Cross-Evaluation of Deep Learning-based Network Intrusion Detection Systems

Ciro Guida*†, Alfredo Nascita*, Antonio Montieri*, Antonio Pescapé*
*University of Napoli "Federico II" (Italy) and †University of Bergamo (Italy)
{ciro.guida, alfredo.nascita, antonio.montieri, pescape}@unina.it

***Abstract*—Network Intrusion Detection Systems are essential tools for protecting networks against attacks. *Deep Learning* approaches are increasingly employed in developing these systems due to their versatility and effectiveness. However, the common procedure for training and testing Deep Learning models typically leverages traffic data entirely collected from the operational network managed by a single organization, posing privacy and security concerns in sharing these data. As a result, the assessment of the performance of these models in real-world scenarios is significantly hindered. On the other hand, given the wide variety of existing attacks and the emergence of new attack types, it is crucial to evaluate the robustness of Intrusion Detection Systems when the network context varies. Indeed, it is highly desirable that the effectiveness of trained Deep Learning models is not severely impacted when ported into other networks.**

To this aim, in this work, we exploit various single-modal and multimodal Deep Learning approaches and leverage a *cross-evaluation* procedure to assess their capability to distinguish malicious from benign traffic in different network contexts. Furthermore, we investigate the impact of various informative fields extracted from traffic on the generalization capability of models. Our cross-evaluation leverages three recent public-available network attack datasets related to diverse scenarios. The results obtained suggest that the availability at training time of traffic generated by attacks conducted in the operational network is crucial for designing a robust Intrusion Detection System that keeps working with minimal F1-score degradation, when the network context changes.

***Index Terms*—Anomaly Detection, Network Intrusion Detection Systems, Network Security, Cross-Evaluation, Deep Learning.**

I. Introduction

The dramatic growth of cyberattacks and their ever-evolving nature require the enforcement of timely and effective network security policies. Indeed, in the last years, severe cyberattacks are globally increased, also fueled by the Russo-Ukrainian (cyber-)war. Detailing, an increment of 53% in global attacks from the 1H 2018 to the 1H 2022 has been observed, with the malware category accounting for 38% of the share.[1] As a result, the impact of the cost of cybercrime for businesses is estimated that will amount to $\$10.5$ trillion by 2025.[2]

Therefore, monitoring the heterogeneous and highly-dynamic traffic flowing across networks and understanding its nature are paramount to guarantee network security (e.g., detection and classification of anomalies, attacks, or malware). Notably, *Network Intrusion Detection Systems (NIDSs)* are increasingly used to perform *Anomaly Detection (AD)*, aiming to distinguish between *benign* traffic generated by common/normal activities, and *malicious* traffic related to illegal/fraudulent ones. In particular, Machine Learning (ML), and especially *Deep Learning (DL)* approaches, have proven their effectiveness in addressing AD [1] being able to cope with modern traffic characteristics. Indeed, such peculiarities severely challenge more traditional methods based on port numbers and deep packet inspection due to the broad usage of non-standard ports and the wide adoption of traffic-encryption schemes, respectively. Moreover, while the successful use of ML approaches is constrained by the design and extraction of handcrafted domain-expert-driven features, DL ones benefit from end-to-end learning capabilities to directly extract highly-structured features from raw traffic data.

Nevertheless, DL approaches require a large amount of labeled data to train supervised models, whose collection is commonly burdensome and time-consuming. This process has to deal with the dynamic nature of network traffic, being a constantly moving target, and it is even harder when collecting malicious traffic, especially for rarely observed attacks with (partially) unknown dynamics. Also, user privacy and business sensitivity concerns limit the sharing of these data. In fact, despite the extensive exploration of DL-based NIDSs in the literature, their evaluation is usually conducted by selecting a single public-available dataset or through private (i.e. self-collected and unshared) datasets [2]. Unfortunately, these procedures raise some questions about the applicability of such proposals in real-world settings.

On the other hand, taking advantage of more than one dataset (possibly collected by different organizations and in different scenarios) allows testing NIDSs with several types of attacks (also unknown ones) and under various network configurations. In other words, such a *cross-evaluation* procedure enables the assessment of the model's ability to generalize on other datasets, thus obtaining more reliable insights for the deployment of DL-based NIDSs in the real world. With this aim in mind, in the present work, we employ different *state-of-art DL models* to perform *supervised AD* and adopt a *cross-evaluation* procedure to reliably assess the *performance* and *robustness* of such designed NIDSs.

To summarize, the key contributions of this work are as follows:

- we leverage and experimentally evaluate the performance of advanced DL architectures when performing

[1] https://bit.ly/clusit-report-2022

[2] https://bit.ly/cybersecurity-ventures-cybercrime-cost-by-2025

supervised AD: a *single-modal 1D Convolutional Neural Network (1D-CNN)*, a *single-modal Recurrent Neural Network (RNN)*, and the *multimodal* MIMETIC architecture [3] which capitalizes on multiple views of a given traffic object;
- we employ *two types of input data* extracted from network traffic and suitable for *"early"* (viz. timely) traffic classification and assess their effectiveness in distinguishing *malicious* from *benign* traffic;
- we assess the *detection capability* of DL-based NIDSs via a *cross-evaluation procedure* that aims to evaluate their *robustness* when tested in network contexts different than those in which they were originally designed and trained (i.e. different network configurations and attack types);
- we deepen the effect of informative traffic fields through an *obfuscation study* to quantify the impact of each field on AD performance when generalizing on malicious traffic collected in different networks;
- we exploit *three real, recent, and public-available network attack datasets* covering different applications, attack types, and network configurations for cross-evaluation purposes: `IoT-23` [4], `IDS2018` [5], and `KITSUNE` [6].

