %	Rec. %	F1 %	Acc. %	Prec. %	Rec. %	F1 %
100%	2.8m	84.13	81.32	84.13	79.79	84.13	81.33	84.13	79.79
0.0170%	50k	82.84	79.84	82.84	80.58	83.25	79.94	83.25	81.56
0.0140%	40k	83.04	79.81	83.04	80.42	83.18	79.89	83.18	81.50
0.0030%	10k	81.49	77.70	81.49	78.84	81.53	78.10	81.73	79.87

Table 13
Multiclass-Random Forest: Evaluating on CIC-IDS2018.

Percentage used CIC-IDS2017	Number of Records	Supervised ML Trained: CIC-IDS2017				Semi-supervised ML Trained: CIC-IDS2017 & PS-Azure2023			
		Acc. %	Prec. %	Rec. %	F1 %	Acc. %	Prec. %	Rec. %	F1 %
100%	2.8m	82.06	72.81	82.06	76.04	82.81	72.90	82.81	77.53
0.0170%	50k	79.20	71.46	79.20	75.05	79.52	72.19	79.52	75.67
0.0140%	40k	78.45	71.32	78.45	74.68	79.04	71.53	79.04	75.03
0.0030%	10k	78.41	71.00	78.41	74.74	81.07	71.79	81.07	76.14

every table we have two groups of evaluation metrics, the first group concerns the evaluation outcomes of the supervised machine learning approach and the second group concerns the evaluation outcomes of the semi-supervised machine learning approach.

Since IDS datasets are unbalanced sets, we used K-Fold Cross Validation with weighted average of accuracy, precision, recall and F-measure metrics [46]. In this context unbalanced dataset means that the records are not uniformly distributed to the classes. As example, most records are labeled as benign and a small percentage of records belongs to SQL Injection label. K-Fold Cross Validation is an evaluation method that splits a dataset K times into a training part and a testing part. Training split consists of (K-1)/K records. While testing part consists of 1/K records. Accuracy shows the overall percentage of correct classifications, while Precision expresses the percentage of correct class predictions. Additionally, Recall expresses the percentage of relevant records that identified correctly for every class. Finally, F1-score is a weighted average of precision and recall. We used 'K' equal to 5 in all the following experiments. In this way, the dataset will be split for 5 times into two different parts with 80% of records for training and 20% of records for testing. This gives us better confidence on our results as the dataset has been split 5 times differently. Furthermore, because our dataset has multiple classes, we use the weighted average evaluation metrics which takes class imbalance into account by weighting the classes based on their presence in the dataset records. The cyberattacks are represented by the minority classes that have very few records. Since these few records are the most rare and the most important, we weight them more.

The primary training dataset, CIC-IDS2017, contains an unusually large number of data instances, which is not representative of industry use cases. Most private-sector businesses will not undertake the very costly and time-consuming process of label annotation for such large datasets. Therefore, we aimed to use a number of data instances in our tests that was similar to actual cybersecurity industry use cases. Thus, we conducted experiments by reducing the number of training data instances to below 100k. It is also important to note that in every sub-dataset, we maintained the same label distribution, and in the testing stage, we used the entire CIC-IDS2017 dataset.

The use of different thresholds for selecting training data instances from the CIC-IDS2017 dataset demonstrates that the proposed methodology is effective regardless of the training dataset size. This means that if we have a small training dataset with only 10k instances, the semi-supervised learning method will introduce a comparatively large amount of new labeled data instances, which improves the performance of the ML model. In a similar way, if we have 2.8m training data instances, the new labeled instances will be comparatively few but they still provide useful knowledge for training the ML model and improves its performance.

The PS-Azure2023 dataset was not used in the experiments with the supervised ML approach but was utilized in all experiments with the semi-supervised approach. The original PS-Azure2023 dataset is unlabeled, so it cannot provide additional knowledge to improve a supervised ML model. However, when we apply LPA to the PS-Azure2023 dataset, it becomes a new source of knowledge that can enhance the performance of the semi-supervised ML model.

5.2.1. IDS using the supervised learning paradigm

We trained and evaluated binary and multiclass decision trees and random forest based IDS models with the CIC-IDS2017. The outcomes are summarized from Table 6 to 9 in the columns Supervised ML. In almost all tests the accuracy, precision, recall and F1-score are higher than 99%. This confirms that a supervised machine learning model has very good performance when it is trained and evaluated in the same network environment. It is worth pointing out that we managed to achieve higher scores than the original paper [11] both for decision trees and random forest. That can be explained by the fact that the authors of the original work were running their experiments on Weka library [47] while we used Scikit-learn [48]. In addition, from their paper it is clear that they used the ID3 algorithm for Decision Trees while we used the CART algorithm that Scikit-learn provides, which is a successor to ID3.

