A Hierarchical Hybrid Intrusion Detection Approach in IoT Scenarios

Giampaolo Bovenzi, Giuseppe Aceto, Domenico Ciuonzo, Valerio Persico, Antonio Pescapé
University of Napoli Federico II (Italy), name.surname@unina.it

GLOBECOM 2020 - 2020 IEEE Global Communications Conference | 978-1-7281-8298-8/20/$31.00 ©2020 IEEE | DOI: 10.1109/GLOBECOM42002.2020.9348167

***Abstract*—Internet of Things (IoT) fosters unprecedented network heterogeneity and dynamicity, thus increasing the variety and the amount of related vulnerabilities. Hence, traditional security approaches fall short, also in terms of resulting scalability and privacy. In this paper we propose H2ID, a two-stage hierarchical Network Intrusion Detection approach. H2ID performs (i) *anomaly detection* via a novel lightweight solution based on a MultiModal Deep AutoEncoder (M2-DAE), and (ii) *attack classification*, using soft-output classifiers. We validate our proposal using the recently-released Bot-IoT dataset, inferring among four relevant categories of attack (DDoS, DoS, Scan, and Theft) and unknown attacks. Results show gains of the proposed M2-DAE in the case of simple anomaly detection (up to -40% false-positive rate when compared with several baselines at same true positive rate) and for H2ID as a whole when compared to the best-performing misuse detector approach (up to $\approx +5\%$ F1 score). Besides the performance advantages, our system is suitable for distributed and privacy-preserving deployments while limiting re-training necessities, in line with the high efficiency as well as the flexibility required in IoT scenarios.**

INTRODUCTION

The number of Internet of Things (IoT) devices has been estimated at ≈ 7 B in 2018, with a $3\times$ expected growth by 2025, considering both consumer and industrial applications.[1] Unluckily, IoT devices are often characterized by a low-cost production process (including hardware and software design choices, e.g., insecure network services, unsafe update mechanism, outdated components) and poor attention to configuration by average consumers. This results in huge security concerns and exposure to several vulnerabilities.[2] Hence, the manifold and significant weaknesses of IoT devices have shifted the main target of daily-released malware, now pointing to infect IoT services.

Once compromised, IoT devices may be *maliciously* leveraged in the form of a *botnet*, which capitalizes their "massive" and "always-on" nature according to newly-exploitable vulnerabilities. Still, IoT botnets differ from classical ones because of the huge number of involved bots and their heterogeneity. These cyberweapons are often used to perform Distributed Denial of Service (DDoS) attacks, but could be leveraged also for social engineering purposes, by intercepting critical information from targets and even to spread malware to perform unauthorized mining with the diffusion of cryptocurrencies. Indeed, a number of IoT botnets have been recently documented, such as the well-known Mirai malware [1], its numerous variants [2], proof-of-concept worms designed to leverage Zigbee protocol to infect high-density IoT devices, i.e. smartbulbs [3], or other noteworthy IoT malware instances such as Bashlite, Hajime, BrickerBot, NewAidra, and VPNFilter. It is worth noticing that malicious activities (e.g. DDoS, phishing) are always represented by a reconnaissance phase (scan) and a data exfiltration phase (theft), *that are likely to leave a trace* in the network traffic [1, 4–6]. Accordingly, Network Intrusion Detection Systems (NIDSs) are meant to monitor network traffic to determine when a system is being targeted by a network attack, or is a source of it. Recently, *Machine Learning* and *Deep Learning* have gained the attention of the networking research community, with the progresses in these areas reflecting their benefits also in network management, analysis, and security [7–9]. Indeed, they have been also applied with good results to NIDSs design [10, 11]. Since IoT is highly dynamic from multiple viewpoints (e.g., the number and variety of devices, their spatial distribution, and the evolution of attacks) an IoT-tailored NIDS must cope with these challenges, and its design is expected to address yet unknown attacks, while retaining high efficiency to be deployed also onto on a massive number of resource-constrained devices. We remark that privacy issues are also present, with IoT devices likely deployed in domestic and other highly-sensitive contexts.