The rest of the manuscript is organized as follows. Section II discusses the related work and positions our contribution accordingly. Section III details the methodology defined to perform the cross-evaluation. Section IV presents the experimental setup. Section V shows the experimental results obtained. Finally, Section VI concludes the paper and discusses future avenues.

II. Related Work

In the present section, we discuss the most relevant works tackling AD through ML and DL approaches. Indeed, AD has found much interest due to the growing need for protecting networking systems from possible and ongoing attacks. Some proposals leverage *supervised ML approaches* to detect anomalies generated by Android malware in mobile network traffic [7, 8] or attacks against Internet of Things (IoT) devices [9]. Differently, other works [10, 11] employ *unsupervised ML techniques* (i.e. trained only on benign traffic) for addressing AD in different scenarios (e.g., detection of distributed denial of service attacks).

Recently, a surging number of papers are exploiting *DL approaches* to deal with AD. Proposals such as [12, 13, 14] feed complex deep neural networks with raw traffic data of new public datasets (e.g., IoT-23 [4], Kitsune [6], or Bot-IoT [15]) fostering the automatic extraction of knowledge via DL. On the contrary, works like [16, 17] train DL models in a counter-productive manner by utilizing manually-extracted features (e.g., "post-mortem" statistics extracted from the full sets of packet/payload lengths, inter-arrival times, etc.). Unfortunately, using handcrafted features hinders the "early" detection of anomalies and undermines the advantage of DL approaches in limiting domain-expert involvement. Furthermore, less recent works [18, 19] exploit datasets collected decades ago (e.g., KDD-Cup-99, NSL-KDD, or Kyoto2006+) which are scarcely representative of current traffic profiles.

Additionally, a corpus of works focuses on the slightly different *attack-traffic classification* task [19, 20, 21, 22], performing (supervised) multi-class traffic classification to infer specific attacks and distinguish them from benign traffic. Similarly, in [23, 24], the security-related problem of classifying the traffic generated by *anonymity tools* at different granularity (e.g., anonymous network, traffic type, and application) is taken into account.

Nevertheless, previous studies discussed above *only investigate the effectiveness of their proposals* by training and testing them on traffic data collected in the same network context, *completely leaving out aspects regarding the model robustness*, namely without analyzing if their proposals will keep working when the network context changes.

The generalization capability of ML-based attack-traffic classifiers is investigated in [25]. The authors release the ToN_IoT traffic dataset and use it to train a set of classifiers which is then tested on the IoT-23 dataset. The results show large performance discrepancies w.r.t. the good results attained on the sole ToN_IoT dataset, highlighting the need for standardization of feature descriptions and attack classes. Likewise, in [26], the authors benefit from a SHAP values-based explainability analysis [27] to highlight what features contribute the most for distinguishing between benign and malicious traffic flows when testing data are collected in network contexts different than the training one. Finally, in [28], the robustness of ML models is addressed by proposing *XeNIDS*, a framework based on Random Forests allowing different network contexts to be simultaneously considered. More specifically, *XeNIDS* is leveraged to support the deployment of NIDSs in an actual-network environment which possibly includes also unknown attacks.

Positioning of Our Contribution. Herein, we perform AD via supervised DL approaches assessed through a *cross-evaluation* procedure. This enables us to design NIDSs that are well-suited for real-world network environments. Differently than previous works [25, 26, 28] exploiting traditional ML approaches, we leverage both *single-modal and multimodal DL architectures* and feed them with different types of *unbiased* raw input data suitable for "early" AD (see Sec. III-A). Moreover, unlike [28], *we quantitatively evaluate the impact of different traffic fields on AD performance* and detect which fields negatively affect the generalization capability of considered models. Finally, we underline that we conduct our investigations by leveraging *recent public-available network-security datasets* [4, 5, 6] to make the obtained outcomes easier to reproduce, comparable against other studies, and generalizable to other datasets/approaches not studied yet.

III. Methodology

In this section, we detail the methodology defined to detect anomalous traffic via DL. Section III-A provides the problem definition, introduces the traffic object and related input data

that are fed to the DL models, and finally describes their architectures and the associated training procedure. Section III-B illustrates the key concepts of the cross-evaluation procedure adopted to assess the DL-based NIDSs.

A. Deep Learning-based Anomaly Detection

Herein, we carry out AD using supervised DL approaches: given a *traffic object* (i.e. an aggregation of traffic packets sharing common properties), we tackle a binary classification task that assigns a label between $\{benign, malicious\}$.

Traffic Object and Input Data. We segment network traffic into *bidirectional flows* (*biflows*), defined as a stream of packets sharing the same 5-tuple (i.e. transport-level protocol, source and destination IP addresses and ports) regardless of the direction of communication. For each biflow, we extract two sets of unbiased input data, namely we remove the bytes/fields (e.g., PCAP metadata, absolute timestamps, local IP addresses, or source/destination ports) that could inflate AD performance and thus lead to misleading results. More in detail, (*i*) `NET` input consists of the first N_b bytes of the network-layer packet (i.e. header and payload) arranged in a byte-wise format after the obfuscation of biased fields (i.e. IP addresses, ports, and checksums) [21]; (*ii*) `PSQ` input includes a set of unbiased informative fields of the first N_p packets: (a) the number of bytes in the network-layer packet (PL), (b) the direction $\in \{-1, 1\}$ (DIR), (c) the TCP window size (WIN) equal to zero for UDP biflows, and (d) the inter-arrival time w.r.t. the previous packet (IAT). We emphasize that both inputs are naturally suited for "early" AD.

