A Novel Multi-Stage Approach for Hierarchical Intrusion Detection

Miel Verkerken, Laurens D'hooge, Didik Sudyana, Ying-Dar Lin, *Fellow, IEEE*, Tim Wauters, *Member, IEEE*, Bruno Volckaert, *Senior Member, IEEE*, and Filip De Turck, *Fellow, IEEE*

***Abstract*—An intrusion detection system (IDS), traditionally an example of an effective security monitoring system, is facing significant challenges due to the ongoing digitization of our modern society. The growing number and variety of connected devices are not only causing a continuous emergence of new threats that are not recognized by existing systems, but the amount of data to be monitored is also exceeding the capabilities of a single system. This raises the need for a scalable IDS capable of detecting unknown, zero-day, attacks. In this paper, a novel multi-stage approach for hierarchical intrusion detection is proposed. The proposed approach is validated on the public benchmark datasets, CIC-IDS-2017 and CSE-CIC-IDS-2018. Results demonstrate that our proposed approach besides effective and robust zero-day detection, outperforms both the baseline and existing approaches, achieving high classification performance, up to 96% balanced accuracy. Additionally, the proposed approach is easily adaptable without any retraining and takes advantage of n-tier deployments to reduce bandwidth and computational requirements while preserving privacy constraints. The best-performing models with a balanced set of thresholds correctly classified 87% or 41 out of 47 zero-day attacks, while reducing the bandwidth requirements up to 69%.**

***Index Terms*—Intrusion detection, binary classification, multi-class classification, multi-stage detection, hierarchical architecture.**

I. Introduction

OUR SOCIETY is continuously exposed to an increased risk of cybersecurity threats due to the ongoing digitization in the modern world [1]. The never-ending growing number and variety of interconnected devices, including critical systems such as power grids, does not only expand the attack surface for a malicious actor but is also negatively affecting the possible consequences in case of a successful attack [2]. Furthermore, the increasing generated load on existing security monitoring systems is exceeding single system capabilities and challenging their scalability to detect threats in near real-time [3]. As a result, the historical arms race between attackers and defenders is shifting advantageously towards the attackers. This demands consistent efforts from the research community to further improve the existing defenses in place as well as develop novel approaches that contribute to the mitigation of the cybersecurity risk.

Traditional security mechanisms such as a firewall are effective at detecting specific types of attacks but are unable to detect unknown or more advanced attacks. Intrusion detection systems (IDS) are often deployed as a second line of defense and are an example of a security monitoring system capable of detecting known as well as unknown and more sophisticated attacks. The detection can happen before, during, or after an attack is executed. In case the system is not only capable of detecting such an intrusion but also actively prevents the attack from succeeding, it is called an intrusion prevention system (IPS). An IDS and IPS can be categorized by the source of the input used for detection. Host-based intrusion detection systems (HIDS) use features gathered from the host machine such as resource usage and system calls, therefore a monitoring module needs to be deployed on every single device that needs to be secured. On the other hand, a network intrusion detection system (NIDS) is deployed on a particular node in the network and monitors all traffic passing through it. When a combination of both sources is used as input for the detection, it is called a hybrid intrusion detection system. These systems can also be differentiated by the applied method for detection. Signature or misuse-based systems rely on a database consisting of patterns or signatures of known attacks. This technique is an effective tool for detecting known attacks with few false alerts but fails to detect any attack not present in the signature database and therefore is unable to detect zero-day attacks. On the contrary, anomaly-based detection models normal behavior instead of the attack itself, and everything deviating too much from normal is flagged as malicious. This approach allows the detection of both known and unknown attacks as long as they diverge sufficiently from the learned baseline. Anomaly-based detection often relies on machine-learning techniques to learn the normal baseline from data.

Currently proposed IDS solutions often rely on a single machine learning model for either attack detection or classification. This poses multiple challenges. First, a single model generally excels either in attack detection or classification. Using a combination of multiple models particularly trained for a specific task could potentially improve the classification

Manuscript received 15 July 2022; revised 13 December 2022; accepted 27 February 2023. Date of publication 21 March 2023; date of current version 9 October 2023. This work is supported by the Belgian Chancellery of the Prime Minister (Grant: AIDE-BOSA). The associate editor coordinating the review of this article and approving it for publication was S. Scott-Hayward. *(Corresponding author: Miel Verkerken.)*

Miel Verkerken, Laurens D'hooge, Tim Wauters, Bruno Volckaert, and Filip De Turck are with the Department of Information Technology, IDLab, Ghent University–imec, 9052 Ghent, Belgium (e-mail: Miel.Verkerken@UGent.be).

Didik Sudyana and Ying-Dar Lin are with the Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu 300, Taiwan.

Digital Object Identifier 10.1109/TNSM.2023.3259474

performance. Secondly, all samples need to transmitted to and processed by this single model. In case the monitored network is distributed, this will not only lead to high computational costs but also have high bandwidth requirements, leading to increased latency. Last, the proposed models often lack the ability to detect unknown or zero-day attacks.

In this paper, a novel multi-stage approach for hierarchical intrusion detection is proposed. Fundamentally, this approach is applicable for each type of IDS, independent of the source of the input or used detection method, considering the characteristics of each layer are met. Our novel approach features improved classification performance over both a single model baseline and existing multi-stage approaches, in addition to its ability to effectively detect zero-day attacks. The classification is performed by a system consisting of three stages. The first stage performs a lightweight outlier detection, associating an anomaly score to each event. By only forwarding events to the second stage with an anomaly score above a certain threshold, this layer acts as a filter. This allows the use of more computationally expensive methods in the next layer as it will only be applied to a small share of the total number of events. The second stage classifies each of the suspicious events from the first stage to a known attack type with a certain confidence. Events with low prediction confidence, most likely do not belong to any of the known attack classes and are forwarded to the third stage. This last stage reuses the anomaly score of the first stage to separate miss-classified benign events from the first stage and zero-day attacks. Events with an anomaly score higher than another threshold which is higher than the previous threshold in the first stage, are flagged as an unknown attack while events with a lower score are corrected as benign. Since each layer employs a threshold to define its prediction, the final performance can easily be tuned by adjusting these values without retraining any of the used models. This flexibility also allows adapting the models' specific trade-offs to changing requirements over time. Finally, the novel approach is designed to take advantage of a hierarchical deployment. Each of the stages can either be deployed separately or combined. This empowers an n-tier deployment of our novel proposed approach that minimizes bandwidth requirements and latency associated with the predictions. In case the first stage is deployed close to the network being monitored, a privacy-aware operation is ensured as only suspicious events are forwarded, retaining most of the benign traffic locally.

The main contributions of this paper are three-fold.

- We propose a novel multi-stage approach for hierarchical intrusion detection performing both binary and multi-class detection. Our highly adaptable novel approach is specifically designed to empower a multi-tier deployment to minimize latency and bandwidth requirements while preserving privacy constraints. Furthermore, both known and zero-day attacks can be detected.
- A hyperparameter optimization using machine learning best practices is performed for two unsupervised, autoencoder and one-class support vector machine, and two supervised, random forest and neural network, algorithms for the first and second stage, respectively.
- An extensive validation and analysis of our novel proposed approach is performed on modern flow-based network intrusion datasets, CIC-IDS-2017 and CSE-CIC-IDS-2018 [4], against both a single baseline model and the existing state-of-the-art multi-stage approach.

The remainder of this paper is structured as follows. First, the related work regarding IDS and more specifically multi-stage IDS is discussed in Section II. Afterward, the novel multi-stage approach for hierarchical intrusion detection is presented in Section III, followed by Section IV describing the methodology used to validate the newly proposed approach, ensuring sound and reproducible results. In Section V the results of both the baseline, all intermediate stages, and the final novel approach on a modern flow-based network dataset are presented. An extensive analysis and discussion of these results are laid out in Section VI. Future work is listed in Section VII before stating a final conclusion in Section VIII.

II. Related Work

In the literature several studies regarding multi-layer or multi-stage IDS exist, starting from a different definition. A first definition refers to a hierarchical context where each layer has different input data available to execute the detection or classification, the final prediction is then a combination of each tier's prediction. Zhang et al. [5] propose an IDS to detect cyberattacks in smart grids. This smart grid exists of three layers where each layer has the ability to monitor a unique set of features used for attack detection. Through internal communication between the different layers, the smart grid is able to identify malicious traffic. Similarly, Ali and Yousaf [6] proposed an approach composed of three tiers to detect intrusions in software-defined networks (SDN). The first tier validates user authentication through an RFID tag and encrypted signatures using routers as edge devices. Next, the second tier located on switches validates the raw network packets using fuzzy filtering. In the third tier, located in network controllers, the reconstructed flows from the raw packets are used as input in a convolutional neural network (CNN) for the detection of malicious traffic.