Next, we trained binary and multiclass decision trees and random forest based IDS with the CIC-IDS2017 and evaluated with the CIC-IDS2018. The outcomes are summarized in the columns Supervised ML from Table 10 to 13. In this case we see that the accuracy ranges from 79.13% to 84.13%, precision 70.14% to 79.85%, recall from 78.45% to 83.15% and F1-score from 74.78% to 80.58%. These experimental outcomes show that the machine learning based IDS have a significant performance degradation when they run in a different infrastructure from the one they trained. This confirms our hypothesis that the supervised machine learning based IDS cannot generalize sufficiently in different network environments.

A machine learning based IDS that is capable of detecting multiple classes of attacks is based on a more complex model and it is harder to be trained compared to its binary counterpart. This explains the fact that we see higher scores across all evaluation metrics for binary decision trees and random forest based IDS. In cybersecurity, there are use cases in which it is enough to know if the network actions concern benign or malicious actions. In this case, we will prefer the binary classification approach. In other use cases a further categorization on the type of the malicious activity is important and should prefer a multiclass classification approach.

Given that the human annotation of datasets is an expensive process, we want to see the number of records required in order to achieve a high performance. It is worth mentioning that ten thousands of records are enough to achieve higher than 98% scores for all the evaluation metrics for the binary and the multiclass approaches when the training and the evaluation take place in the same network environment. Marginal improvement is seen when increasing the number of records, allowing the cybersecurity experts to draw a decision in the trade off between the cost of gathering labeled data and the small amelioration noticed in the IDS performance. When we evaluate the IDS in a different environment from the one it was trained, we see that with ten thousands records we have evaluation metrics that range from 72% to 80%. Increasing the number of training records a more noticeable improvement of the IDS performance is noticed, which however remains relatively low.

5.2.2. IDS using the semi-supervised learning paradigm

In the semi-supervised learning paradigm we mixed gradually the labeled CIC-IDS2017 dataset with the unlabeled PS-Azure2023 datasets and we used the label propagation algorithm to assign labels in the PS-Azure2023 records as we discussed in subsection 4.4. By training with this new dataset and evaluating it in the network environment that generated the CIC-IDS2017 we can see that the evaluation metrics in Tables 6 - 9 to be the same or increased. For instance, in binary classification with random forests using only ten thousand records we can see the evaluation metrics to go from 98.08% to 99.96% which is a significant improvement. These results show that the semi-supervised approach can improve the performance of an IDS that is trained and evaluated in the same network environment.

Tables 10 to 13 show that we can have a significant improvement in the evaluation metrics when we train decision trees and random forest with the new dataset and evaluate in the new network environment that generated the CIC-IDS2018. This confirms our hypothesis that the enrichment with new records that have labels with a semi-supervised approach can tackle the limitation of IDS domain generalization. In addition, similar with the supervised approach we can see an improvement of the evaluation metrics when the number of records is increased.

The comparison of machine learning methodologies except from the accuracy should also include the inference time and the training time. Regarding the inference time the semi-supervised approach does not have any overhead compared with the supervised, since the type and the size of the models are the same. For the training time we have two types of delays. The first is the delay of LPA. The time complexity of LPA is nearly linear, each iteration has complexity $O(e)$, with e being

Table 14
Binary-Decision Trees: Evaluating the Generalizability on NF-CSE-CIC-IDS2018.

	Percentage used	Number of Records	Evaluated on CIC-IDS2017				Evaluated on NF-CSE-CIC-IDS2018			
			Acc. %	Prec. %	Rec. %	F1 %	Acc. %	Prec. %	Rec. %	F1 %
Trained: CIC-IDS2017	100%	2.8m	98.31	98.34	98.31	98.32	47.65	71.17	47.65	56.84
	0.0170%	50k	98.11	98.11	98.11	98.11	17.86	86.94	17.86	14.05
	0.0030%	10k	97.59	97.63	97.59	97.59	23.86	74.78	23.86	27.00
Trained: CIC-IDS2017 & PS-Azure2023	100%	2.8m	98.37	98.41	98.37	98.38	48.17	71.32	48.17	57.27
	0.0170%	50k	99.45	99.49	99.45	99.45	87.96	81.52	87.96	82.56
	0.0030%	10k	99.80	99.80	99.80	99.80	82.06	77.07	82.06	79.47

Table 15
Multiclass-Decision Trees: Evaluating the Generalizability on NF-CSE-CIC-IDS2018.