DL Architectures and Training Procedure. We employ state-of-art DL approaches well-suited for AD and attack-traffic classification [21]. Specifically, we consider two single-modal architectures (1D-CNN and RNN) and a multimodal one (MIMETIC [3]). The 1D-CNN is fed with the `NET` input and is made up of two 1D convolutional layers (with 16 and 32 filters, respectively) each followed by a max-pooling layer (with unit stride and spatial extent equal to 3), a flatten layer, a fully-connected layer (with 256 neurons), and the final softmax that fulfills AD. The RNN is fed with the `PSQ` input and consists of a bidirectional Gated Recurrent Unit (with 64 units) followed by a flatten and a fully-connected layer (with 256 neurons) before the final softmax.

The multimodal MIMETIC combines the two single-modal networks described above, each fed with the relevant input, to capitalize on multiple views of the same traffic object. In more detail, the outputs of the last fully-connected layers of the two single-modal branches (i.e. before the final softmax) are concatenated via a merge layer and fed to a fully-connected (shared-representation) layer. The MIMETIC architecture is completed with a final softmax layer performing AD. MIMETIC is trained via a two-phase procedure encompassing: (*i*) *pre-training*, where each single-modal branch is independently trained and (*ii*) *fine-tuning*, which involves the fully-connected layers of both single-modal branches and the shared-representation layers.

Both the single-modal architectures are trained for a maximum of 25 epochs. The whole training phase of MIMETIC encompasses 90 epochs at most: the pre-training of single-modal branches is performed for 25 epochs in line with the training of single-modal architectures, the fine-tuning for 40 epochs. For all DL architectures, we minimize a binary cross-entropy via the Adam optimizer set with a batch size of 256 and take advantage of the early stopping technique (with patience of 15 epochs and minimum delta of 0.01) to prevent overfitting. To further promote regularization, we apply a 20% dropout after (a) each fully-connected layer (including the merge layer) and (b) after flattening the 2D representation of both the stack of convolutional/pooling layers and the Gated Recurrent Unit.

B. Cross-evaluation of DL-based NIDSs

The *cross-evaluation* of DL-based NIDSs allows us to assess these systems in network operational scenarios different from the training one. This evaluation can be performed according to different contexts. More formally, with the term *context* we refer to the composition of the data used for training and testing DL models leveraged to realize the NIDSs in terms of *benign* or *malicious* samples (in the case of binary AD). In particular, depending on the context, models are trained and/or tested in the same network scenario (i.e. same network configuration and attack types) or in different scenarios (realized by combining different datasets).

In the present work, we consider 3 contexts where AD can be performed, reflecting real-world use cases [28].

Baseline. This context is the most investigated in the state-of-the-art literature. In this case, both the training and test sets include *benign* and *malicious* biflows belonging to the same dataset. The *Baseline* context refers to a use case in which an organization trains and evaluates its NIDS with benign and malicious traffic collected into its network infrastructure.

Generalization. This context aims to assess if the NIDS can detect attacks not included in the training set. In more detail, considering two datasets D_A and D_B, the training set contains *benign* and *malicious* biflows belonging to D_A, while the test set is composed of *benign* biflows of D_A and *malicious* biflows of D_B. The *Generalization* context represents the use case in which an organization trains its NIDS with traffic collected into its network infrastructure but wants to assess the NIDS' ability to detect attack traffic originating from a different network. Given the diverse and increasing number of threats in today's networks, the cross-evaluation performed according to the *Generalization* context is crucial for organizations to assess NIDSs' robustness in different network conditions and evaluate their ability to handle different or unseen attacks.

Extension. This context comes into play when there is a need to balance low detection performance against unknown attacks (i.e. the NIDS shows low generalizability). To this aim, attack traffic originating from a different network is included in the training set to potentially extend the detection capabilities of the NIDS. Precisely, taking two datasets D_A and D_B, the training set contains *benign* biflows of D_A and

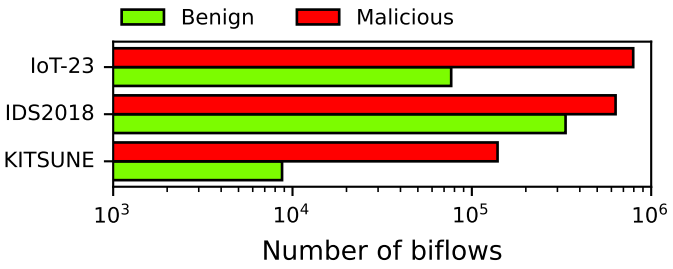

Fig. 1: Number (in log scale) of *benign* and *malicious* biflows for each considered dataset.

malicious biflows of both the datasets D_A and D_B; the test set encompasses *benign* biflows of D_A and *malicious* biflows of D_B. The difference with the *Generalization* context is that the *malicious* biflows of D_B are added to the training set to help the detection process. The *Extension* context represents the use case where an organization—having noticed poor generalization performance—adds malicious traffic collected into a different network to the malicious traffic of its network infrastructure. The goal is to understand whether by exploiting such augmented knowledge during training, its NIDS becomes able to improve AD performance w.r.t. the *Generalization* context.

IV. Experimental Setup

This section outlines our experimental setup. Section IV-A describes the datasets employed for cross-evaluation. Section IV-B summarizes the metrics considered to evaluate performance. Finally, Section IV-C details the preliminary steps (viz. the pre-processing operations) to the cross-evaluation.