A second definition found in the literature describes multi-stage approaches as a cascade of detection and classification methods on the same input data. The goal is often to achieve a higher classification performance. Li et al. [7] proposed such a cascade to classify network traffic to the correct attack type. The first stage consists of a collection of binary classifiers whose output is aggregated to form a prediction. Traffic for which the aggregated prediction is uncertain is sent to the second stage to determine the correct class using the k-nearest neighbors model (KNN). Likewise, Pajouh et al. [8], [9] used a two-stage feature reduction method followed by a two-stage classification approach chaining the naive Bayes algorithm and certainty factor KNN (CF-KNN) to classify network traffic. All traffic classified as benign by the naive Bayes algorithm is forwarded to the CF-KNN for further investigation. The final prediction is formed by combining both stages. On the contrary, Al-Yaseen et al. [10] chained multiple stages with each stage consisting of a classifier able to detect a single attack

TABLE I
TAXONOMY OF THE RELATED WORK

Study	Detection	Classification	Stages	Zero-Day	Hierarchical	Computational Reduction
Zhang et al. [5]	±	✓	3	✗	✓	✗
Ali and Yousaf [6]	±	✓	3	✗	✓	✗
Li et al. [7]	✗	✓	2	✗	✗	✗
Pajouh et al. [8] [9]	✗	✓	2	✗	✗	✗
Al-Yaseen et al. [10]	✗	✓	5	✓	✗	✗
Ji et al. [11]	✓	✓	2	✗	✗	✗
Khan et al. [12]	✓	✓	2	✗	✗	✗
Divyatmika and Sreekesh [13]	✓	✓	3	✓	✗	±
Umer et al. [14]	✓	✓	2	✗	✗	✗
Abuadlla et al. [15]	✓	✓	2	±	✗	✗
Bovenzi et al. [16]	✓	✓	2	✓	✓	±
Verkerken et al.	✓	✓	3	✓	✓	✓

type instead of using a second layer to reclassify predictions with low confidence. This chain forms a waterfall pattern where each sample propagates to the next classifier until successful detection of an attack type. In case the last classifier also fails to classify the sample as a known attack, the sample is classified as an unknown attack. The study conducted by Ji et al. [11] proposed a two-stage model with the same goal of improving the classification performance but uses rule-based detection followed by an anomaly-based model for the first and second stage, respectively. The approach proposed by Khan et al. [12] consists of two layers with both layers containing a deep autoencoder (DAE) and a soft-max classifier. The DAE is used to construct the latent space of the input data in an unsupervised manner. On top of this latent space, a soft-max classifier is placed which is fine-tuned using a limited set of labeled data. The first layer will output an anomaly probability which is then used as an extra feature in the second layer. The second layer performs the final classification based on both the input data and anomaly probability outputted by the first layer.

Next to achieving a higher classification performance by employing multiple detection methods, the third cluster of studies exists that relies on a combination of anomaly detection, often based on unsupervised machine learning techniques, and a multi-class classifier. Here the goal of the first stage is to filter out suspicious samples in a lightweight manner, which are then sent to the next, more computational complex stage for attack type classification. The goal of these studies is to achieve a high classification performance while reducing latency, bandwidth and computational requirements, often combined with unknown attack detection capabilities [17]. Divyatmika and Sreekesh [13] use a three-stage approach to classify network traffic, where the first stage relies on KNN model to compare incoming traffic against the baseline traffic of the network. New behavior and attacks are not matched and are sent to the following stages where the multi-layer perceptron (MLP) algorithm and reinforcement learning is used for misuse and anomaly detection, respectively. The first stage is acting as a filter to reduce the load by the computational more expensive algorithms in the succeeding stages. The study by Umer et al. [14] presents a multi-stage model for next-generation networks consisting of an anomaly detection and a multi-class classifier relying on solely unsupervised machine-learning techniques. The first layer uses a one-class support vector machine (OC-SVM) for binary detection. Only traffic predicted as malicious by the first layer is classified by the second layer using a self-organizing map. Abuadlla et al. [15] present an easy expandable two-stage model. Both stages rely on a neural network (NN) for respectively, anomaly detection and multi-class classification in the first and second stages. The study mentions the ability to detect unknown attacks but does not further specify in detail how the supervised NN achieves this in the absence of labeled samples of unknown attacks. The two-stage hierarchical approach proposed by Bovenzi et al. [16] consists of a multi-modal DAE and soft output classifiers in the first and second stage, respectively. The soft output classifier in the second stage is able to detect unknown attacks using a threshold on the confidence of the prediction. In case the confidence of a sample belonging to a known attack class or benign traffic is lower than the threshold, the sample is predicted as an unknown attack. The multi-class classifier is trained using the open-set approach, removing one of the known attack types from the training data and considering it as an unknown attack. By repeating this approach for each of the known attack types, the threshold on the confidence can be optimized. Further, this model is optimized for a distributed and privacy-preserving deployment with the need for limited retraining to adjust performance trade-offs. This study is the closest to our novel approach and is used as the current state-of-the-art multi-stage approach for comparative purposes.

Other studies in the literature also describe their work as multi-layer or multi-stage but do not refer to classification in multiple steps or a chain of classifiers. For example, Injadat et al. [18] simply refers to the multiple steps in a data processing pipeline, such as preprocessing, feature selection, hyperparameter optimization, and classification, to claim her proposed model as multi-stage.

While few of the previously discussed related works mentioned the reduction of the required computational capacity of the proposed architecture, none of them provided experimental results or analyzed this statement more thoroughly. This study discusses the reduction on basis of the share of samples propagating to the second layer, thus requiring extra computational resources and bandwidth in a hierarchical deployment. A taxonomy of the related work can be found in Table I.

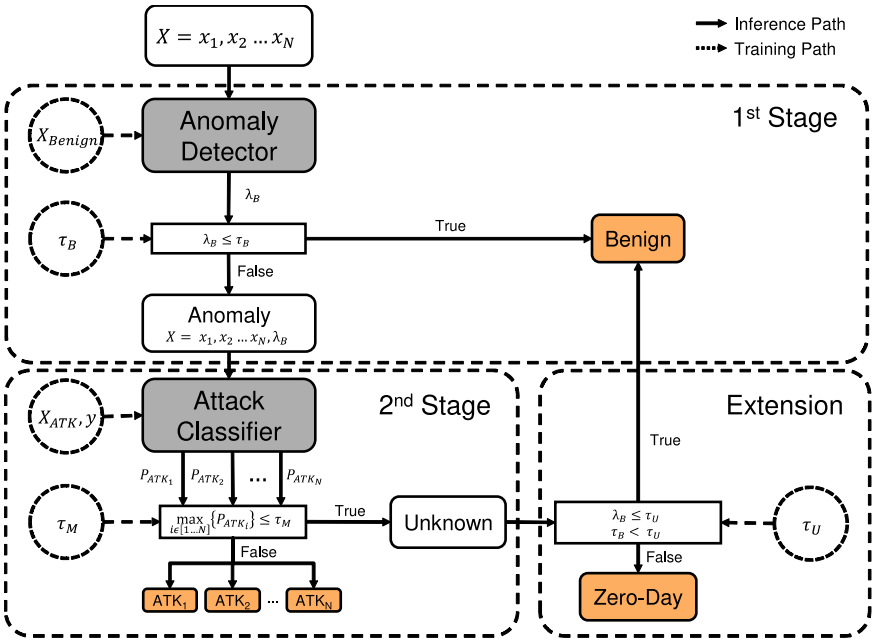


Fig. 1. The proposed architecture of the multi-stage hierarchical intrusion detection system.

For each study it is examined if separate anomaly detection and classification are obtainable, the number of used stages, the ability to detect zero-day attacks, the possibility for a hierarchical deployment, and achieving reduction of required computational capacity. The taxonomy clearly shows the unique characteristics of this study.

The use of machine learning techniques for intrusion detection is a well-studied topic by the research community. Several surveys [19], [20], [21] advocate the usage of both supervised and unsupervised methods to assist existing IDS or even replace them completely. More specifically, promising results have previously been achieved by employing AE [22], [23] and OC-SVM [24] to detect anomalies in streams of network traffic. Similarly, tree-based algorithms, such as an RF [25], [26], and deep learning algorithms, in particular convolutional neural networks [27], [28] and artificial neural networks [29] have been confirmed to achieve promising results on academic benchmark datasets. This study evaluates the use of an AE and OC-SVM for anomaly detection and RF and NN for multi-class classification, respectively, in the first and second stage of the proposed novel multi-stage approach. Furthermore, a RF trained using an open-set option for zero-day detection serves as a single model baseline.

III. Novel Approach

In this section, the novel multi-stage approach for hierarchical intrusion detection is described. First, the overall architecture is introduced and the design choices made throughout the development are explained in depth. Afterward, the stages composing the overall architecture are individually discussed. Finally, the benefits of a hierarchical deployment are highlighted as well as the implementation choices to be made for an operational system.

A. General Architecture

The multi-stage hierarchical architecture proposed in this paper consists of three stages with each a distinct characteristic and objective. Figure 1 presents the overall architecture. The feature vector denoted as X serves as an input to the system and is sent to the first stage. The first stage consists of an anomaly detector that outputs an anomaly score, λ_B, specifically a high value indicates a high probability that X is not benign and thus an intrusion. If this anomaly score is lower than a threshold τ_B, the model is confident enough to predict the sample as benign and no further processing is necessary. However, if the anomaly score is higher than τ_B, the sample is forwarded to the second stage where an attack classifier predicts if the sample belongs to one of the known attack classes (ATK_i). If the attack classifier fails to match the sample to a known attack class with a certainty higher than a threshold τ_M, the sample is sent to the third and last stage. The last stage or extension stage does not introduce another classifier but rather reuses the anomaly score λ_B outputted by the first layer. In case the anomaly score is lower than the threshold τ_U, the output of the first layer is corrected by eventually classifying the sample as benign, on the contrary, the sample is predicted as an unknown or zero-day attack if the anomaly score is higher than τ_U.

Additionally, live adjustment of the thresholds τ_B, τ_M, and τ_U enables our novel approach to adjust its classification performance in real-time without the need for any retraining. This is done by balancing the trade-off between the number of false positives and false negatives in each stage.

The proposed architecture is developed with a scalable hierarchical deployment in mind. The first stage acts as a lightweight filter with minimal hardware requirements, forwarding only suspicious samples. This results in minimal computational cost for most of the benign traffic as these are

not subjected to further analysis by the subsequent stages. Ideally, the first stage is located close to the network being monitored, for example in a fog or edge device, while the other stages can be deployed further away, for instance in a centralized cloud. Section III-E describes in more detail the configuration and benefits of a hierarchical deployment.

$$\begin{aligned} T_{stage1} &= \mathcal{O}(T_B) \\ T_{stage2} &= \mathcal{O}(T_B + T_M) \\ T_{stage3} &= \mathcal{O}(T_B + T_M + T_U) \\ T_{total} &= \mathcal{O}(T_B + \alpha T_M + \alpha\beta T_U) \\ &\Rightarrow \mathcal{O}(T_B + T_M) \end{aligned} \tag{1}$$

where: T_B = Time Complexity Anomaly Detector
T_M = Time Complexity Attack Classifier
T_U = Time Complexity Extension Stage
α = fraction of flows forwarded by 1st stage
β = fraction of flows forwarded by 2nd stage

Equation (1) uses the big O notation to describe the computational complexity of our novel hierarchical approach, regardless of the algorithms used in each stage. The sum of all the stages creates the final performance, which will be impacted accordingly by the algorithms that are ultimately selected at each stage. Note that the computational complexity of the extension stage is equal to $\mathcal{O}(1)$ and thus has no impact on the final complexity since it reuses the anomaly score of the first stage with a new threshold.