	Percentage used	Number of Records	Evaluated on CIC-IDS2017				Evaluated on NF-CSE-CIC-IDS2018			
			Acc. %	Prec. %	Rec. %	F1 %	Acc. %	Prec. %	Rec. %	F1 %
Trained: CIC-IDS2017	100%	2.8m	98.30	98.38	98.30	98.30	85.33	77.47	85.33	81.21
	0.0170%	50k	98.07	98.06	98.07	98.04	8.65	85.66	8.65	11.53
	0.0030%	10k	97.55	97.57	97.55	97.55	0.57	28.37	0.57	0.76
Trained: CIC-IDS2017 & PS-Azure2023	100%	2.8m	98.37	98.44	98.37	98.37	85.39	77.48	85.33	81.21
	0.0170%	50k	99.45	99.44	99.45	99.44	87.36	77.68	87.36	82.24
	0.0030%	10k	99.82	99.81	99.82	99.81	64.83	74.76	64.83	69.44

the number of edges [49]. The total delay to annotate the PS-Azure2023 dataset with labels in a commodity personal computer was close to 27 secs. The second delay is about the training of the decision tree or random forest. The training with the new dataset that includes the labeled PS-Azure2023 and the CIC-IDS2017 adds an overhead close to 10 secs compared to the training with only CIC-IDS2017. Since the LPA and the training of the machine learning model takes place only one time, these delays are not considered important.

We demonstrate the efficacy of our suggested method and strengthen the scientific validity of our study by utilizing the whole, labeled PS-Azure2023 in the experimental evaluation for the various thresholds. For all peers, but particularly in the cybersecurity industry where data scarcity has been a major issue, using as much available data as possible is the way to go. It is noteworthy to mention that the data from PS-Azure2023 originate from our suggested solution, which added a label annotation to the previously inexpensive data, turning them into valuable ones.

A comparison between Table 1 and Table 2 highlights the issue of different data distributions across two network environments. Subsequently, a comparison between Tables 6 and 10, as well as Tables 7 and 11, demonstrates that the results of a supervised model trained on the CIC-IDS2017 network environment are invalidated when the model runs in a different network environment. However, our proposed methodology, which employs a semi-supervised learning paradigm, significantly mitigates the problem of varying data distributions, as shown in Tables 8 to 13. The new data instances introduced from the PS-Azure2023 dataset, in combination with LPA, provide additional knowledge to the machine learning model, enabling it to generalize efficiently in the new network environment.

5.2.3. Demonstration of generalizability & robustness

To demonstrate the generalization capability of our proposed methodology using a different dataset and showcasing its scientific robustness, we made experiments with the NF-CSE-CIC-IDS2018-v2 dataset. The NF-CSE-CIC-IDS2018-v2 dataset is presented in the papers [50] and [51] and features 43 attributes aligned with the NetFlow network metadata collection protocol. In the experiments we used the tree-based classifiers, Decision Trees and Random Forest, as detailed in the subsection 4.4. In order to evaluate the classifiers using the NF-CSE-CIC-IDS2018-v2 dataset, we first had to retrain our models with CICIDS2017, transforming the latter to align with the former. This involved matching the target labels and ensuring feature alignment between CICIDS2017 and NF-CSE-CIC-IDS2018-v2 as described in the subsection 4.2.

In the new experimental evaluation we used for training the CIC-IDS2017 dataset with 10k, 50k and 2.8m records and for evaluation the NF-CSE-CIC-IDS2018-v2 dataset. We made experiments again with decision trees and random forests for binary and multiclass classification. The experimental outcomes are summarized in the Tables 14, 15, 16, and 17. The evaluation results for both models and classification types further confirm our scientific claim regarding performance degradation when ML-based Network IDS models are tested in different environments. In all cases, the performance evaluation metrics are highly accurate when assessed using the CIC-IDS2017 dataset. However, they show a significant decline when evaluated with the NF-CSE-CIC-IDS2018-v2 dataset. Applying our semi-supervised methodology with the Azure2023 dataset shows significant improvements compared to the supervised machine learning approach that does not use unlabeled records. Furthermore, the results show that having access to the entire dataset provides little improvement, even with the addition of PS-Azure2023, though the impact is more noticeable for smaller data samples. Moreover, we observed significant drops in the performance of the Decision Tree classifier when evaluated with the NF-CSE-CIC-IDS2018-v2 dataset, which is mitigated when we enriched the dataset with the PS-Azure2023. These findings clearly demonstrate how a change in the testbed can cause supervised ML-based IDS models to fail, while our semi-supervised proposed methodology can be an effective way to address this issue.