A. Datasets Description

Figure 1 depicts the composition of the three datasets to perform our cross-evaluation in terms of the number of *benign* and *malicious* biflows. Given the high heterogeneity of datasets, we pre-process them to have a comparable number of biflows, to keep the intra-dataset balance between benign and malicious biflows among all the datasets, and to reduce the computational burden.[3] To carry out AD, for each dataset, we label all the biflows attributed to the various attack classes as *malicious*. We refer to the datasets' papers/websites for details on the collection environment, equipment, and process.

`IoT-23` [4]. The `IoT-23` dataset encompasses 23 PCAP traffic traces captured within a controlled IoT environment with unrestrained network connections, with no defense solutions being enforced. A Raspberry Pi infected with a certain malware is exploited to generate malicious traffic, while three real IoT devices (i.e. a Philips HUE Smart Led Lamp, an Amazon Echo Home, and a Somfy Smart Doorlock) generate benign one. Overall, 20 traces contain malicious traffic, and 3 benign traffic. Detailing, `IoT-23` comprises the following attacks: PortScan, Okiru, DDoS, HeartBeat, Torii, Command&Control.[4] Since the dataset exhibits a severe class imbalance problem (i.e. highly-populated classes have more than $15M$ biflows while others have less than $40k$ biflows), we randomly down-sampled (without replacement) the majority classes to 0.25%. Consequently, the `IoT-23` dataset contains $870.6k$ biflows.

`IDS2018` [5]. The `CSE-CIC-IDS2018 on AWS` (thereafter `IDS2018` for brevity) dataset contains benign traffic along with the following attack classes: Brute-force, DoS, Infiltration, Botnet, DDoS, and PortScan. The attack infrastructure consists of 50 machines, while the victim organization has 5 departments and includes 420 machines and 30 servers. The dataset includes the network traffic of each machine captured over 10 days. Similarly to `IoT-23`, we down-sample the malicious traffic to 20% and the benign traffic to 1% for each of the 10 capture days. As a result, the `IDS2018` dataset contains $966.2k$ biflows.

`KITSUNE` [6]. The `KITSUNE` dataset contains benign traffic and network-attack traffic collected in a commercial IP-based surveillance system by setting up an IoT network consisting of two deployments of four monitoring cameras each. The attacks are conducted via different tools (e.g., Nmap, Hping3, Ettercap) and are targeted to affect the availability and integrity of video uplinks. `KITSUNE` includes Reconnaissance, Man-in-the-middle, DoS, Injection, Flooding, and Botnet attacks. Overall, the dataset contains $147.9k$ biflows. We highlight that for privacy reasons, each packet payload is trimmed to 200 bytes. We take into account this constraint in the pre-processing steps needed for cross-evaluation (see Sec. IV-C for details).

B. Evaluation Metrics

Our evaluation leverages a *stratified hold-out technique*: all datasets are split into training (80%) and test (20%) sets *by keeping the proportion of samples* of the *benign* and *malicious* biflows. To carry out cross-evaluation, such training and test sets are properly combined according to the investigated context (see Sec. III-B). Since generalization and extension contexts involve mixing data from different organizations, to avoid the inference of biased insights due to peculiar or unfortunate combinations of data in training and/or test sets, we repeat each experiment 10 times by varying the stratified hold-out pseudo-random seed in each repetition. Accordingly, for each metric described hereinafter, we report the average and standard deviation over the 10 repetitions.

To cope with the problem of class imbalance between benign and malicious biflows, we benefit from the following metrics: (a) $Precision = \frac{T_P}{T_P+F_P}$ calculates the ratio of positive class predictions that are actually positive; (b) $Recall = \frac{T_P}{T_P+F_N}$ calculates the ratio of positive class predictions made out of all positive samples in the test set; (c) $F1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision+Recall}$ is the harmonic mean of precision and recall that aims to take into account their

[3]More specifically, the original `IoT-23` and `CSE-CIC-IDS2018 on AWS` datasets required downsampling having a number of biflows several orders of magnitude higher than `KITSUNE`.

[4]For more details on malicious traffic collected, we refer to [4].

TABLE I: F1-score [%] of 1D-CNN, RNN and MIMETIC in the *Baseline* context. Results are shown as $avg \pm std$ over 10 repetitions. For each dataset, the F1-score of the best-performing model is highlighted in boldface.

Dataset	1D-CNN	RNN	MIMETIC
IoT-23	78.10 (±2.53)	71.57 (±9.83)	**91.19 (±2.11)**
IDS2018	99.27 (±0.63)	99.25 (±0.87)	**99.96 (±0.01)**
KITSUNE	91.59 (±1.79)	82.02 (±4.33)	**91.98 (±1.74)**

trade-off. Specifically, we exploit the macro (viz. arithmetically-averaged) F1-score of benign and malicious classes. In the above formulas, T_P denotes the true positives, F_P the false positives, and T_N the true negatives.

C. Pre-processing Steps

Before carrying out the cross-evaluation of DL approaches exploited for AD, a number of pre-preprocessing steps are required. Firstly, for each biflow, we extract the first $N_b = 200$ bytes and $N_p = 12$ packets to construct the NET and PSQ input, respectively (see Sec. III-A for details).