While our proposed architecture is independent of both the used input source and employed classification methods, this paper evaluates the novel approach with machine-learning models for detection and classification on a modern flow-based network intrusion detection dataset.

B. Stage 1: Anomaly Detection

The first stage has the goal to filter out malicious samples in a computationally efficient manner. This way the number of samples that need to be analyzed in the subsequent stages is greatly reduced. Because the first stage is applied to each sample and ideally located close to the monitored network generating the input, the binary classifier is required to perform the anomaly detection in a lightweight manner on limited hardware.

The binary model in the first stage is exclusively trained on benign data. This allows the anomaly detector to learn a representation of the normal behavior of the monitored network. During training, the optimal hyper-parameters are selected using the area under the receiver operating characteristic (AUROC) as a validation metric with a validation set consisting of both malicious and benign data. The AUROC is selected because it is independent of the eventually employed threshold τ_B on the anomaly score, which determines the final prediction. After the hyper-parameter optimization of a model, multiple candidate thresholds τ_B are selected by computing the F1 till F9 metric for every unique value encountered in the anomaly scores of the validation set. The value corresponding to a maximum f-score is added to the list of candidate thresholds for this particular model. The eventually optimal threshold depends on the intended result. Since τ_B balances both a performance and computational tradeoff, it is crucial to be carefully set. The impact of the candidate thresholds is analyzed more in detail when the overall performance of the multi-stage approach is assessed.

This study evaluates unsupervised machine-learning techniques to execute the anomaly detection because of their ability to detect both known and unknown intrusions, even when only trained on benign data. But in practice, any technique that outputs an anomaly score is suited.

C. Stage 2: Multi-Class Classification

The second stage attempts to classify the samples predicted as malicious by the first stage to a known attack class using a multi-class classifier. This classifier is trained on exclusively malicious data and is accordingly only able to classify samples to the attack classes present in the training data. A validation set consisting of both malicious and benign data is used to select the optimal hyper-parameters for the model using the F1 score as the validation metric. The classifier outputs a vector, $[P_{ATK_1}, P_{ATK_2}, \ldots, P_{ATK_N}]$, with the probabilities for each of the known attack classes that the input vector X belongs to that attack. The attack with the highest probability is then the predicted class, except if this probability is lower than the threshold τ_M, the sample is predicted as unknown and forwarded to the last stage. The threshold is set to the value that yields the highest weighted f1-score on the validation set. This threshold is responsible for defining how confident the model needs to be before assigning a sample to a known attack class, as a result, it balances a false positive, false negative tradeoff. Since the validation set is composed of both benign and malicious data, the model is not only forced to correctly classify the malicious samples but also to output a low probability for each of the known attack classes for the benign samples, to achieve a high validation score. Similarly, an unknown attack will be associated with a low probability for each of the known attack classes, which will successfully send the sample to the next stage. This resolves the challenge to detect unknown attacks without corresponding labeled data.

This study evaluates supervised machine-learning techniques for the attack classification of known intrusions but could be replaced by any technique able to produce prediction probabilities. However, the model should be able to output if a sample does not correspond to any of the learned attacks by explicitly outputting an extra probability estimation for an unknown class or implicitly by associating low probabilities to each of the known classes.

D. Stage 3: Extension

The third stage or also extension stage does not implement another classifier but rather reuses the anomaly score outputted by the first stage, to produce its prediction. The main goal of this stage is to reduce the number of false positives, namely benign traffic being classified as an attack while providing the ability to effectively detect zero-day attacks. Currently, one of the main challenges anomaly-based IDS are facing is the high number of false positives.

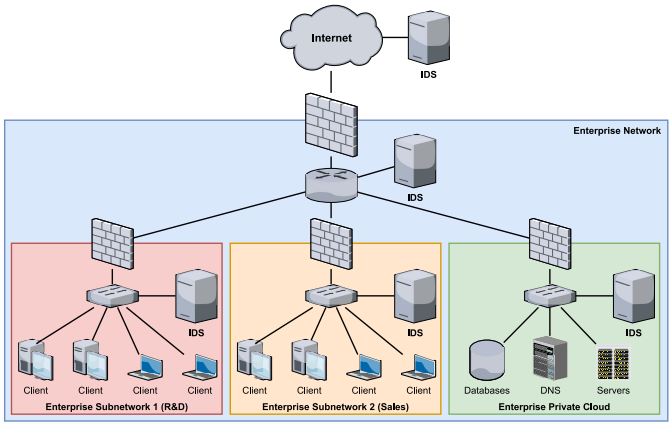


Fig. 2. Representation of an enterprise network consisting of a three-tier IDS and three local subnetworks: R&D, sales, and private cloud.

Since this stage reuses the model of the first stage, no new training or validation sets are required to train the model. Only the threshold τ_U needs to be defined using the same validation set of the first stage. Because one of the main purposes of this stage is to reduce the number of false positives, the threshold is set using the quantile of the anomaly score of the benign samples. As a result, the maximum false positive rate can approximately be set using this threshold. The third stage is only effective if the threshold in this layer (τ_U) is greater than the threshold in the first stage (τ_B). This way the extension layer can achieve a higher precision because a sample needs a higher anomaly score before being marked as an attack. The resulting samples that are eventually detected as malicious by the extension layer have a high probability to be a zero-day attack or known attack but not trained on in the second stage.

E. Hierarchical Deployment

Our novel multi-stage approach for intrusion detection is specifically designed to empower an n-tier hierarchical deployment. To illustrate the benefits of a hierarchical deployment we will introduce a simple enterprise network, see Figure 2, with three local subnetworks secured by a three-tier IDS. To illustrate the advantages of our proposed approach we will use a simple scenario where each stage is placed bottom-up in the three-tier IDS, respectively, the first, second and extension stage are placed in the bottom, middle and upper tier.

The local deployment of the first stage has multiple advantages. On one hand, locating the first stage in the same (sub)network, close to the systems being monitored, will result in low latency predictions and a privacy-preserving operation since most of the benign samples are filtered out by the anomaly detector and not forwarded to the next stage, thus never leaving the local premises. On the other hand, this allows the deployment of multiple unique anomaly detectors trained on local data to learn a representation of normal behavior specific to the monitored (sub)network. As a result, a higher degree of flexibility is achieved, potentially improving the global classification performance. For example in our simple enterprise network, what is benign in the private cloud might not be in the sales and R&D department or vice-versa. Similarly, certain (sub)networks might require more strict thresholds. An unique anomaly detector specifically developed and configured for each (sub)network can then offer a solution.

The threshold τ_B previously introduced in the first stage to differentiate benign from attack, now also influences the overall computational and bandwidth requirements because τ_B controls the number of samples forwarded to the next stage for further analysis, which need to be transmitted over the network. Since the extension layer has the ability to correct benign samples falsely classified as malicious in the first layer, the thresholds τ_B and τ_U allow to balance both the computational requirements and classification performance.

In earlier work [30] we simulated such a three-tier IDS and optimized the capacity and task allocation of the individual stages using simulated annealing and queueing theory. The study concluded that either a single edge or dual edge-cloud deployment are optimal for the lowest delay and stable performance, with the latter being favored by low cloud computing costs.

IV. Methodology

This section describes the applied methodology to evaluate the novel multi-stage approach for hierarchical intrusion detection using a modern flow-based network dataset and contains all the necessary information to reproduce the reported results in Section V. First, Section IV-A introduces the dataset used for validation together with the applied preprocessing steps. Next, the algorithms used in both the first and second stages are presented before the evaluation strategy is laid out in Section IV-C. At last, the hardware specification on which the experiments are executed is given in Section IV-D.

A. Data

The CIC-IDS-2017 dataset is a modern flow-based network intrusion dataset developed by Sharafaldin et al. [4]. Before generating CIC-IDS-2017, Sharafaldin et al. already developed NSL-KDD, an altered version of the most popular intrusion detection dataset of the last decades, KDD99 [31]. NSL-KDD resolved many of the found issues present in the original version developed by the Defense Advanced Research Projects Agency (DARPA). In 2016, a list with 11 criteria was published that a proper intrusion detection dataset needs to satisfy [32]. The CIC-IDS-2017 was built from scratch to be the first to successfully fulfill all 11 criteria, justifying the use of this dataset for benchmark purposes. The generation of the dataset was spanned over a period of 5 days using 14 machines. The dataset consists of both benign and malicious traffic. The benign traffic is simulated using B-profiles, derived from the benign behavior of a group of 25 humans using statistical techniques and machine learning. On the other hand, the malicious traffic is generated by executing existing attack tools at specific time windows. The benign and malicious traffic is combined into a single dataset and distributed in machine-learning friendly flow-based CSV files and packet captures (PCAP). CICFlowMeter [33] is used to generate the bidirectional flows from the raw PCAP files. A biflow

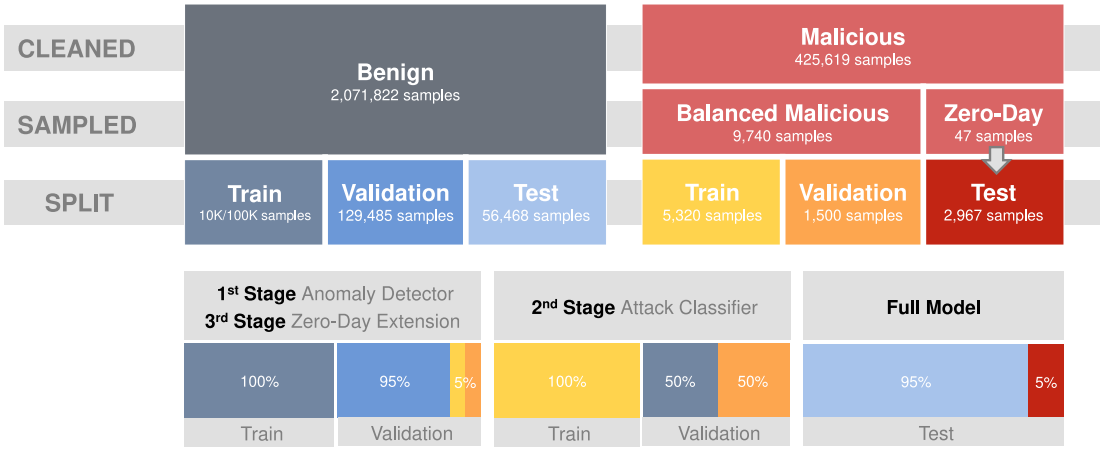


Fig. 3. Visualization of the used train, validation and test strategy for all the stages and final multi-stage approach.