5.2.4. Comparison with SOTA IDS methods

In order to compare the performance of our proposed methodology with the latest algorithms in the field, we explored state of the art IDS methods capable of addressing modern-day cyberattacks. The state of the art method Multi-Tiered Hybrid IDS (MTH-IDS) [20] is also evaluated with the CICIDS2017 dataset. The MTH-IDS approach involves several stages of data processing, including data pre-processing and feature engineering, to prepare the data for their prediction models. The next key step involves the four tiers of learning models, specifically four tree-based supervised learning algorithms. Finally, the method incorporates a stacking ensemble model and optimizes the supervised learners using Bayesian optimization with a Tree Parzen Estimator (BO-TPE).

For data preprocessing in the MTH-IDS methodology, only a small subset of CICIDS2017 was used, simulating real-world data scarcity in industry applications. In the next phase, feature selection was conducted

Table 16
Binary-Random Forests: Evaluating the Generalizability on NF-CSE-CIC-IDS2018.

	Percentage used	Number of Records	Evaluated on CIC-IDS2017				Evaluated on NF-CSE-CIC-IDS2018			
			Acc. %	Prec. %	Rec. %	F1 %	Acc. %	Prec. %	Rec. %	F1 %
Trained: CIC-IDS2017	100%	2.8m	98.32	98.36	98.32	98.33	37.01	80.34	37.01	44.30
	0.0170%	50k	98.18	98.17	98.18	98.18	37.01	80.34	37.01	44.30
	0.0030%	10k	97.76	97.79	97.76	97.76	87.20	77.53	87.20	82.04
Trained: CIC-IDS2017 & PS-Azure2023	100%	2.8m	98.38	98.42	98.38	98.39	48.92	71.30	48.92	57.87
	0.0170%	50k	99.48	99.48	99.48	99.48	54.61	73.90	54.61	62.41
	0.0030%	10k	99.83	99.83	99.83	99.83	88.04	77.52	88.04	82.45

Table 17
Multiclass-Random Forests: Evaluating the Generalizability on NF-CSE-CIC-IDS2018.

	Percentage used	Number of Records	Evaluated on CIC-IDS2017				Evaluated on NF-CSE-CIC-IDS2018			
			Acc. %	Prec. %	Rec. %	F1 %	Acc. %	Prec. %	Rec. %	F1 %
Trained: CIC-IDS2017	100%	2.8m	98.31	98.39	98.31	98.31	88.04	77.52	88.04	82.45
	0.0170%	50k	98.16	98.15	98.16	98.12	75.95	77.31	75.95	76.60
	0.0030%	10k	97.82	97.80	97.82	97.80	54.37	82.82	54.37	65.40
Trained: CIC-IDS2017 & PS-Azure2023	100%	2.8m	98.38	98.45	98.38	98.38	88.04	77.52	88.04	82.45
	0.0170%	50k	99.49	99.48	99.49	99.48	88.04	77.52	88.04	82.45
	0.0030%	10k	99.84	99.83	99.84	99.84	88.04	77.52	88.04	82.45

Table 18
Evaluation Outcomes with the MTH-IDS method.

	Evaluated on CIC-IDS2017				Evaluated on CIC-IDS2018			
Method	Acc. %	Prec. %	Rec. %	F1 %	Acc. %	Prec. %	Rec. %	F1 %
Extra Tress	99.71	99.71	99.71	99.71	82.59	69.79	82.59	75.06
DecicionTrees	99.20	99.21	99.20	99.21	76.39	69.92	76.39	71.97
RandomForest	99.56	99.56	99.56	99.56	82.91	68.86	82.91	75.24
XGBoost	99.79	99.79	99.79	99.79	78.24	69.75	78.24	73.74
Ensemble	99.56	99.56	99.56	99.56	82.59	68.80	82.59	75.06

using two techniques: Information Gain (IG) and Fast Correlation-Based Filter (FCBF). Additionally, the SMOTE technique was applied to address class imbalance.