For the NET input, the choice for $N_b = 200$ bytes is compliant with the size of the available packet bytes of KITSUNE. Indeed, this choice has allowed us to use a uniform input size and format for all three datasets. Also, we obfuscate some bytes of the network-layer packet of the NET input (i.e. those corresponding to IP addresses, ports, and checksums are replaced with zeros). Our aim is to remove information that could cause overfitting (and consequently biased results) as it is representative of a particular network environment and not of the traffic nature [21]. On the contrary, selecting $N_p = 12$ packets per biflow for the PSQ input is based on extensive analyses performed in our previous works [3] and further preliminary validations not reported for brevity.

For both NET and PSQ inputs, if a sample is longer or shorter than the prescribed N_b or N_p length, truncation or zero-padding is applied, respectively. Also, to distinguish actual from padded zeros, we add 1 to each byte of the NET input and to the WIN field of the PSQ input.

V. EXPERIMENTAL EVALUATION

The present section shows the results of the experimental evaluation conducted using our cross-evaluation procedure. Section V-A reports the performance of 1D-CNN, RNN, and MIMETIC in the *Baseline* context. Then, Section V-B and Section V-C discuss their detection capabilities in the *Generalization* and *Extension* contexts, respectively.

A. Baseline Context Performance

Table I reports the experimental results attained in the *Baseline* context in terms of F1-score. MIMETIC *is the best-performing DL approach* on all considered datasets, while the 1D-CNN achieves similar (but slightly lower) results on KITSUNE and IDS2018. Overall, the highest mean F1-score value of 99.96% is reached by MIMETIC on the IDS2018 dataset. Similarly, the 1D-CNN performs well with the same dataset (99.27% F1-score), while it reaches a lower mean F1-score of 78.10% when assessed on IoT-23. On the other hand, even though the RNN presents a mean F1-score of 99.25% with IDS2018, it shows the worst performance on all the datasets, down to 71.57% F1-score with the IoT-23 dataset.

Regarding the AD performance obtained on the diverse datasets, we can notice that on IoT-23 the multimodal MIMETIC—capitalizing on multiple views of the same biflow—experiences significantly higher F1-score values than both single-modal approaches: +13% over 1D-CNN and +20% over RNN. Conversely, on IDS2018 all DL approaches reach F1-score values higher than 99%, while on KITSUNE MIMETIC outperforms RNN of +10%.

Finally, we can observe that performance variability is limited with standard deviation values generally less than 3%. The sole RNN experiences a slightly higher variability, particularly on the IoT-23 dataset having a standard deviation of 9.83%.

B. Generalization Context Performance

Figure 2 shows the performance in terms of F1-score achieved in the *Generalization* context. For each cell, the top x-axis reports the composition of *benign* and *malicious* biflows of the training set, while the bottom x-axis shows the composition of the test set. We recall that the *Generalization* context is the most challenging one since the NIDS is tested for detecting *malicious* traffic not seen during the training.

Indeed, as expected, the models do not achieve generally satisfactory performance. Nevertheless, we can draw interesting remarks by focusing on specific cases. MIMETIC *attains the highest mean F1-score* of 76.36% when trained with IoT-23 biflows and generalizing to KITSUNE malicious biflows during testing. Similarly, the 1D-CNN shows good generalization capability (72.17% F1-score) with the same training set but when tested on IDS2018 malicious biflows. Conversely, the worst results are obtained when training on IDS2018 and testing on IoT-23 attack traffic, with all DL models exhibiting F1-score values lower than 24%.

Focusing on the specific DL networks, the RNN exhibits the lowest performance, also when trained on IoT-23. It is also characterized by high variability of performance, with F1-score standard deviation up to 32.65% when trained on KITSUNE and tested on IDS2018 malicious biflows. Conversely, the 1D-CNN and MIMETIC have almost always better performance than the RNN, even though always lower than 44% F1-score when trained on IDS2018 or KITSUNE. Notably, in the latter case, the RNN outperforms both the other models regardless of the specific test set considered.

Finally, we point out that some experiments reveal large standard deviations, particularly, as aforementioned, those related to the RNN. In other words, the performance depends on the similarity/difference of the malicious biflows constituting the training and test sets. Specifically, the more similar the latter biflows (collected in a different network context and not seen during training) are to the malicious biflows of the training set, the more they should be distinguishable from the

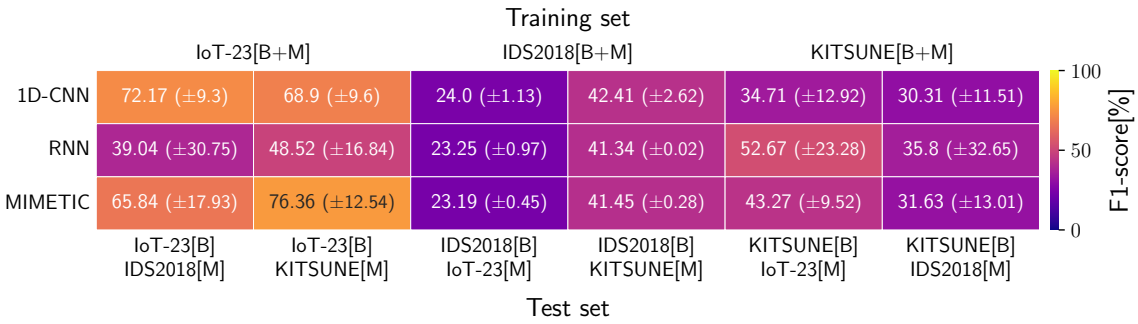

Fig. 2: F1-score [%] of 1D-CNN, RNN, and MIMETIC in the *Generalization* context. Results are shown as $avg \pm std$ over 10 repetitions. The top and bottom x-axes specify the compositions of training and test sets in terms of *benign* [B] and *malicious* [M] biflows related to the three considered datasets, respectively.