TABLE II
ORIGINAL AND DOWN-SAMPLED ATTACK OCCURRENCES IN CIC-IDS-2017

Attack Class	Details	Original	Sampled
Benign	ALL	2,071,822	2,071,822
(D)DOS	ALL	3,216,637	1,948
	Hulk	172,726	1,046
	DDOS	128,014	775
	GoldenEye	10,286	63
	DoS slowloris	5,383	33
	Slowhttptest	5,228	31
Port scan	ALL	90,694	1,948
Brute Force	ALL	9,150	1,948
	FTP-Patator	5,931	1,263
	SSH-Patator	3,219	685
Web-Attack	ALL	2,143	1,948
	Brute Force	1,470	1,336
	XSS	652	593
	SQL Injection	21	19
Botnet	ALL	1,948	1,948
Unknown	ALL	47	47
	Infiltration	36	36
	Heartbleed	11	11

aggregates and computes 80 statistical network features over all the sent packets within a single connection and is identified by the source and destination IP address and port as well as a timestamp.

The original CIC-IDS-2017 dataset is cleaned in three steps by removing columns with redundant information, dropping rows with missing or infinity values, and eventually, the resulting dataset is filtered from duplicates. After cleaning, the data is scaled by normalizing the features using the StandardScaler implementation from scikit-learn [34]. The exact same cleaning and scaling approach is described more in-depth in previous work [35]. Following best practices to avoid any bias, it is important to note that a separate scaler is used to normalize the data for the anomaly detector and multi-class classifier fitted on their respective training set.

An overview of the number of occurrences for each of the attack classes and benign traffic of the cleaned data is given in Table II. This study aggregated the original attack classes into six more high-level attack categories. For example, both FTP- and SSH-patator are categorized as brute force. Because the dataset is highly unbalanced, containing mostly (D)DOS traffic, a sampling technique is used. The minimum number of occurrences of the newly created categories, 1948 in the botnet category, is used to downsample the other categories to the same number so that all attack categories are equally represented in the final dataset. A stratified random sampling technique is chosen because of the abundant number of samples in the majority classes. Since only 11 and 36 samples are present in the original dataset for the attack class infiltration and heartbleed, respectively, these two classes are well suited to be used as a proxy for unknown or zero-day attacks. As a result, the unknown category contains 47 samples, which will only be used to evaluate the proposed approach zero-day detection.

Figure 3 visualizes the train, validation, and test split strategy for each of the stages and eventually the complete multi-stage approach. The first two rows indicate the applied downsampling technique. Afterward, the resulting dataset is split into three parts for training, validation, and testing. The down-sampled dataset with malicious samples is split into 70% train and 30% test. The 47 zero-day samples are added to the test set with malicious samples. From the malicious train set, 300 samples for each attack class are sampled, resulting in a validation set containing 1,500 samples. The cleaned benign samples are also split into train, validation, and test sets to match the required distribution between benign and malicious in the final datasets. Eventually, the composition of the train and validation sets for the individual stages, together with the test is presented. The anomaly detector in the first stage is trained using only benign data, respectively, 10,000 and 100,000 samples for the OC-SVM and AE. On the other hand, the validation set is composed of 95% benign and only 5% malicious traffic. An unbalanced validation set is chosen because the anomaly detector needs to filter malicious traffic out of a stream of mainly benign traffic. On the contrary, the classifier in the second stage is trained using a balanced set containing only malicious traffic. This is important to prevent any bias towards a known attack category. Since no samples

of unknown attacks exist, they can not be used to validate the classifier in the second stage. Instead, benign traffic is used in such a way that classifiers that output a vector with low probabilities for each of the trained known attacks are rewarded with a higher validation score. The validation set for the second stage consists equally of benign and malicious samples. Finally, the test set consists of 95% benign and 5% malicious traffic not used before to train or validate any of the stages, simulating a realistic network stream of mainly benign traffic. Important to note is that all train, validation, and test sets are sampled in a stratified manner from the previously balanced dataset.

The CSE-CIC-IDS-2018 dataset, the successor of the CIC-IDS-2017, is generated using the same tools but deployed in the cloud rather than on a local university network. From the 2018 dataset 127,844 additional infiltration samples are obtained and cleaned in an identical manner as described before. These samples are kept separately to test the robustness of the zero-day capability of our novel proposed approach.

B. Algorithms

1) Anomaly Detection: The first stage relies on an anomaly detector to filter the malicious samples from benign traffic. Unsupervised machine learning models are well suited for this task since they model the normal baseline traffic and everything deviating from this is flagged as an anomaly. Since the model only learns the benign baseline from data, it is capable to detect known as well as unknown attacks. This study evaluates the use of both an autoencoder (AE) and one-class support vector machine (OC-SVM) as anomaly detector in the first stage.

An AE is a neural network composed of an encoder and decoder. The encoder projects the input vector onto a lower-dimensional or latent space. The decoder reconstructs the input vector as close as possible from this latent space. The reconstruction error, calculated as the sum of squared errors (SSE), is used as an anomaly score. In case the encoder and decoder are built from more than one layer, it is called a deep autoencoder (DAE). The Keras framework [36] on top of Tensorflow [37] is used for the experimental implementation. The following hyper-parameters are optimized during training: number of hidden layers, number of neurons per layer, activation function, and regularisation terms.

An OC-SVM is a special SVM that instead of separating two classes using a hyper-plane, encloses a single class as tight as possible using a hyper-sphere. The use of the radial basis function as (rbf) kernel function allows a complex, non-linear boundary for this hyper-sphere. The scikit-learn library is used for the implementation. The input features are first reduced using a PCA transformation. During training, the number of components in the PCA transformation, kernel coefficient, and regularization parameter is optimized.

2) Multi-Class Detection: The second stage classifies forwarded suspicious samples from the first stage using a multi-class classifier to a known attack class. Supervised machine learning models are well suited to learn this representation from labeled data. This study evaluates two classifiers: a random forest (RF) and a neural network (NN).

An RF is an ensemble of decision trees where each tree is built using the whole or sub-sample of the dataset. The final prediction is defined as the average output of all the trees, or in the case of classification as the most predicted class. The hyper-parameters considered during optimization are the number of trees, sub-sample percentage, and the number of features used for each split in a single tree. The scikit-learn implementation is used in this study.

A NN is composed of an input layer, one or more hidden layers, and an output layer. Each layer consists of multiple nodes which are connected with a certain weight to the nodes of the next layer. Data is sent from one layer to another if the value is above a particular threshold, defined by the activation function. The structure and name are inspired by the human brain and form the basis of deep learning algorithms. The number of layers, the respective number of nodes per layer, and a regularization term are optimized during hyperparameter tuning. For the implementation, the Keras framework on top of TensorFlow is used.

C. Evaluation Strategy

The novel multi-stage approach for hierarchical intrusion detection proposed in this study is composed of three stages. Before the performance of the complete IDS can be analyzed and compared with both a single model baseline implementation and existing state-of-the-art multi-stage approaches, the individual models in the stages and their corresponding thresholds need to be defined. Both the anomaly detector in the first stage and multi-class classifier in the second stage are trained individually using the training and validation set introduced in Section IV-A. The optimal hyper-parameters for each model, as described in Section IV-B, are obtained by performing hyper-parameter optimization with as validation metrics the AUROC and weighted F1 score for the first and second stage, respectively. Optuna [38], an open-source optimization framework, is used for the implementation with a Tree-structured Parzen Estimator (TPE) as sampling algorithm. The possible thresholds corresponding to a model are also computed on the same validation set. For each model in the first stage, ten possible values for τ_B are proposed, selected by the anomaly score corresponding to the maximum F1 to F9 score. The threshold used in the extension layer, τ_U is also computed on the same anomaly score but is selected using quantiles on the anomaly score of the benign flows in the validation set. Four possible candidates are examined using the 0.995, 0.99, 0.975, and 0.95 quantiles. Finally, only a single candidate for the threshold τ_M in the second stage is proposed, computed by the cut-off value on the confidence of the prediction achieving the maximum weighted f1 score.