The authors' decision to use tree-based classifiers reinforces our own findings that tree-based learners outperform other machine learning methods. The MTH-IDS integrates four supervised classifiers: Decision Trees, Random Forest, Extra Trees, and Extreme Gradient Boosting (XGBoost). For each classifier, hyperparameter optimization (HPO) was performed using Bayesian optimization with the tree-based Parzen Estimator (BO-TPE). In the final stage, these four methods were combined into an ensemble model for training and evaluation. To demonstrate the validity of our research, we replicated the MTH-IDS experiments, training the four tree-based classifiers. In the final step, we evaluated the models in a different environment using the CICIDS2018 dataset as our evaluation testbed.

The results confirm the superiority of our proposed methodology. As shown in Table 18, when evaluated using records from the same network environment as the training dataset (CICIDS2017), the model achieved a very high accuracy of nearly 99.50%. However, when the same model was tested in a different network environment (CICIDS2018), the performance metrics dropped significantly, confirming the degradation in performance across different network environments. Furthermore, when comparing the experimental results of the MTH-IDS method in Table 18 with the results of our proposed method in Table 10, it is clear that our semi-supervised approach outperforms the ensemble machine learning models. Specifically, the four-tier stacking ensemble model, even when optimized using HPO, underperformed compared to our semi-supervised method with the PS-Azure2023 dataset. This underscores the effectiveness of our methodology in the persistent challenges faced by modern IDS models when tested in varying network environments.

5.3. Limitations of our approach

While we proposed an IDS semi-supervised approach based on the label propagation algorithm, there is still room for exploring alternative methodologies that may offer improved efficiency and performance. Although LPA demonstrates commendable performance, its efficacy could potentially be surpassed by other techniques that warrant investigation in future research endeavors.

The inherent characteristic of IDS datasets as significant imbalance datasets, with the malicious class comprising the minority class and the benign class constituting the majority one, poses a notable challenge. Our approach does not directly address the issue of class imbalance, which could result in the IDS missing some malicious attacks and ultimately yielding suboptimal performance. Future research should prioritize the development of strategies to effectively mitigate class imbalance in IDS datasets, thereby enhancing the detection capabilities of IDS.

Furthermore, it is imperative to acknowledge the necessity for an online learning approach within the context of IDS. An online learning approach would enable the IDS to dynamically adapt and update its model parameters based on incoming network traffic in real-time. This capability is crucial for ensuring the responsiveness of IDS to evolving cyber threats and maintaining robust performance over time. Future research directions should focus on the exploration and development of online learning techniques tailored specifically to the requirements of IDS, thereby facilitating continuous learning and adaptation in dynamic network environments.

6. Conclusion and future works

In this paper, we have shown experimentally the performance degradation of machine learning based IDS when they run in a different

network environment from the one they were trained, revealing its generalization limitation. Additionally, a new methodology was proposed that follows the semi-supervised learning paradigm and tackles sufficiently this performance degradation. Our workflow began with the easy and cheap collection of unlabeled data that takes labels using the label propagation algorithm and a public available dataset. Thus, this way, a new labeled and updated dataset was collected. When the machine learning based IDS was trained with this new and enhanced dataset the performance degradation was mitigated in both binary and multiclass classification scenarios.

As future work, and besides addressing the limitations identified in the previous section, we want to examine data augmentation techniques in order to train the IDS with larger synthetic datasets. Given a small number of labeled records we can generate synthetic labeled records. This raises the research question whether the training with this new synthetic dataset can improve the performance of the IDS. In addition, the generation of synthetic data records can also solve the class imbalance problem, which characterizes the cyberattack datasets, by generating records from minority labels and making the IDS capable of detecting rare types of attacks.

CRediT authorship contribution statement

Panagis Sarantos: Writing – original draft, Visualization, Validation, Software, Resources, Investigation, Data curation. **John Violos:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **Aris Leivadeas:** Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

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

Acknowledgment

This work has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), grant No. RGPIN-2019-05250.