To fulfill their goals, NIDSs may implement two main approaches: *Anomaly Detection* (AD) or *Misuse Detection* (MD), aiming at capturing any deviation from the profiles of normal activities, or identifying patterns of known attacks, respectively. Indeed, Machine Learning-based intrusion detection has been widely adopted in last years, with researches investigating both AD [10, 12, 13], trained only on benign traffic (i.e. anomalies are identified as outliers), and MD [11, 14, 15], trained on both benign and malicious traffic [16]. MD could be binary (benign vs. malicious) or multi-class (benign vs. specific attacks). Further, several approaches fall within Attack Classification (AC), where a preliminary phase, skimming benign events, is assumed. Also, a number of proposals for network intrusion detection in IoT environments can be found in literature [17–21]. Differently from our study, IoT-aware IDSs in literature do not use an IoT dataset for validation, or target AD or MD separately. Also, the most related proposal [19] discriminates only among known attacks (i.e. no ability to detect unknown attacks). In this work, we propose and evaluate a hybrid approach targeting both at the

[1] https://tinyurl.com/iot-dev-2018
[2] https://www.owasp.org/index.php/OWASP_Internet_of_Things_Project

978-1-7281-8298-8/20/$31.00 ©2020 IEEE

same time: **H2ID**—Hierarchical Hybrid Intrusion Detection—is a hierarchical intrusion detection approach tailored for the demands of IoT scenarios.

The contribution of this work is four-fold: *(i)* Our proposal is able to detect *both known and unknown attacks*, leveraging network traffic characteristics for identifying and discriminating among classes of attacks performed against IoT networks, performing a hybrid task of *anomaly detection* and open-set *attack classification*: to the best of our knowledge, in literature no other system targets this hybrid task in the context of IoT. *(ii)* We devise the system in a hierarchical fashion to benefit from its intrinsic operational and (re-)training efficiency [9], leveraging Machine Learning and Deep Learning for the design of both stages. In detail, our proposal is based on a two-stage architecture (enabled by a double-censoring mechanism) that consists of *a lightweight first stage* performing anomaly detection, and a *second stage* addressing open-set attack classification *only* on anomalies detected by the first stage. Such architecture naturally supports performance-enhancing distributed deployments while remaining *privacy-preserving*, thus avoiding the need for purposely-designed privacy-aware learning approaches. *(iii)* We design a novel solution based on Deep Auto Encoders, exploiting *multimodality*, able to efficiently manage both numerical and categorical features, to obtain a lightweight anomaly-detection stage deployable on resource-constrained devices. *(iv)* We evaluate our proposal on a recent dataset (i.e. BotIoT [22]) suitable for IoT traffic analysis, thus validating the effectiveness of our proposal on a realistic IoT botnet scenario.

The paper is organized as follows. Sec. II presents the proposed H2ID, while Sec. III discusses the benefits deriving from its distributed deployment in real scenarios, with a focus on privacy issues; experimental results are reported in Sec. IV; finally, Sec. V provides conclusions and future perspectives.

Proposed Architecture

In this section we describe our H2ID, depicted in Fig. 1: we first introduce the overall architecture and then detail the *two main stages* composing it, via the respective design choices. Before going into the details, we recall that a preliminary phase is required to perform the *traffic segmentation* of the raw traces (RT) into Traffic Objects (TOs) and to extract the relevant *input data* from them. Although several segmentation criteria exist, those leading to *flows* and *biflows* are the most common. In detail, a flow is a set of packets with the same 5-tuple (i.e. source IP, source port, destination IP, destination port, and transport-level protocol). Instead, biflows (i.e. bidirectional flows) include both directions of traffic. Specifically, the first stage of H2ID corresponds to an **Anomaly Detector** (**S1**, cf. Sec. II-A), whose output may activate an **Attack Classifier** implemented as the second stage (**S2**, cf. Sec. II-B). Indeed, if a TO is flagged as anomalous (ANM) by S1, deviations from normal activities are identified and are further inspected by S2 in search for known attacks. S2 is in charge of classifying the TO according to a set of known attacks