The ten instances for each algorithm with the highest validation score and their corresponding thresholds are further analyzed for their use in the multi-stage approach. The final performance on the test set is evaluated for each permutation of the two models, anomaly detection and attack classification, and the three values for the applied thresholds. These

TABLE III
BEST RESULTS FIRST STAGE: ANOMALY DETECTION

Algorithm	AUROC	AUPR	F1	F2	F3	F4	F5	F6	F7	F8	F9	Training (s)	Inference (s)
AE	0.9117	0.3265	0.3984	0.5650	0.6666	0.7351	0.7863	0.8370	0.8715	0.8958	0.9135	39.671 ± 2.046	1.785 ± 0.017
OC-SVM	0.8947	0.3256	0.3893	0.5016	0.6426	0.7267	0.7823	0.8276	0.8584	0.8804	0.8976	0.771 ± 0.021	8.117 ± 0.277

TABLE IV
BEST RESULTS SECOND STAGE: ATTACK CLASSIFICATION

Algorithm	F1 weighted	F1 macro	Accuracy	Balanced Accuracy	Training (s)	Inference (s)
RF	0.9870	0.9094	0.9846	0.9654	1.249 ± 0.034	0.610 ± 0.005
NN	0.9525	0.7096	0.9403	0.9113	4.885 ± 0.189	1.555 ± 0.190

permutations are then ranked and analyzed based on several criteria such as accuracy, bandwidth requirements, and ability to detect unknown attacks. Eventually, the performance of the complete multi-stage approach is compared with a baseline RF implementation and previous state-of-the-art multi-stage approach as described by Bovenzi et al. [16]. The baseline RF model is trained on a mixed dataset of benign and malicious samples, using an open-set approach to add the ability to detect zero-day attacks. The threshold used by the baseline model to predict a sample as a zero-day is computed analogously to the threshold τ_M in the second stage.

Eventually, the robustness of the zero-day detection is evaluated by classifying the additional obtained infiltration samples from the CSE-CIC-IDS-2018 dataset. The fully optimized model, trained on data from the CIC-IDS-2017 dataset, is used to predict all the additional infiltration samples. A robust model is expected to transfer the zero-day detection capabilities with only a small drop in performance.

D. Hardware Setup

The experimental results in this study are obtained on GPULab, a distributed job-based platform hosted and maintained by the university department IDLab in collaboration with IMEC. All the experiments executed are submitted as a job that runs in an isolated container built upon a basic Python 3.7 Docker image with all the needed libraries installed. Each job was assigned 4 CPUs, Intel Xeon Silver 4108 CPU @ 1.80GHz and 16GB of RAM.

V. RESULTS

A. Anomaly Detection

An overview of the classification performance using the recommended metrics for binary detection, as well as training and inference timings, are given in Table III. The table contains two records with the test results of respectively the AE and OC-SVM model achieving the highest validation AUROC score for the first stage. An AE with a single hidden layer consisting of 42 nodes and a regularisation term of $2.45e-5$ achieved the highest AUROC of 0.9062 during hyper-parameter optimization on the validation set of stage 1 after 7 epochs. On the test set, this resulted in an AUROC of 0.9117. The fully optimized OC-SVM scored a slightly lower AUROC on both the validation and test set with, respectively, 0.8931 and 0.8947, but it eventually performs better in combination with the next stages in the multi-stage approach. The best hyper-parameters for the OC-SVM in the first stage are a kernel coefficient of 0.0633, a regularisation term of $2.317e-4$, and PCA transformation with 56 components. Both models are able to process the test set just shy of 60 thousand samples in a matter of seconds, in other words, a single sample takes less than a millisecond. Even with 2 orders of magnitude difference in training time between the AE and OC-SVM, both models are suitable to be practically used because training is not required to happen in real-time on a resource-constrained device. The main reason for the gap in training time is the number of used training samples, respectively 10 thousand and 100 thousand for the OC-SVM and AE. For comparison purposes, the OC-SVM is also retrained on the identical dataset as the AE resulting in a similar classification performance but increased training time of $273.0 \pm 0.2s$. This confirms our previous work [39] showing that models quickly converge even with limited available training samples on academic intrusion datasets. The non-linear decrease in training time combined with no classification degradation when reducing the number of training samples, allowed us to perform the hyper-parameter optimization for the OC-SVM without limiting the search space using a smaller training set.

B. Multi-Class Classification

Table IV summarizes the results in the second stage for the RF and NN model achieving the highest classification performance on the test set after hyper-parameter optimization. The RF outperforms the NN across all metrics for the task at hand. The best performing random forest consists of 97 fully grown trees with only 90% of the samples used to train each individual tree and with 12 features considered at each split. This RF achieved a weighted F1 score of 0.9710 and 0.9870 on the validation and test set, respectively. The NN consisting of a single layer with 41 nodes, 0.0379 as regularisation term and 17 epochs of training, achieves a weighted F1 of 0.9203 on the validation set of stage 2 and 0.9525 on the test set. The macro F1 score dropped from 0.9198 to 0.7096 for the validation and test set, respectively. This is because the test set in contrary to the validation set consists for 50% of benign data. Therefore, even a small percentage of benign traffic classified as one of the known attacks can heavily affect the F1 score associated with the known attacks and thus also the macro F1 score. Even with the NN model classifying less than half of the number of samples in the same time period as the RF

TABLE V
RESULTS FULL MULTI-STAGE APPROACH

	τ_B	τ_M	τ_U	F1 weighted	F1 macro	Accuracy	Bal. Accuracy	Bandwidth reduction	Zero-day recall	Inference (s)
Max F-score	F5-8	F1	0.995	**0.9897**	**0.8276**	**0.9877**	0.8954	68.75%	0.5957	7.808 ± 0.009
Max bACC	F9	F1	0.95	0.9580	0.7496	0.9341	**0.9608**	57.91%	**0.9574**	8.043 ± 0.065
Balanced	F5-8	F1	0.99	0.9875	0.8231	0.9834	0.9342	68.75%	0.8723	7.882 ± 0.054
RF Baseline	-	-	-	0.9849	0.7981	0.9832	0.8877	-	0.8936	1.525 ± 0.013
Bovenzi et al. [16]	F3	F1	-	0.9383	0.7549	0.8957	0.8550	**86.22%**	**0.9574**	6.969 ± 0.399

model, both models are well suited to be implemented in a high-speed network environment, with a single sample taking less than a fraction of a millisecond of processing time. The training time for the RF and NN currently only takes a few seconds, allowing for more training samples if available.

C. Full Model Performance

The performance of the complete multi-stage model can be assessed based on several criteria. First, the overall classification performance of the three stages combined. Next, the ability to not only reduce the computation requirements but also decrease the bandwidth requirements in case of a hierarchical deployment, and lastly, the capability to detect unknown attacks. Unfortunately, not a single permutation of models and thresholds scores the highest on all of the criteria but a trade-off needs to be made using the defined thresholds τ_B, τ_M, and τ_U. The best performing permutations are consistently composed of the best performing OC-SVM and RF models from the first and second stage, respectively.

Table V gives an overview of the results for three interesting permutations of the complete multi-stage model, the single model baseline and the current state-of-the-art multi-stage approach for hierarchical intrusion detection. The first row presents the results of the permutation achieving the highest classification performance in terms of accuracy and both weighted and macro F1 scores. This permutation used the anomaly score corresponding to the maximum F5 to F8 score, an identical anomaly score, in the first stage as a value for τ_B. The anomaly score corresponding to the 0.995 quantiles is used as the value for τ_U in the extension stage. Using these thresholds, a maximum weighted F1 score is achieved of 0.9897 and an accuracy of 0.9877 on the test set, but only 59.57% of the unseen attacks are successfully classified as a zero-day attack when using the complete multi-stage approach. In case it is deployed in a hierarchical manner a bandwidth reduction of almost 69% is achieved between the local anomaly detector and centralized attack classifier, in comparison with an IDS forwarding all its traffic to a centralized system.

The second row in the table presents the results of the permutation scoring the highest on the balanced accuracy with a value of 0.9608. This configuration also achieves an extremely high recall of zero-day attacks, with 45 out of 47 samples, while preserving high overall classification scores. Contrary to the first row, a higher value is used for τ_B, corresponding to the anomaly score with the maximum F9 score in the first stage, leading to more traffic forwarded to the next stage. On the other hand, the value for τ_U is decreased to the anomaly score corresponding to the 0.95 quantiles. This results in a more swift classification as zero-day and thus higher recall of zero-day attacks.

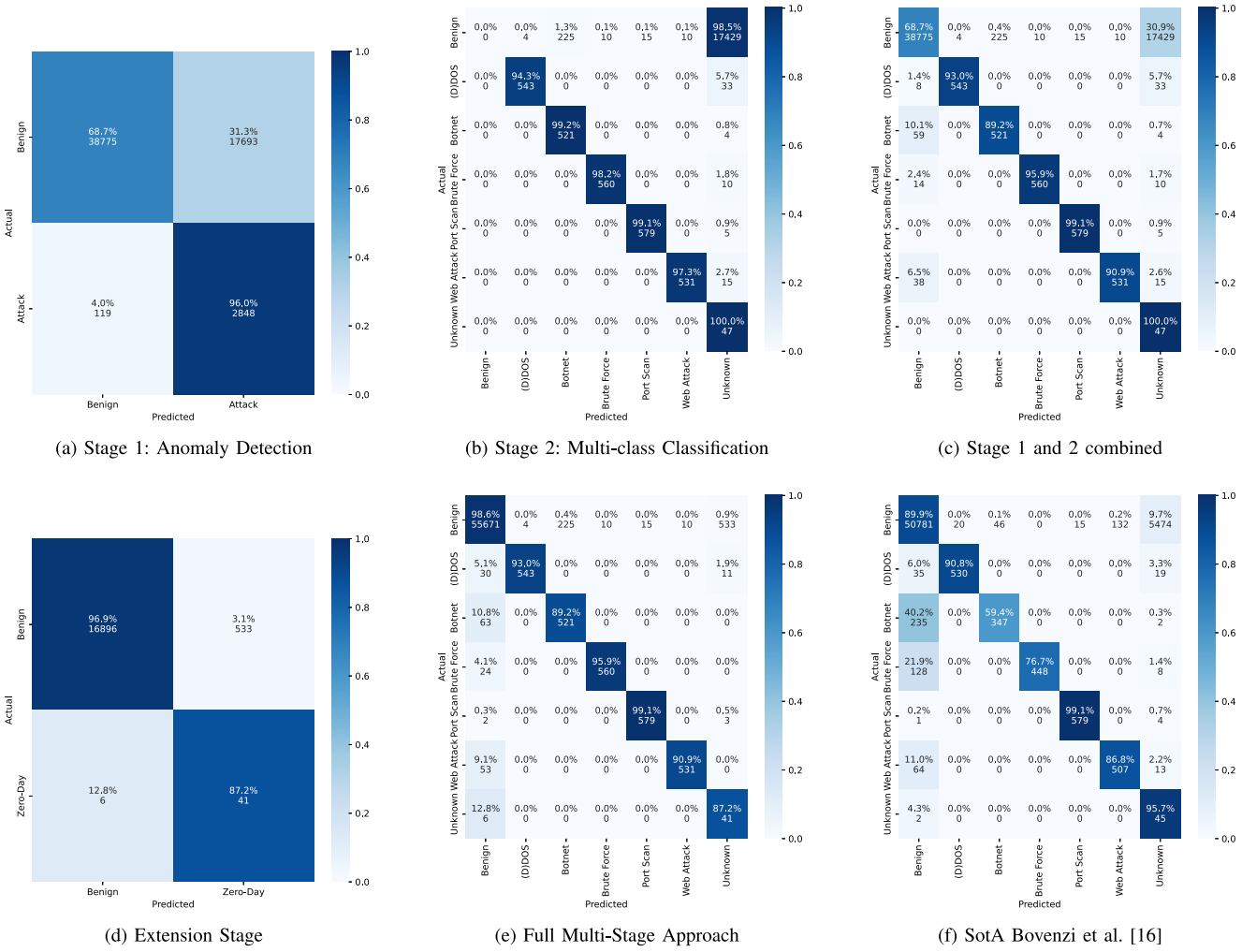
The third row presents the results for a permutation balancing between the high classification performance of the first row and the high zero-day recall of the second row. For both the threshold τ_B and τ_U, a value is chosen in between the thresholds selected in the first and second row. This permutation serves as a middle ground in the challenge to balance the multiple trade-offs.