References

[1] K. Shaukat, S. Luo, V. Varadharajan, I.A. Hameed, M. Xu, A survey on machine learning techniques for cyber security in the last decade, in: IEEE Access, IEEE Access 8 (2020) 222 310–222 354.
[2] N.H. Chowdhury, M.T.P. Adam, G. Skinner, The impact of time pressure on cybersecurity behaviour: a systematic literature review, Behav. Inf. Technol. 38 (12) (Dec. 2019) 1290–1308, https://doi.org/10.1080/0144929X.2019.1583769, [Online].
[3] D.M. Cappelli, A.P. Moore, R.F. Trzeciak, The CERT Guide to Insider Threats: How to Prevent, Detect, and Respond to Information Technology Crimes (Theft, Sabotage, Fraud), Addison-Wesley, Jan. 2012, google-Books-ID: VvBLK91q4LEC.
[4] NIST special publication on intrusion detection systems, Tech. Rep., section: Technical Reports. [Online]. Available: https://apps.dtic.mil/sti/citations/ADA393326.
[5] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, Survey of intrusion detection systems: techniques, datasets and challenges, Cybersecurity 2 (1) (Jul. 2019) 20, https://doi.org/10.1186/s42400-019-0038-7, [Online].
[6] H. Alqahtani, I.H. Sarker, A. Kalim, S.M. Minhaz Hossain, S. Ikhlaq, S. Hossain, Cyber intrusion detection using machine learning classification techniques, in: N. Chaubey, S. Parikh, K. Amin (Eds.), Computing Science, Communication and Security, in: Communications in Computer and Information Science, Springer, Singapore, 2020, pp. 121–131.
[7] J. Wang, C. Lan, C. Liu, Y. Ouyang, T. Qin, W. Lu, Y. Chen, W. Zeng, P.S. Yu, Generalizing to unseen domains: a survey on domain generalization, in: IEEE Transactions on Knowledge and Data Engineering, IEEE Trans. Knowl. Data Eng. 35 (8) (Aug. 2023) 8052–8072, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9782500.
[8] A. Leivadeas, M. Falkner, A survey on intent-based networking, IEEE Commun. Surv. Tutor. 25 (1) (2023) 625–655.
[9] J.E. van Engelen, H.H. Hoos, A survey on semi-supervised learning, Mach. Learn. 109 (2) (Feb. 2020) 373–440, https://doi.org/10.1007/s10994-019-05855-6, [Online].
[10] I. Redko, E. Morvant, A. Habrard, M. Sebban, Y. Bennani, A survey on domain adaptation theory: learning bounds and theoretical guarantees, arXiv:2004.11829 [cs, stat], Jul. 2022, [Online]. Available: http://arxiv.org/abs/2004.11829.
[11] I. Sharafaldin, A. Gharib, A. Habibi Lashkari, A. Ghorbani, Towards a reliable intrusion detection benchmark dataset, Softw. Netw. 2017 (Jan. 2017) 177–200.
[12] I. Sharafaldin, A. Habibi Lashkari, A.A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization, in: Proceedings of the 4th International Conference on Information Systems Security and Privacy, SCITEPRESS - Science and Technology Publications, Funchal, Madeira, Portugal, 2018, pp. 108–116.
[13] H.-J. Liao, C.-H. Richard Lin, Y.-C. Lin, K.-Y. Tung, Intrusion detection system: a comprehensive review, J. Netw. Comput. Appl. 36 (1) (Jan. 2013) 16–24, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804512001944.
[14] A. Leivadeas, M. Falkner, I. Lambadaris, G. Kesidis, Dynamic traffic steering of multitenant virtualized network functions in sdn enabled data centers, in: 2016 IEEE 21st International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD), 2016, pp. 65–70.
[15] A. Khraisat, I. Gondal, P. Vamplew, An anomaly intrusion detection system using C5 decision tree classifier, in: M. Ganji, L. Rashidi, B.C.M. Fung, C. Wang (Eds.), Trends and Applications in Knowledge Discovery and Data Mining, in: Lecture Notes in Computer Science, Springer International Publishing, Cham, 2018, pp. 149–155.
[16] T. Saba, A. Rehman, T. Sadad, H. Kolivand, S.A. Bahaj, Anomaly-based intrusion detection system for IoT networks through deep learning model, Comput. Electr. Eng. 99 (Apr. 2022) 107810, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790622001100.