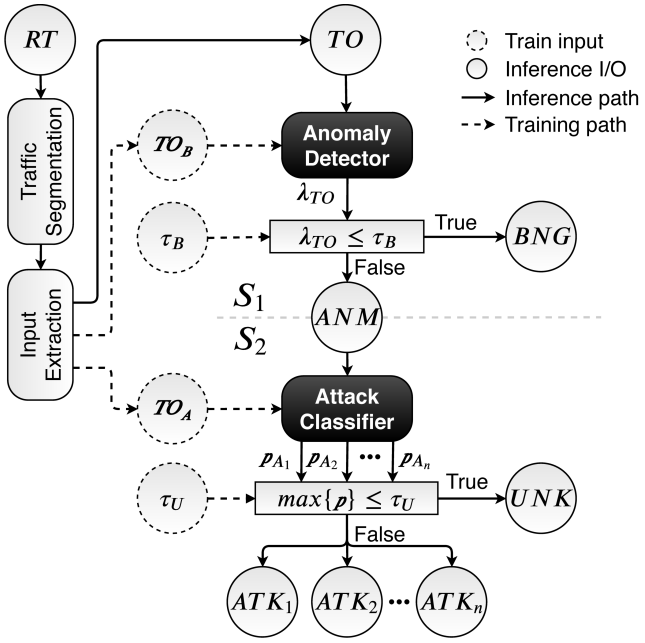
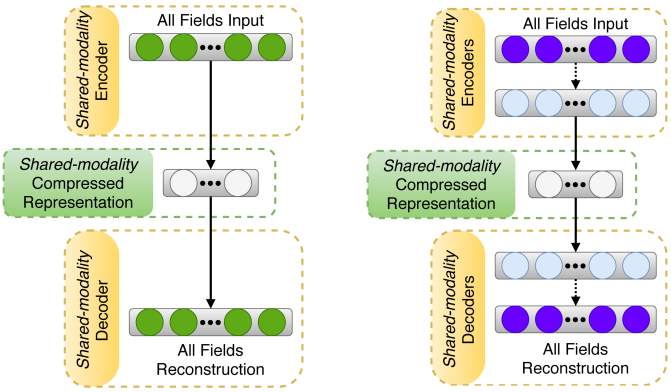


Fig. 1: Architecture of H2ID. Solid arrows represent operational dataflow (inferring), while dashed arrows refer to training phase only. **Legend.** Raw trace (RT), traffic object (TO), benign (B, BNG), anomaly (ANM), attack (A, ATK), thresholds (τ_{B}, τ_{U}), loss (λ), and probability (p).

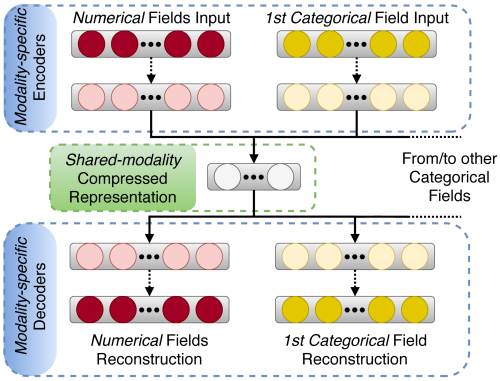
$\{\mathrm{ATK}_1, \ldots, \mathrm{ATK}_n\}$ or detect an unknown attack (UNK) possibly modeling *zero-day* attacks. This task is based on open-set classification methodology (i.e. unknown attacks are not assumed to be seen during the training phase). Differently, no other computation is required if S1 declares the TO as benign (BNG). Notably, in our design H2ID implements *a double-censoring mechanism* based on two independent thresholds τ_{B} and τ_{U} that allow the architecture to be adapted according to the required performance trade-off (cf. Sec. II-C). Looking at the architecture in its entirety, S1 is meant to provide a pre-filtering (aimed at identifying benign TOs) that can be conducted with low overhead also on limited hardware, thus *being suitable for IoT contexts*. Accordingly, S2 is activated *on-request* (based on the verdict of S1): this results in a lightweight path for benign traffic that is not subjected to a second-stage analysis. While this architecture is generalizable (i.e. different choices can be made to implement either stage), in what follows we detail our specific implementation, consisting of a *MultiModal Deep AutoEncoder* (M2-DAE) and an *Machine Learning-based classifier* at stages S1 and S2, respectively.