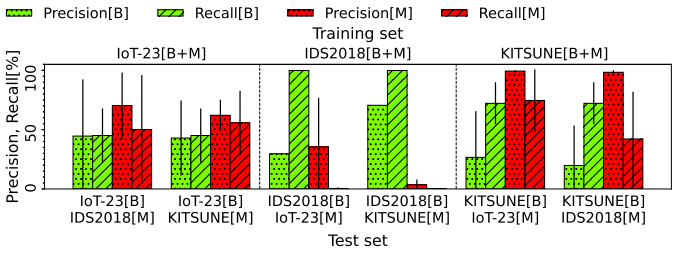

Fig. 3: Precision and Recall [%] of RNN for *benign* and *malicious* classes in the *Generalization* context. Results are shown as $avg. \pm std.$ over 10 repetitions. The top and bottom x-axes specify the compositions of training and test sets in terms of benign [B] and malicious [M] biflows related to the three considered datasets, respectively.

benign biflows of the test set. Therefore, such performance variations are likely due to the different composition of training and test sets across the considered repetitions (based on a pseudo-random hold-out split).

Fine-Grained Analysis: Precision and Recall. Herein, we aim to provide further insights into the (poor) performance of the RNN in the *Generalization* context via a fine-grained analysis showing the *precision* and *recall* of both the *benign* and *malicious* classes, separately. Figure 3 depicts the results of this analysis for all the combinations of training and test sets in the *Generalization* context. Overall, the correct detection of *benign* and malicious biflows is highly related to such combinations.

Detailing, when training on `IoT-23`, poor performance is obtained for both *benign* biflows ($\approx 40\%$ precision/recall regardless of the specific test set) and *malicious* biflows for which precision equals 70% (resp. 65%) and recall is 50% (resp. 55%) when generalizing to the `IDS2018` (resp. `KITSUNE`) dataset. A notable behavior is observed when training on `IDS2018`: high recall values ($> 95\%$) for *benign* biflows (i.e. low false negatives) are associated with very poor performance in the detection of malicious ones. Such an outcome points out that the RNN tends to predict malicious biflows of `IoT-23` and `KITSUNE` datasets as benign, namely it fails to generalize to malicious samples not seen during the training. Lastly, when training on `KITSUNE`, the precision of *benign* biflows detection drops down to 30% (i.e. high false positives). This result indicates that the RNN misclassifies *benign* biflows as *malicious*, although `KITSUNE` biflows have been seen during the model training, highlighting the toughness of AD in the *Generalization* context.

As discussed before, the RNN exhibits large variations for most of the training-test set combinations. Since the high deviations for the *benign* class are mainly related to low precision and recall values, we speculate that they are due to repetitions where greater confusion between *benign* and *malicious* biflows occurred. Similar considerations can be drawn for the *malicious* class. However, in this case, the variability can be associated with the differences in *malicious* samples contained in training and test sets at each repetition.

Obfuscation Study. In light of the above considerations, we explore how the RNN performance varies when obfuscating (i.e. set to 0 during both training and test) the fields of the `PSQ` input one by one for all the N_p packets. Our aim is to quantify the impact of each field and pinpoint whether the poor performance of RNN can be attributable to a specific one.

Figure 4 shows the difference w.r.t. the F1-score obtained by the RNN fed with the whole `PSQ` input (i.e. no obfuscation is enforced). A specific field that significantly affects AD performance is not clearly identifiable for all combinations of datasets considered in the *Generalization* context: the F1-score variation depends on the specific combination. Nevertheless, the lowest and highest differences are both obtained when training on `IoT-23` and generalizing to `IDS2018` and range from a mean drop of -23.16% (obfuscating `WIN`) to a mean boost of $+30.12\%$ (obfuscating `DIR`).

Going into the details of specific fields, the obfuscation of `IAT` generally leads to performance improvements or to negligible variations, on average. Particularly, the highest improvement is attained when training on `KITSUNE`, with up to $+29.55\%$ F1-score when generalizing to `IoT-23`. We recall that in the latter context, the RNN exhibits very low precision on the *benign* biflows (see Fig. 3); then, we speculate that

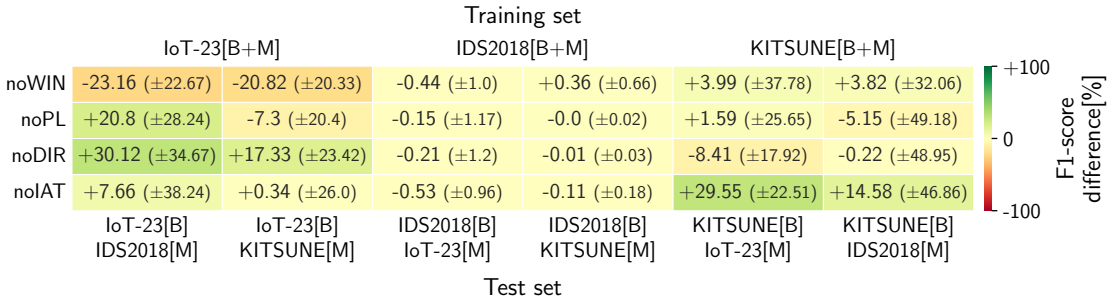

Fig. 4: Obfuscation study in the *Generalization* context. Results show the difference of the F1-score [%] of the RNN fed with the obfuscated and the whole PSQ input set, and they are shown as $avg. \pm std.$ over 10 repetitions. The top and bottom x-axes specify the compositions of training and test sets in terms of benign [B] and malicious [M] biflows related to the three considered datasets, respectively.