[17] W.-C. Lin, S.-W. Ke, C.-F. Tsai, CANN: an intrusion detection system based on combining cluster centers and nearest neighbors, Knowl.-Based Syst. 78 (Apr. 2015) 13–21, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705115000167.
[18] T. Saranya, S. Sridevi, C. Deisy, T.D. Chung, M.K.A.A. Khan, Performance analysis of machine learning algorithms in intrusion detection system: a review, Proc. Comput. Sci. 171 (Jan. 2020) 1251–1260, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050920311121.
[19] A. Thakkar, R. Lohiya, A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions, Artif. Intell. Rev. 55 (1) (Jan. 2022) 453–563, https://doi.org/10.1007/s10462-021-10037-9, [Online].
[20] L. Yang, A. Moubayed, A. Shami, MTH-IDS: a multitiered hybrid intrusion detection system for Internet of vehicles, in: IEEE Internet of Things Journal, IEEE Int. Things J. 9 (1) (Jan. 2022) 616–632, [Online]. Available: https://ieeexplore.ieee.org/document/9443234.
[21] M.K. Ngueajio, G. Washington, D.B. Rawat, Y. Ngueabou, Intrusion detection systems using support vector machines on the KDDCUP'99 and NSL-KDD datasets: a comprehensive survey, in: K. Arai (Ed.), Intelligent Systems and Applications, Springer International Publishing, Cham, 2023, pp. 609–629.
[22] A. Abbas, M.A. Khan, S. Latif, M. Ajaz, A.A. Shah, J. Ahmad, A new ensemble-based intrusion detection system for Internet of things, Arab. J. Sci. Eng. 47 (2) (Feb. 2022) 1805–1819, https://doi.org/10.1007/s13369-021-06086-5, [Online].
[23] J. Lansky, S. Ali, M. Mohammadi, M.K. Majeed, S.H.T. Karim, S. Rashidi, M. Hosseinzadeh, A.M. Rahmani, Deep learning-based intrusion detection systems: a systematic review, in: IEEE Access, IEEE Access 9 (2021) 101 574–101 599, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9483916.
[24] S.M. Kasongo, A deep learning technique for intrusion detection system using a recurrent neural networks based framework, Comput. Commun. 199 (Feb. 2023) 113–125, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366422004601.
[25] V. Hnamte, J. Hussain, DCNNBiLSTM: an efficient hybrid deep learning-based intrusion detection system, Telemat. Inform. Rep. 10 (Jun. 2023) 100053, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2772503023000130.
[26] Z. Wu, H. Zhang, P. Wang, Z. Sun, RTIDS: a robust transformer-based approach for intrusion detection system, in: IEEE Access, IEEE Access 10 (2022) 64 375–64 387, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9794665.
[27] E.E. Abdallah, W. Eleisah, A.F. Otoom, Intrusion detection systems using supervised machine learning techniques: a survey, Proc. Comput. Sci. 201 (Jan. 2022) 205–212, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050922004422.
[28] N. Alkhatib, M. Mushtaq, H. Ghauch, J.-L. Danger, Unsupervised network intrusion detection system for AVTP in automotive ethernet networks, in: 2022 IEEE Intelligent Vehicles Symposium (IV), Jun. 2022, pp. 1731–1738, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9827285.
[29] I. Amit, J. Matherly, W. Hewlett, Z. Xu, Y. Meshi, Y. Weinberger, Machine learning in cyber-security - problems, challenges and data sets, arXiv:1812.07858 [cs, stat], Apr. 2019, [Online]. Available: http://arxiv.org/abs/1812.07858.
[30] M.F. Umer, M. Sher, Y. Bi, Flow-based intrusion detection: techniques and challenges, Comput. Secur. 70 (Sep. 2017) 238–254, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404817301165.
[31] C. Göpfert, S. Ben-David, O. Bousquet, S. Gelly, I. Tolstikhin, R. Urner, When can unlabeled data improve the learning rate?, in: Proceedings of the Thirty-Second Conference on Learning Theory, PMLR, Jun. 2019, pp. 1500–1518, ISSN: 2640-3498, [Online]. Available: https://proceedings.mlr.press/v99/gopfert19a.html.