S1: Anomaly Detection through M2-DAE

In this section, we describe the proposed approach for AD, leveraging a particular class of Deep Learning models, namely the Deep AutoEncoders (DAEs), in an innovative way. In this context, we are interested in leveraging the DAE as an anomaly detector, i.e. by using (during testing phase) the loss metric between the observed input and the DAE-based reconstruction as measure of "anomaly-ness" [23]. The reason for this choice



(a) AE (b) DAE



(c) M2-DAE

Fig. 2: Comparison among AE (a), DAE (b), and M2-DAE (c) architectures.

is the successful application of AEs (and recently DAEs) in several AD works [23–25]. The working principle for AEs is depicted in Fig. 2. AEs are commonly employed as an unsupervised feature extractors, and their aim is to set the output equal to the input $\hat{\boldsymbol{x}}(m) \approx \boldsymbol{x}(m)$, $\forall m$ within the training set, by learning a compressed data representation of benign traffic which minimizes the loss $\lambda(\hat{\boldsymbol{x}}(m), \boldsymbol{x}(m))$ (e.g. the *mean squared error*). For basic AE (Fig. 2a) the first AE layer (i.e. the encoder) provides a lower-dimensional data representation, whereas the second layer (i.e. the decoder) tries to reconstruct the data from the compressed representation. Differently, the DAE [20] is an AE with several encoding and decoding layers (Fig. 2b).

In this paper, we propose an M2-DAE network (Fig. 2c), handling different input types of the TO as separate "modalities". To accomplish this task, the M2-DAE network is made of both modality-specific and shared-modality layers (both encoding and decoding). The proposed M2-DAE has the appeal of learning a compressed representation keeping a more-efficient (viz. less parameters) neural network structure than DAE, which only uses shared (encoding/decoding) layers and processes different input types in a flattened form. In detail, we associate a single modality to all *numerical* input fields and one modality to each *categorical* field. In the latter case, TO-*based* categorical fields (one-instance-per-TO, e.g. port) are represented via the well-known one-hot-encoding format, whereas *packet-based* categorical fields (one-instance-per-packet within the TO, e.g. TCP flags) are arranged herein in a multinomial-encoded format, e.g. the vector average of one-hot-encoding over the packets of the TO. In the case of M2-DAE, the reconstruction loss λ is a weighted sum of the *per-modality* losses.

S2: Machine Learning-based Attack Classification

In this section we discuss the design of the S2 stage. Its goal is to classify each anomalous TO received from the S1 stage, associating it to either one of the known attacks $\{\mathrm{ATK}_1, \ldots, \mathrm{ATK}_n\}$ or the *unknown attack class* (UNK). The generality of the proposed hierarchical architecture allows for any n-class classifier to be employed at the S2 stage; hence, any Machine/Deep Learning-based supervised classifier can be adopted. The sole requirement for each classifier is to be able to provide its soft-output vector $\mathbf{p} = [p_1, p_2, \ldots, p_n]$, p_i being the confidence probability associated to the (known) class ATK_i, needed by the censoring mechanism we adopt (see Sec. II-C). Given the presence of the unknown class, S2 naturally fits the problem of *open-set classification*: during the operational phase, the TOs declared as anomalous may reveal attacks never observed in the training phase, when not matching any known attack. This choice also supports loop-based mechanisms [10] that can adaptively identify new classes for initially unforeseen attacks. Based on the existing literature on AC, for our proposal we consider several options for implementing the S2 stage. In particular, we investigate three different options: Random Forest (RF), Naïve Bayes (NB), and Multi-Layer Perceptron (MLP). RF is a classifier based on an ensemble of different decision trees built at training time exploiting the ideas of bootstrap aggregating and random-feature selection to mitigate overfitting. Differently, NB is a simple probabilistic classifier that assumes class conditional independence of the features, being not the case for real-world problems, but working well in practice and leading to low complexity. Finally, MLP is a particular feedforward neural network, consisting of an output layer and at least one hidden layer, being able to learn non-linear mapping between inputs and outputs.