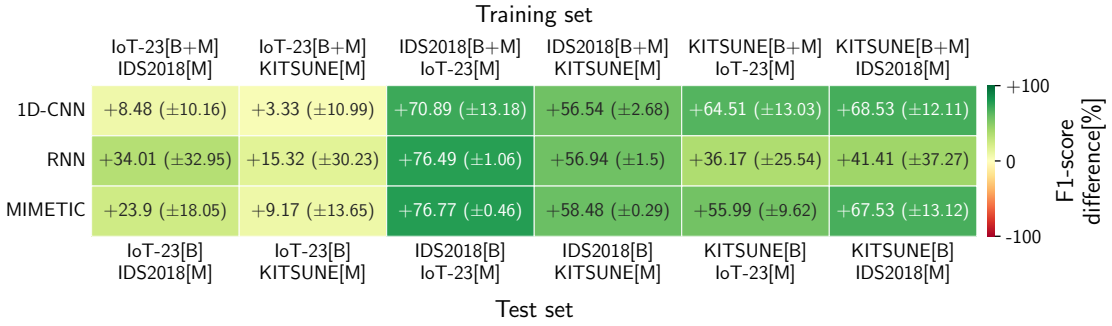

Fig. 5: Performance of 1D-CNN, RNN, and MIMETIC in the *Extension* context. Results show the difference of the F1-score values [%] attained in the *Extension* context with those of the same model attained in the *Generalization* context, and they are shown as $avg. \pm std.$ over 10 repetitions. The top and bottom x-axes specify the compositions of training and test sets in terms of benign [B] and malicious [M] biflows related to the three considered datasets, respectively.

obfuscating IAT could mitigate such performance drop when generalizing to IoT-23 or IDS2018. Similarly, obfuscating DIR yields an F1-score improvement of $+30.12\%$ (resp. $+17.33\%$) when training on IoT-23 and generalizing to IDS2018 (resp. KITSUNE), while it proves to be a valuable field when training on KITSUNE and generalizing to IoT-23 with an F1-score drop of -8.41% observed without DIR.

We emphasize that the dataset composition in *Generalization* contexts, in terms of *benign* and *malicious* samples within the different repetitions, is an important factor to be taken into account, in line with what was discussed in Sec. V-B: large variations from mean F1-score values are evident in some cases. For instance, obfuscating WIN when training on KITSUNE and generalizing to IoT-23 gives an F1-score improvement of $+3.99\%$ but with a standard deviation of 37.78%, demonstrating that in some repetitions, performance decay occurred despite an overall improvement.

C. Extension Context Performance

In this section, we discuss the performance obtained for the last cross-evaluation analysis related to the *Extension* context. To this end, Fig. 5 reports the difference between the F1-score values attained in the *Extension* context with those of the *Generalization* context, given that the AD capability of DL models is evaluated in the same scenario (i.e. same test set). On the other hand, we recall that in the *Extension* context, the training set exploited in the *Generalization* context is extended with malicious traffic collected from the tested network.

Overall, we can notice that such augmented knowledge *significantly enhances DL model detection capability*, on average. Especially, extending IDS2018 with IoT-23 (resp. KITSUNE) *malicious* biflows leads to an improvement of more than 70% (resp. 57%) for all the models. Accordingly, MIMETIC *ramps up to* $> 99\%$ *F1-score*, and this consistent performance boost is also corroborated by F1-score standard deviations less than 1%. Finally, we highlight that, although the average F1-score values are always higher than those observed in the *Generalization* context, high standard deviations reveal that for some repetitions lower F1-score values are obtained. Nevertheless, this commonly occurs for training-test combinations showing quite good performance already in the *Generalization* context (e.g., training on IoT-23 extended with IDS2018/KITSUNE *malicious* traffic).

VI. Conclusions and Future Perspectives

In this work, we performed supervised AD via state-of-the-art DL approaches based on single-modal and multimodal architectures. Specifically, we conducted a *cross-evaluation procedure* to assess their performance and examine the robustness of these DL-based NIDSs in operational scenarios

different from the training one. To this end, we exploited three real, recent, and public-available datasets encompassing *benign* and *malicious* traffic data collected in different network scenarios: `IoT-23`, `IDS2018`, and `KITSUNE`.

The experimental results showed that all DL models attain satisfactory performance (up to $> 99\%$ F1-score with the multimodal approach) when operated in the same network where they were trained. Conversely, a significant performance drop was experienced when AD is performed in different network conditions, i.e. when detecting attacks unseen during training. In this case, DL models fed with informative fields extracted from packet sequences exhibited significantly lower performance (down to 24% F1-score) than the models fed with (unbiased) raw traffic data (up to 76% F1-score). Accordingly, we conducted an obfuscation study to identify packet fields that could cause such performance drops. We found that some fields could be detrimental for AD and obfuscating them led to better performance (e.g., the obfuscation of the inter-arrival times generally allows a performance improvement up to $+29\%$ F1-score). Nevertheless, the benefit of obfuscation is highly dependent on the specific network scenario. Therefore, we considered a further use case where *malicious* traffic collected from the operational network is added to the original traffic during training. We obtained that such augmented knowledge improves DL model detection capability (up to $> 99\%$ F1-score reached with the multimodal approach).