The performance results of the single model baseline can be found in the fourth row. There are no values present for the thresholds as well as for the bandwidth reduction since these are not relevant. The hyper-parameters obtained through optimization for the baseline RF are 57 fully grown trees with only 87% of the samples used to train each individual tree and with 30 features considered at each split. Additionally, the threshold used in the open-set classification is set to 0.93, yielding a zero-day recall of 89%.

The last row contains the performance metrics for the state-of-the-art multi-stage approach. There is no value for the threshold τ_U since their approach lacks an extension stage. Overall the approach by Bovenzi et al., which was not previously validated on CIC-IDS-2017, performs fairly well. Especially the bandwidth reduction and zero-day recall are high, while the other classification metrics are consistently out-preformed by our novel approach.

The last column in Table V lists the execution time to classify all the samples in the test. The single model RF baseline yields the lowest execution time, while the novel approach has the longest execution time.

Figure 4 presents six confusion matrices for each of the individual stages, intermediate result and final result of the complete multi-stage approach on the test set using the thresholds of the permutation balancing the multiple trade-offs together with the final confusion matrix of the existing multi-stage approach by Bovenzi et al. [16] applied to the CIC-IDS-2017 dataset. The confusion matrix in Figure 4(a) visualizes the true negatives, false positives, false negatives, and true positives in, respectively, the upper left, upper right, lower left, and lower right quadrant for the anomaly detector in the first stage. Only the positive samples are forwarded to the next stage, therefore already 119 samples or 4% of the fraud samples are misclassified which are unable to be corrected by the succeeding stages. Simultaneously, nearly 69% of the benign samples are correctly classified and prevented from propagating further in the system. Figure 4(b) contains the confusion matrix for all the suspicious classified samples by the anomaly detector in the first stage, corresponding to the



(a) Stage 1: Anomaly Detection

(b) Stage 2: Multi-class Classification

(c) Stage 1 and 2 combined

(d) Extension Stage

(e) Full Multi-Stage Approach

(f) SotA Bovenzi et al. [16]

Fig. 4. Confusion Matrices.

positives from the first confusion matrix. Most of the known, as well as unknown attacks, are correctly classified, while most of the benign samples are properly predicted as unknown and thus forwarded to the final extension stage. Before the confusion matrix of the extension layer is presented, the matrix for the intermediate result of the first and second stage combined is given in Figure 4(c). Except for the high number of benign samples classified as unknown, the results already look good. To correct the mistakes in the first stage to send these benign samples to the second stage, the extension stage was introduced. Figure 4(d) shows the confusion matrix for the extension stage. Clearly, most of the benign traffic is being successfully corrected and eventually classified as benign. The confusion matrix in Figure 4(e) eventually presents the results of the predictions of all three stages combined for our novel proposed approach. The last confusion matrix in Figure 4(f) is of the existing multi-stage approach applied to the test dataset. The results are similar to the confusion matrix in Figure 4(c) albeit the number of benign samples classified as unknown is lower but still significantly high. Furthermore, most attack classes are reliably detected except for botnet followed by brute-force attacks.

D. Robustness Zero-Day Detection

The robustness of the zero-day detection capabilities is evaluated on 127,844 additional samples from the CSE-CIC-IDS-2018 dataset. Our proposed approach correctly classifies 100,199 samples as unknown attack, equal to a recall of 78.38%. The baseline RF only succeed to classify 76 samples correctly as unknown, which is less than 0.01% recall. The approach by Bovenzi et al. classifies 111,215 samples correctly, resulting in a recall of 86.99%.

VI. Discussion

A. Improved Classification Over Baseline and State-of-the-Art

Section V and more specifically Table V present the results for three permutations of the proposed novel multi-stage approach, the open-set RF baseline, and the previously proposed multi-stage approach by Bovenzi et al.

When we compare our novel proposed multi-stage approach against the baseline RF and state-of-the-art approach, the classification metrics are consistently being out-preformed. Only the state-of-the-art approach can present the same zero-day

recall and even higher bandwidth reduction. This can be explained by the extension stage in our novel approach since it creates the opportunity for the model to correct mistakes of the first stage. As a result, the threshold τ_B in the first stage can be configured less strictly which has a negative effect on the bandwidth reduction but more significant improvement of classification performance. When bandwidth reduction is a priority, our novel approach also allows tuning τ_B to obtain at least an equal reduction.

Recent literature questions the marginal improvements made in studies to not improve performance in the real world because of the lack of generalization power [35], [40], [41]. As a result, the main takeaway is that the classification performance at least matches the state-of-the-art while providing additional features such as extra flexibility, zero-day detection, and bandwidth reduction.

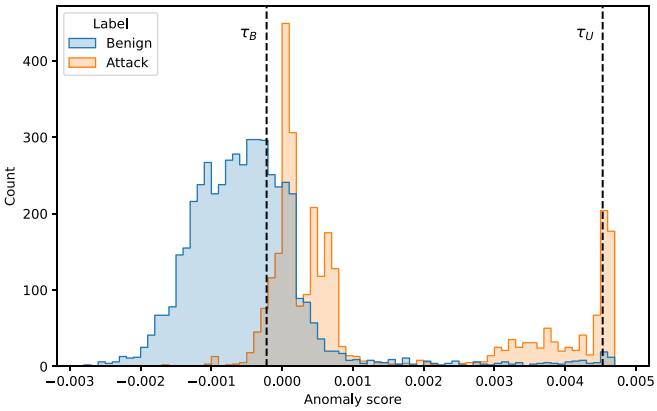


Fig. 5. Histogram of the anomaly score outputted by the first stage for both benign and fraud traffic with the possible thresholds τ_B and τ_U visualized.

B. Advantage of Extension Stage

The extension stage serves multiple purposes. First of all, it improves the classification performance by predicting samples classified as unknown by the second stage as benign. In Figure 4(c), more than 30% of all the benign samples are classified as unknown. Without an additional stage, the model would fail to correct them. Next, the extension stage permits the first stage to make mistakes. Therefore, the first stage can mark more samples as suspicious, knowing that forwarded benign samples can be corrected by the following stages. The improved classification performance comes at a tradeoff with the bandwidth requirement since forwarded samples need to be transmitted over the network in a hierarchical deployment.

C. Hierarchical Deployment

The novel multi-stage approach proposed in this study is designed for a scalable hierarchical architecture. Each of the individual stages can be deployed on its own location in the network, for example in the edge, fog, or cloud. Ideally, the anomaly detector is situated near the network being monitored. This way privacy is preserved during operation because mostly malicious data is forwarded from the local premises.

Furthermore, the bandwidth and computational reduction are achieved by applying a threshold on the anomaly score predicted by the first stage such that only a fraction of all samples are forwarded to the next stage, which requires transmission over a network in case of a n-tier deployment. Since only a small share of all data is being forwarded by the first stage, most samples are only processed by the lightweight filter skipping the potentially more computational expensive multi-class classifier in the second stage. Contrarily, recent studies such as Khan et al. [12] which employs a similar architecture consisting of binary detection followed by attack classification is less suited for a hierarchical deployment because the anomaly score is used as an extra feature for improved classification in the second stage and not as a lightweight filter.

The execution times in Table V show a small increase in overhead between the approach by Bovenzi et al. and our novel approach due to the addition of an extra stage. Evaluating the execution times between the different configurations of our novel approach only show small differences. But keep in mind that these results are obtained on a single machine, reiterating the same experiment with a n-tier deployment will also include the effect of the bandwidth reduction as shown in our previous work [30].

D. Thresholds τ_B, τ_M and τ_U

Figure 5 plots the distribution of both the benign and malicious samples in function of the anomaly score for the OC-SVM with the highest AUROC and eventually used as the anomaly detector in the complete multi-stage model. The values for τ_B and τ_U are marked using a vertical dashed line. All samples left from the first dashed line, thus having a lower anomaly score than the threshold τ_B are classified as benign, while all samples on the right side with an anomaly score higher than τ_B are forwarded to the next stage. Carefully selecting this threshold is crucial to obtaining proper results. If the threshold is set too low, many samples are forwarded to the next stage, consuming both computational resources and bandwidth with an additional risk of classifying benign samples wrongly as a known attack. When the threshold is set too high, the first stage classifies actual intrusion as benign. As a result, the threshold τ_B balances the computational and bandwidth requirements as well as final classification performance. The second dashed line visualizes the applied cut-off value to classify a sample as a zero-day attack. If the anomaly score is higher than τ_U and thus on the right side of the line, the sample is marked as a zero-day attack. However, if the anomaly score is between τ_B and τ_U the sample will be classified as benign in the extension stage. Setting the threshold τ_U too low will wrongly classify benign samples as zero-day attacks while setting it too high prevents zero-day attacks from being discovered. Accordingly, defining τ_U is essential to successfully detect zero-day attacks.