Double-censoring mechanism

In our design *both stages* implement a *threshold-based mechanism*, whose effect is to censor the output of each stage in case of low-confidence [9]. This provides the architecture with adaptability, with minimal impact on complexity. With regards to S1, an anomaly is detected if $\lambda > \tau_{\mathrm{B}}$ (in such a case the TO is passed to S2). Herein the threshold τ_{B} is designed to balance the **anomaly TPR-FPR tradeoff**. Differently, with reference to S2, the soft-output vector $\boldsymbol{p} = [p_1, p_2, \ldots, p_n]$ of the S2 classifier (gathering the predicted class probabilities of $\mathrm{ATK}_1, \ldots, \mathrm{ATK}_n$) is used to label the TO with the attack

$a \triangleq \arg\max\{\boldsymbol{p}\}$, *only* when $\max\{\boldsymbol{p}\} > \tau_{\mathrm{U}}$. Differently, when $\max\{\boldsymbol{p}\} \leq \tau_{\mathrm{U}}$, the TO is associated to the unknown class UNK. Hence, τ_{U} represents the threshold balancing **discrimination of known attacks against unknown attack detection**. Notably, the two thresholds are *independently* set, allowing for separately tune the sensitivity to anomalies and unknown attacks. Ultimately, this allows to adhere to different performance trade-off requirements.

Deployment Scenarios

The proposed two-stage architecture perfectly fits typical IoT deployment scenarios. In fact, it results in practical benefits achieved by means of deployment modularity and flexibility, as discussed hereinafter. Since IoT infrastructures are intrinsically hierarchical (e.g., made of devices, gateways, and edge/cloud layers), the two constitutive stages can be deployed at different layers and trained separately. The latter aspect limits the need for retraining the whole architecture. Indeed, S1-*Anomaly Detection* can be deployed *locally*, i.e. on each IoT device (or on the local gateways). On the other hand, S2-*Attack Classification* can be implemented in a centralized fashion (e.g., at a remote gateway or at the edge/cloud layer). As IoT devices are often characterized by limited computing, storage, memory, and energy resources, it is paramount that the locally-deployed functionalities do not conflict with these constraints. In our proposal this is guaranteed by the lightweight nature of the S1 stage (implementing M2-DAE as opposed to common DAEs). Moreover, for both the training and operation phases, the proposed H2ID architecture adheres to privacy requirements characterizing IoT networks, where operational (benign) traffic is naturally subjected to privacy concerns (e.g., smart-home or health-related applications, to name a few). More specifically, the proposed two-stage architecture allows both *training and operation phases* to be performed without sharing the benign traffic with the remote nodes implementing AC. Indeed, the AD stage is the only component to be fed with the benign traffic, and is locally deployed. Only the traffic marked as anomalous is sent to the (remote) AC stage. Accordingly, no exchange of benign privacy-concerned traffic is put in practice neither in training nor in operation. This notwithstanding, the training process is not negatively impacted since: i) S1-*Anomaly Detection*—requiring benign traffic only—is still able to learn patterns based on local observations; ii) S2-*Attack Classification* is based on anomalous traffic only, and does not need the—privacy concerning—local traffic for training. Thus, S1 can be pre-trained by using already known traffic patterns of IoT devices connected to the user's network, and fine-tuned with the specific traffic behavior of each locally-connected device. Since the IoT-generated traffic patterns could be influenced by the deployment position (e.g. distance from the cloud/edge) and by user behavior and needs, the fine-tuning phase should take this heterogeneity into account by balancing the importance of both contributions (i.e. location- and behavior-dependent features). Instead, S2 can be more effectively trained by using samples observed from multiple distributed devices. This is in line with the fact that the benign traffic is expected to be characteristic of the specific IoT devices (and of their position in the network) while the attacks are expected to assume their own typical patterns which is dictated by the attacking strategies rather than the attacked network. Accordingly, our architecture naturally allows IoT devices to concur in training S2-*Attack Classification* by exchanging traffic at any granularity via a wide class of existing federated learning approaches [26, 27], without incurring in loss of privacy.