[32] Z.-W. Zhang, X.-Y. Jing, T.-J. Wang, Label propagation based semi-supervised learning for software defect prediction, Autom. Softw. Eng. 24 (1) (Mar. 2017) 47–69, https://doi.org/10.1007/s10515-016-0194-x, [Online].
[33] Y. Guo, A review of machine learning-based zero-day attack detection: challenges and future directions, Comput. Commun. 198 (Jan. 2023) 175–185, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366422004248.
[34] J. Lyngdoh, M.I. Hussain, S. Majaw, H.K. Kalita, An intrusion detection method using artificial immune system approach, in: A.K. Luhach, D. Singh, P.-A. Hsiung, K.B.G. Hawari, P. Lingras, P.K. Singh (Eds.), Advanced Informatics for Computing Research, in: Communications in Computer and Information Science, Springer, Singapore, 2019, pp. 379–387.
[35] O. Chapelle, B. Scholkopf, A. Zien, Semi-supervised learning (Chapelle, O. et al., Eds.; 2006) [Book reviews], in: IEEE Transactions on Neural Networks, IEEE Trans. Neural Netw. 20 (3) (Mar. 2009) 542, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/4787647.
[36] L. Rokach, O. Maimon, Classification trees, in: O. Maimon, L. Rokach (Eds.), Data Mining and Knowledge Discovery Handbook, Springer US, Boston, MA, 2010, pp. 149–174, [Online].
[37] O. Maimon, L. Rokach (Eds.), Data Mining and Knowledge Discovery Handbook, Springer US, Boston, MA, 2010, [Online]. Available: http://link.springer.com/10.1007/978-0-387-09823-4.
[38] A. Parmar, R. Katariya, V. Patel, A review on random forest: an ensemble classifier, in: J. Hemanth, X. Fernando, P. Lafata, Z. Baig (Eds.), International Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI) 2018, in: Lecture Notes on Data Engineering and Communications Technologies, Springer International Publishing, Cham, 2019, pp. 758–763.
[39] About the CIC | Canadian Institute for Cybersecurity | UNB, [Online]. Available: https://www.unb.ca/cic/about/index.html.
[40] S. Liao, C. Zhou, Y. Zhao, Z. Zhang, C. Zhang, Y. Gao, G. Zhong, A comprehensive detection approach of nmap: principles, rules and experiments, in: 2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Oct. 2020, pp. 64–71, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9329410.
[41] C.I.f. Cybersecurity, CanadianInstituteForCybersecurity/CICFlowMeter, [Online]. Available: https://github.com/CanadianInstituteForCybersecurity/CICFlowMeter, Aug. 2022, original-date: 2020-02-03T17:34:31Z.
[42] M. Verleysen, D. François, The curse of dimensionality in data mining and time series prediction, in: J. Cabestany, A. Prieto, F. Sandoval (Eds.), Computational Intelligence and Bioinspired Systems, in: Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2005, pp. 758–770.
[43] CICFlowMeter/ReadMe.txt at master · ahlashkari/CICFlowMeter, [Online]. Available: https://github.com/ahlashkari/CICFlowMeter/blob/master/ReadMe.txt.
[44] G. Wu, J. Xu, Optimized approach of feature selection based on information gain, in: 2015 International Conference on Computer Science and Mechanical Automation (CSMA), 2015, pp. 157–161.
[45] Kurniabudi, D. Stiawan, Darmawijoyo, M.Y. Bin Idris, A.M. Bamhdi, R. Budiarto, CICIDS-2017 dataset feature analysis with information gain for anomaly detection, in: IEEE Access, IEEE Access 8 (2020) 132 911–132 921.
[46] D.M.W. Powers, What the F-measure doesn't measure: features, flaws, fallacies and fixes, arXiv:1503.06410 [cs, stat], Sep. 2019, [Online]. Available: http://arxiv.org/abs/1503.06410.
[47] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update, ACM SIGKDD Explor. Newsl. 11 (1) (Nov. 2009) 10–18, https://doi.org/10.1145/1656274.1656278, [Online].
[48] E. Bisong, Introduction to Scikit-learn, in: E. Bisong (Ed.), Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners, Apress, Berkeley, CA, 2019, pp. 215–229, [Online].
[49] X. Hu, W. He, H. Li, J. Pan, Role-based label propagation algorithm for community detection, arXiv:1601.06307 [physics], Jan. 2016, [Online]. Available: http://arxiv.org/abs/1601.06307.
[50] M. Sarhan, S. Layeghy, M. Portmann, Towards a standard feature set for network intrusion detection system datasets, Mob. Netw. Appl. 27 (1) (Feb. 2022) 357–370, https://doi.org/10.1007/s11036-021-01843-0, [Online].
[51] M. Sarhan, S. Layeghy, N. Moustafa, M. Portmann, NetFlow datasets for machine learning-based network intrusion detection systems, in: Z. Deze, H. Huang, R. Hou, S. Rho, N. Chilamkurti (Eds.), Big Data Technologies and Applications, Springer International Publishing, Cham, 2021, pp. 117–135.