The threshold τ_M controls the cut-off on the confidence of the prediction by the attack classifier in the second stage. In case τ_M would be equal to zero, none of the samples would be classified as unknown and thus not forwarded to the extension stage. Rather, if τ_M is set to one, only samples where the attack classifier is absolutely sure will be classified as one

of the known attack classes. Consequently, setting the threshold too low allows loose classification to one of the known classes which will wrongly predict benign samples to one of the known attacks. Yet, setting the threshold too high will prevent actual attacks to be correctly classified.

E. Manual Selection of Thresholds

In this study, an optimization technique is used to select the optimal values for the thresholds τ_B, τ_M, and τ_U with mixed validation set consisting of both benign and malicious samples. This is a valid approach to obtaining good results on benchmark datasets but not always feasible in a real-world setting. The collection of labeled malicious samples is complex and time-demanding. The inclusion of benign samples in the validation set of the multi-class classifier breaks the privacy preserved training and becomes increasingly complex when samples of multiple (sub)networks need to be included.

Following rules of thumb can help with the initial configuration of the thresholds. Afterward, the performance tradeoff can be iteratively adjusted until a favorable configuration is obtained.

1) τ_B: This study optimized the value for the threshold yielding the maximal f-score. Interesting is that this f-score corresponds to a certain false positive ratio (fpr). When we look at the top hundred permutations with the highest global classification performance from Section IV-C, then we find that the average and median fpr is equal to 0.2259 and 0.2107 $\pm$ 0.0661, respectively. As a result, a value for the threshold τ_B can be selected using only benign samples.

2) τ_U: The same approach as with τ_B can be taken for τ_U with as difference that the selected fpr is an upper limit for the fpr of the whole multi-stage approach. Both the median and average fpr used in the top hundred best-performing permutations are 0.995 $\pm$ 0.001.

3) τ_M: The selection of τ_M can also intuitively be done but often vary depending on the used multi-class classifier. Most classifiers produce a probability value associated with each of the classes, the threshold then determines the minimum required confidence to classify a sample as a known attack. The mean and average τ_M for the best performing classifiers after optimization are respectively, 0.95 and 0.93 $\pm$ 0.03.

F. Adaptability

All the thresholds τ_B, τ_M, and τ_U allow live adjustment to dynamically balance the trade-offs in the individual stages, and as a result, adapt the final classification performance without the need for retraining the machine-learning models.

Our novel approach has not been designed specifically to tackle concept-drift challenges in mind, but rather to enable these performance trade-offs in real-time. Although, we expect this approach to withstand concept-drift to a certain degree, for example, a radical shift of the underlying benign traffic will at least require retraining of the anomaly detector in the first stage.

G. Robustness of Zero-Day Detection

A limitation of our study is that only 47 samples from the CIC-IDS-2017 dataset are used as unknown or zero-day attack. Therefore, the robustness of the zero-day detection capabilities is evaluated on 127,844 additional infiltration samples extracted from the CSE-CIC-IDS-2018 dataset. The results show that the baseline RF fails completely to maintain a similar zero-day detection capability with a recall that dropped from almost 90% to less than 0.01%. On the contrary, the approach by Bovenzi et al. and our novel approach are able to maintain the majority of their zero-day detection capabilities with a drop in recall of less than 10%. As a result, a multi-stage approach is not only able to achieve a higher zero-day detection rate but is also more robust.

VII. Future Work

The experimental implementation of the multi-stage approach for hierarchical intrusion detection only relies on network features as input while the general proposed architecture is also applicable to both host-based and hybrid IDS. Future work could extend this study by evaluating the proposed approach on multiple input sources. For instance, ensemble techniques can then be applied to aggregate the output of a machine-learning model on each of the input vectors.

Recently published work [42], [43] together with our previous work demonstrates that most, if not all, currently proposed models trained on network IDS datasets lack generalization strength. Future work should evaluate if a multi-stage model is more resilient against this classification performance degradation. Moreover, our proposed approach is capable of having a separate anomaly detector specifically trained for each subnetwork in a hierarchical deployment. This enables the simulation of a more realistic diverse network, for example consisting of both IoT and general purpose devices, using a multi-data set evaluation.

Parallel to this study, we have evaluated 10 possible task allocations, which assign to each task a capacity in a three-tier network consisting of the edge, fog, and cloud. A simulation is performed with queueing theory, resulting in multiple optimal configurations depending on specific requirements [44]. A follow-up on this work will try to confirm the theoretical results using an experimental setup of a multi-stage IDS on a multi-tier architecture.

VIII. Conclusion

In this study, a novel multi-stage approach for hierarchical intrusion detection is proposed. First, the general architecture is introduced and the design choices made, are justified. The strengths of the new approach are the high adaptability without the necessity to retrain any of the classifiers, the empowerment of an n-tier deployment to reduce the bandwidth and computational requirements, and the capability to detect zero-day attacks. Additionally, in the case of a hierarchical deployment, privacy is preserved during both training and operational service.

After the novel approach is introduced an experimental implementation is evaluated using two modern network intrusion datasets, CIC-IDS-2017 and CSE-CIC-IDS-2018. An AE and OC-SVM are evaluated for anomaly detection, while an RF and NN are evaluated for the attack classification. The experimental results show that the classification performance at least matches or even outperforms both the baseline single model and existing multi-stage approaches for most of the evaluated metrics. Besides the improved classification performance, the novel multi-stage approach proved its robust capability to detect unseen, zero-day attacks and the ability to reduce the computational and bandwidth requirements. The implementation of the novel multi-stage approach that balances the trade-off between a high zero-day recall, bandwidth reduction, and classification performance achieves a weighted F1 and balanced accuracy score of 0.9875 and 0.9342, respectively, as opposed to 0.9383 and 0.8550 for the state-of-the-art approach for multi-stage intrusion detection. In particular, 41 out of 47 or 87% zero-day samples were correctly classified from CIC-IDS-2017. Moreover the robustness of the zero-day detection is validated by correctly classifying 100,199 out of 127,844 or 78% zero-day samples from CSE-CIC-IDS-2018. The bandwidth was reduced by almost 69% between the local anomaly detector and centralized attack classifier in case of a hierarchical deployment in comparison with an IDS forwarding all its traffic to a centralized system. The specific results depend on the selected thresholds and are highly adaptable to obtain the desired trade-offs.

REFERENCES

[1] A. Jurcut, T. Niculcea, P. Ranaweera, and N.-A. Le-Khac, "Security considerations for Internet of Things: A survey," *SN Comput. Sci.*, vol. 1, pp. 1–19, Jun. 2020.

[2] T. Alam, "A reliable communication framework and its use in Internet of Things (IoT)," *Int. J. Sci. Res. Comput. Sci. Eng.*, vol. 3, pp. 450–456, May/Jun. 2018.

[3] E. Viegas, A. Santin, A. Bessani, and N. Neves, "BigFlow: Real-time and reliable anomaly-based intrusion detection for high-speed networks," *Future Gener. Comput. Syst.*, vol. 93, pp. 473–485, Apr. 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X18307635

[4] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Security Privacy Vol. 1 (ICISSP)*, 2018, pp. 108–116.

[5] Y. Zhang, L. Wang, W. Sun, R. C. Green, II, and M. Alam, "Distributed intrusion detection system in a multi-layer network architecture of smart grids," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 796–808, Dec. 2011.

[6] A. Ali and M. M. Yousaf, "Novel three-tier intrusion detection and prevention system in software defined network," *IEEE Access*, vol. 8, pp. 109662–109676, 2020.

[7] L. Li, Y. Yu, S. Bai, Y. Hou, and X. Chen, "An effective two-step intrusion detection approach based on binary classification and k -NN," *IEEE Access*, vol. 6, pp. 12060–12073, 2018.

[8] H. H. Pajouh, G. Dastghaibyfard, and S. Hashemi, "Two-tier network anomaly detection model: A machine learning approach," *J. Intell. Inf. Syst.*, vol. 48, no. 1, pp. 61–74, Feb. 2017.

[9] H. H. Pajouh, R. Javidan, R. Khayami, A. Dehghantanha, and K.-K. R. Choo, "A two-layer dimension reduction and two-tier classification model for anomaly-based intrusion detection in IoT backbone networks," *IEEE Trans. Emerg. Topics Comput.*, vol. 7, no. 2, pp. 314–323, Apr.–Jun. 2019.

[10] W. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, "Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system," *Expert Syst. Appl.*, vol. 67, pp. 296–303, Jan. 2017.

[11] S.-Y. Ji, B.-K. Jeong, S. Choi, and D. H. Jeong, "A multi-level intrusion detection method for abnormal network behaviors," *J. Netw. Comput. Appl.*, vol. 62, pp. 9–17, Feb. 2016.

[12] F. A. Khan, A. Gumaei, A. Derhab, and A. Hussain, "A novel two-stage deep learning model for efficient network intrusion detection," *IEEE Access*, vol. 7, pp. 30373–30385, 2019.

[13] Divyatmika and M. Sreekesh, "A two-tier network based intrusion detection system architecture using machine learning approach," in *Proc. Int. Conf. Electr. Electron. Optim. Techn. (ICEEOT)*, Mar. 2016, pp. 42–47.

[14] M. F. Umer, M. Sher, and Y. Bi, "A two-stage flow-based intrusion detection model for next-generation networks," *PLoS One*, vol. 13, no. 1, Jan. 2018, Art. no. e0180945.