Experimental Evaluation

In this section we report the experimental evaluation of H2ID. In Sec. IV-A, we introduce evaluation metrics, whereas in Sec. IV-B we describe the evaluation setup. Finally, experimental results are shown and discussed in Sec. IV-C.

Evaluation Metrics

Our comparison is based on the performance measures usually adopted in the context of intrusion detection [16]. In particular, for AD we provide a comparison among models in terms of the *True Positive Rate* (TPR, i.e. the ratio of correctly detected anomalies) vs. the *False Positive Rate* (FPR, i.e. the ratio of benign samples incorrectly declared as anomalies), referred to as *Receiver Operating Characteristic* (ROC). To evaluate AC capabilities, we consider *macro F-measure* (*F1*, i.e. the harmonic mean of precision and recall) and confusion matrices to highlight fine-grained error patterns. Metrics refer to values averaged across a 10-fold cross-validation process.

Evaluation Setup

To foster replicability, we provide herein the details of our evaluation setup, in terms of dataset, procedure, and tools.

Dataset. Our results have been obtained leveraging the Bot-IoT dataset[3], collected by Koroniotis et al. [22] in an emulated IoT environment. Attack traffic refers to four attack categories, namely (i) *information gathering* (Scan), *denial of service*, both (ii) *single sourced* (DoS) and (iii) *distributed* (DDoS), and (iv) *information theft* (Theft). For our study, we took advantage of the traces (in *pcap* format) composing the full dataset, in order to extract specific features. Each sample in the dataset, is labeled at multiple levels: (a) attack, (b) category, and (c) subcategory. For each sample in the dataset, we consider only the first two levels. The first one (binary-labelled, i.e. tells whether the sample is part of either a benign communication or an attack) is used for training and evaluating the first detection stage (S1), whereas the second is used to perform attack classification at the second stage (S2). In detail, S1 is trained only with benign traffic traces, whereas S2 is trained via traffic exclusively related to known attacks. These two training phases are completely independent (S1 and S2 are trained on non-overlapping sets of data).

[3]As far as we know, this is the sole dataset releasing PCAP traces of (IoT) attacks and benign behaviors, enabling the extraction of engineered sets of features.

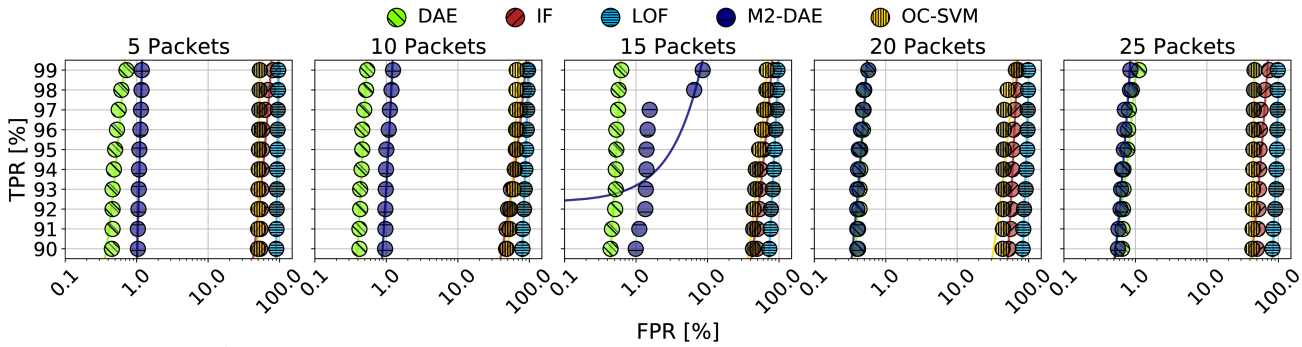


Fig. 3: Partial ROC for baselines and proposed M2-DAE, varying the number of packets.