In future work, we plan to perform cross-evaluation with finer granularity to conduct multi-class attack-traffic classification and investigate eXplainable Artificial Intelligence techniques to interpret the decisions of DL-based NIDSs in the challenging scenarios of cross-evaluation and guide their proper design.

ACKNOWLEDGMENTS

The authors would like to thank Nicola D'Ambra, Alberto Urraro, and Marco Vaiano for their collaboration with the preliminary experiments of this study. This work is partially supported by the Italian Research Program "PON *Ricerca e Innovazione* 2014–2020 (PON R&I) – Asse IV *REACT-EU* – Azione IV.4", and the *Ce.S.M.A. institute* of the University of Napoli "Federico II" via the research grant "*BSRicerca/CESMA/2022/CMOBILITY_01* – Soluzioni avanzate per l'analisi del traffico di rete".

REFERENCES

[1] N. Chaabouni, et al. Network intrusion detection for IoT security based on learning techniques. *IEEE Communications Surveys & Tutorials*, 21 (3):2671–2701, 2019.

[2] A. Khraisat et al. A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. *Cybersecurity*, 4:1–27, 2021.

[3] G. Aceto, et al. MIMETIC: Mobile encrypted traffic classification using multimodal deep learning. *Computer networks*, 165:106944, 2019.

[4] A. Parmisano, et al. Stratosphere Laboratory. A labeled dataset with malicious and benign IoT network traffic., January 2022. URL https://www.stratosphereips.org/datasets-iot23.

[5] I. Sharafaldin, et al. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp*, 1:108–116, 2018.

[6] Y. Mirsky, et al. Kitsune: an ensemble of autoencoders for online network intrusion detection. *arXiv preprint arXiv:1802.09089*, 2018.

[7] A. H. Lashkari, et al. Towards a network-based framework for android malware detection and characterization. In *PST 2017*. IEEE, 2017.

[8] G. Bovenzi, et al. A Comparison of Machine and Deep Learning Models for Detection and Classification of Android Malware Traffic. In *2022 ISCC*, pages 1–6, 2022. doi: 10.1109/ISCC55528.2022.9912986.

[9] A. Kumar, et al. Machine learning-based early detection of IoT botnets using network-edge traffic. *Computers & Security*, 117:102693, 2022.

[10] A. Karami. An anomaly-based intrusion detection system in presence of benign outliers with visualization capabilities. *Expert Systems with Applications*, 108:36–60, 2018.

[11] K. Yang, et al. Ddos attacks detection with autoencoder. In *NOMS 2020-2020 IEEE/IFIP network operations and management symposium*, pages 1–9. IEEE, 2020.

[12] A. Yehezkel, et al. Network anomaly detection using transfer learning based on auto-encoders loss normalization. In *ACM AISec 2021*, 2021.

[13] I. Ullah et al. A Framework for Anomaly Detection in IoT Networks Using Conditional Generative Adversarial Networks. *IEEE Access*, 9: 165907–165931, 2021.

[14] I. Guarino, et al. On the use of Machine Learning Approaches for the Early Classification in Network Intrusion Detection. In *2022 IEEE International Symposium on Measurements & Networking (M&N)*, pages 1–6, 2022. doi: 10.1109/MN55117.2022.9887775.

[15] N. Koroniotis, et al. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *Future Generation Computer Systems*, 100:779–796, 2019.

[16] M. Woźniak, et al. Recurrent neural network model for IoT and networking malware threat detection. *IEEE Transactions on Industrial Informatics*, 17(8):5583–5594, 2020.

[17] R. Kozik, et al. A new method of hybrid time window embedding with transformer-based traffic data classification in IoT-networked environment. *Pattern Analysis and Applications*, 24(4):1441–1449, 2021.

[18] K. Wu, et al. A novel intrusion detection model for a massive network using convolutional neural networks. *IEEE Access*, 6:50850–50859, 2018.

[19] N. Shone, et al. A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(1):41–50, 2018.

[20] M. A. Ferrag, et al. Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *Journal of Information Security and Applications*, 50:102419, 2020.

[21] A. Nascita, et al. Machine and deep learning approaches for iot attack classification. In *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops, INFOCOM 2022 - Workshops, New York, NY, USA, May 2-5, 2022*, pages 1–6. IEEE, 2022. doi: 10.1109/INFOCOMWKSHPS54753.2022.9797971. URL https://doi.org/10.1109/INFOCOMWKSHPS54753.2022.9797971.

[22] G. Bovenzi, et al. A hierarchical hybrid intrusion detection approach in IoT scenarios. In *GLOBECOM 2020*. IEEE, 2020.

[23] A. Montieri, et al. A Dive into the Dark Web: Hierarchical Traffic Classification of Anonymity Tools. *IEEE Transactions on Network Science and Engineering*, 7(3):1043–1054, 2020. doi: 10.1109/TNSE.2019.2901994.

[24] L. Wang, et al. Multilevel identification and classification analysis of Tor on mobile and PC platforms. *IEEE Transactions on Industrial Informatics*, 17(2):1079–1088, 2020.

[25] T. M. Booij, et al. ToN_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets. *IEEE Internet of Things Journal*, 9(1):485–496, 2021.

[26] S. Layeghy et al. On Generalisability of Machine Learning-based Network Intrusion Detection Systems. *arXiv preprint arXiv:2205.04112*, 2022.

[27] S. M. Lundberg et al. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[28] G. Apruzzese, et al. The cross-evaluation of machine learning-based network intrusion detection systems. *IEEE Transactions on Network and Service Management*, 19(4):5152–5169, 2022. doi: 10.1109/TNSM.2022.3157344.