[15] Y. Abuadlla, G. Kvascev, S. Gajin, and Z. Jovanovic, "Flow-based anomaly intrusion detection system using two neural network stages," *Comput. Sci. Inf. Syst.*, vol. 11, no. 2, pp. 601–622, 2014, doi: 10.2298/CSIS130415035A.

[16] G. Bovenzi, G. Aceto, D. Ciuonzo, V. Persico, and A. Pescapé, "A hierarchical hybrid intrusion detection approach in IoT scenarios," in *Proc. IEEE Global Commun. Conf.*, Dec. 2020, pp. 1–7.

[17] L. Yang, A. Moubayed, and A. Shami, "MTH-IDS: A multitiered hybrid intrusion detection system for Internet of Vehicles," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 616–632, Jan. 2022.

[18] M. Injadat, A. Moubayed, A. B. Nassif, and A. Shami, "Multi-stage optimized machine learning framework for network intrusion detection," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 2, pp. 1803–1816, Jun. 2021.

[19] R. Coulter, Q.-L. Han, L. Pan, J. Zhang, and Y. Xiang, "Data-driven cyber security in perspective—Intelligent traffic analysis," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3081–3093, Jul. 2020.

[20] Y. Miao, C. Chen, L. Pan, Q.-L. Han, J. Zhang, and Y. Xiang, "Machine learning–based cyber attacks targeting on controlled information: A survey," *ACM Comput. Surv.*, vol. 54, no. 7, pp. 1–36, Jul. 2021. [Online]. Available: https://doi.org/10.1145/3465171

[21] I. H. Sarker, A. S. M. Kayes, S. Badsha, H. Alqahtani, P. Watters, and A. Ng, "Cybersecurity data science: An overview from machine learning perspective," *J. Big Data*, vol. 7, no. 1, p. 41, Jul. 2020.

[22] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: An ensemble of autoencoders for online network intrusion detection," 2018, *arXiv:1802.09089*.

[23] S. Zavrak and M. İskefiyeli, "Anomaly-based intrusion detection from network flow features using variational autoencoder," *IEEE Access*, vol. 8, pp. 108346–108358, 2020.

[24] Q. T. Nguyen, K. P. Tran, P. Castagliola, T. T. Huong, M. K. Nguyen, and S. Lardjane, "Nested one-class support vector machines for network intrusion detection," in *Proc. IEEE 7th Int. Conf. Commun. Electron. (ICCE)*, 2018, pp. 7–12.

[25] I. Ahmad, M. Basheri, M. J. Iqbal, and A. Rahim, "Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection," *IEEE Access*, vol. 6, pp. 33789–33795, 2018.

[26] P. A. A. Resende and A. C. Drummond, "A survey of random forest based methods for intrusion detection systems," *ACM Comput. Surv.*, vol. 51, no. 3, pp. 1–36, May 2018. [Online]. Available: https://doi.org/10.1145/3178582

[27] R. Vinayakumar, K. P. Soman, and P. Poornachandran, "Applying convolutional neural network for network intrusion detection," in *Proc. Int. Conf. Adv. Comput. Commun. Inform. (ICACCI)*, 2017, pp. 1222–1228.

[28] P. Wu and H. Guo, "LuNet: A deep neural network for network intrusion detection," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, 2019, pp. 617–624.

[29] M. Al-Zewairi, S. Almajali, and A. Awajan, "Experimental evaluation of a multi-layer feed-forward artificial neural network classifier for network intrusion detection system," in *Proc. Int. Conf. New Trends Comput. Sci. (ICTCS)*, 2017, pp. 167–172.

[30] Y.-C. Lai et al., "Task assignment and capacity allocation for ML-based intrusion detection as a service in a multi-tier architecture," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 1, pp. 672–683, Mar. 2023.

[31] "KDD cup 1999 data." University of California. Oct. 1999. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

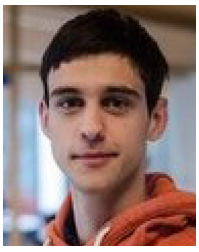
[32] A. Gharib, I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "An evaluation framework for intrusion detection dataset," in *Proc. Int. Conf. Inf. Sci. Security (ICISS)*, Dec. 2016, pp. 1–6.

[33] "CICFlowMeter." University of California. 2022. [Online]. Available: https://github.com/ahlashkari/CICFlowMeter

[34] F. Pedregosa et al., "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.

[35] M. Verkerken, L. D'hooge, T. Wauters, B. Volckaert, and F. De Turck, "Towards model Generalization for intrusion detection: Unsupervised machine learning techniques," *J. Netw. Syst. Manag.*, vol. 30, no. 1, p. 12, Oct. 2021.
[36] F. Chollet et al. "Keras." 2015. [Online]. Available: https://keras.io
[37] M. Abadi et al. "TensorFlow: Large-scale machine learning on heterogeneous systems." tensorflow.org. 2015. [Online]. Available: https://www.tensorflow.org/
[38] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proc. 25rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2019, pp. 2623–2631.
[39] L. D'hooge, T. Wauters, B. Volckaert, and F. De Turck, "Classification hardness for supervised learners on 20 years of intrusion detection data," *IEEE Access*, vol. 7, pp. 167455–167469, 2019.
[40] L. D'hooge, T. Wauters, B. Volckaert, and F. De Turck, "Inter-dataset generalization strength of supervised machine learning methods for intrusion detection," *J. Inf. Security Appl.*, vol. 54, p. 13, Oct. 2020, doi: 10.1016/j.jisa.2020.102564.
[41] C. F. T. Pontes, M. M. C. de Souza, J. J. C. Gondim, M. Bishop, and M. A. Marotta, "A new method for flow-based network intrusion detection using the inverse Potts model," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 2, pp. 1125–1136, Jun. 2021.
[42] S. Layeghy and M. Portmann, "On generalisability of machine learning-based network intrusion detection systems," May 2022, *arXiv:2205.04112*.
[43] M. Catillo, A. Del Vecchio, A. Pecchia, and U. Villano, "Transferability of machine learning models learned from public intrusion detection datasets: The CICIDS2017 case study," *Softw. Qual. J.*, vol. 30, pp. 955–981, Mar. 2022. [Online]. Available: https://doi.org/10.1007/s11219-022-09587-0
[44] Y.-C. Lai et al., "Machine learning based intrusion detection as a service: Task assignment and capacity allocation in a multi-tier architecture," in *Proc. 14th IEEE/ACM Int. Conf. Utility Cloud Comput. Companion*, 2021, pp. 1–6. [Online]. Available: https://doi.org/10.1145/3492323.3495613

Miel Verkerken received the M.Sc. degree in information engineering technology in 2018, after which he first gained some international professional experience before starting as a Researcher. He is currently pursuing the Ph.D. degree with the Internet and Data Science Lab (IDLab-imec), Ghent University and has been a Teaching Assistant since September 2019. His research interests lie in the field of cybersecurity and machine learning, with a specific focus on applying AI to intrusion detection systems.



Laurens D'hooge received the M.Sc. degree in information engineering technology from Ghent University in 2018, where he is currently pursuing the Ph.D. degree with Internet and Data Science Lab (IDLab-imec). His area of research lies at the intersection of cybersecurity, more specifically network security, and applied machine learning.

Didik Sudyana received the M.S. degree in informatics from Universitas Islam Indonesia, Indonesia, in 2016. He is currently pursuing the Ph.D. degree with the Electrical Engineering and Computer Science International Graduate Program, National Yang Ming Chiao Tung University. He is a Lecturer and a Researcher of Informatics with STMIK Amik Riau, Indonesia. His research interests include cybersecurity, machine learning, and network design and optimization.

Ying-Dar Lin (Fellow, IEEE) received the Ph.D. degree in computer science from the University of California at Los Angeles in 1993. He is the Chair Professor of Computer Science with National Yang Ming Chiao Tung University, Taiwan. He was a Visiting Scholar with Cisco Systems, San Jose, from 2007 to 2008, a CEO with Telecom Technology Center, Taiwan, from 2010 to 2011, and a Vice President of National Applied Research Laboratories, Taiwan, from 2017 to 2018. He cofounded L7 Networks Inc., in 2002, and O'Prueba Inc., in 2018. His research interests include cybersecurity, wireless communications, network softwarization, and machine learning for communications. He has served or is serving on the editorial boards of several IEEE journals and magazines, including Editor-in-Chief of IEEE COMMUNICATIONS SURVEYS AND TUTORIALS from 2017 to 2020.

Tim Wauters (Member, IEEE) received the M.Sc. and Ph.D. degrees in electro-technical engineering from Ghent University, in 2001 and 2007, respectively. He has been working as a Postdoctoral Fellow of F.W.O.-V. with the Department of Information Technology, Ghent University. He is currently active as a Senior Researcher with imec. His work has been published in more than 150 scientific publications. His research interests include design and management of networked services, covering multimedia distribution, cybersecurity, big data, and smart cities.

Bruno Volckaert (Senior Member, IEEE) received the Ph.D. degree in resource management for grid computing from Ghent University, in 2006, where he is currently a Professor of Advanced Distributed Systems and a Senior Researcher with imec. He has worked on over 45 national and international research projects and is the author or coauthor of more than 150 peer-reviewed papers published in international journals and conference proceedings. His current research deals with reliable and high performance distributed software systems for a.o. Smart Cities, scalable cybersecurity detection and mitigation architectures, and autonomous optimization of cloud-based applications.

Filip De Turck (Fellow, IEEE) leads the Network and Service Management Research Group, Ghent University, Belgium, and imec. He has coauthored over 700 peer reviewed papers. His research interests include design of secure and efficient softwarized network and cloud systems. He was elevated as an IEEE Fellow for outstanding technical contributions. He is involved in several research projects with industry and academia, served as the Chair of the IEEE Technical Committee on Network Operations and Management and a Steering Committee Member of the IFIP/IEEE IM, IEEE/IFIP NOMS, IEEE/IFIP CNSM, and IEEE NetSoft conferences.