Traffic Segmentation and Feature extraction. To conduct our analysis we segment raw traces in *biflows*. In addition to TCP and UDP, we consider also ARP, due to its importance in LANs (we rely on MAC addresses to index this traffic[4]). For each biflow, we take into consideration an increasing number of packets (i.e. 5, 10, 15, 20, and 25), thus enabling the evaluation of our proposal in several scenarios (including *early-detection*, in which the very initial sequence of packets is taken into account). From the raw traces we extract the input fields reported in Tab. I. They are differentiated according to the layer of the TCP/IP stack they belong to (i.e. network or transport), their type (i.e. numerical or categorical), and granularity (i.e. TO- or packet-based). To reduce the number of malicious biflows and obtain a balanced dataset, we perform a random undersampling of extracted TOs: we select all the biflows related to benign traffic (i.e. $\approx$ 7k) and at most 500 biflows (selected at random) per attack subcategory, for a total of 4.5k anomalous samples.

Tools. To conduct our experiments, we leverage `keras` (with functional APIs), `scikit-learn` and `python-keras-wrapper` libraries for Python. Deep Learning models are trained with Adadelta optimizer (default parameters), with `batch_size=32`, `epochs=100`, and early stopping (on validation loss) with `patience=1` and `min_delta=`10^{-6}. Moreover, Weka implementation of RF is used with default parameters, whereas NB with `-D` option.

Experimental Results

Our performance assessment consists of *two* phases: **S1 analysis** and **IDS analysis**. In **S1 analysis** we compare via ROC analysis M2-DAE against three One-Class Classifiers (OCCs) commonly used in AD, and a standard DAE, inspired by Meidan et al. [20]. Among the OCCs in the literature, we choose three state-of-the-art models, i.e. One-class Support Vector Machine (OC-SVM), Isolation Forest, and Local Outlier Factor. OCCs are fed with the same engineered input as (M2-)DAE, which consists of 4+4 encoding/decoding layers with *relu* activations. In **IDS analysis** H2ID is compared with a multi-class MD (Multi-MD), designing both for open-set classification. The two approaches are compared based on the F1 score of the open-set problem ($\{\text{ATK}\}_{i=1}^{n} \cup \text{UNK}$) vs. the unknown threshold (τ_{U}). A former comparison of RF, NB, and MLP ability in performing AC and MD tasks aims to select the best model for S2 and Multi-MD, respectively.

[4]MAC addresses can be used to extend the 5-tuple identifying each biflows—obtaining a *7-tuple*. This traffic object allows to catch mismatches between layer-2 and layer-3 addresses, enabling the framework to naturally manage other kinds of anomalous (e.g. spoofed) traffic. However, since in the considered dataset we observed a perfect match between 5-tuples and 7-tuples, we omit this specification without prejudice for the generality of the approach or the correctness of reported results.

TABLE I: Input data extracted from Bot-IoT dataset.

TCP/IP Stack Layer	Field Name	Stat(s)	Type	Granularity	#
Transport	`n. wrong fragments`	+	N	B	3
	`destination port`	n.d.	C	B	1
	`payload bytes seq.`	A	C	P	3
	`payload length`	A, S, m, M	N	P	12
	`TCP flags combin.`	+	C	P	3
	`TCP window size`	A, S, m, M	N	P	12
Network	`n. packets`	+	N	B	3
	`byte rate`	n.d.	N	B	3
	`duration`	n.d.	N	B	3
	`inter-arrival-time`	A, S, m, M	N	P	12
	`protocol`	n.d.	C	B	1
	`time-to-live`	A, S, m, M	N	P	12

Legend:
Input Dimensionality (#);
Stats: sum (+), average (A), std dev (S), minimum (m), maximum (M);
Type: Numerical (N), Categorical (C);
Granularity: Biflow-based (B), and Packet-based (P).
Fields present bidirectional, upstream, and downstream representation.
`destination port` and `protocol` are only bidirectional.

Results of S1 analysis. In Fig. 3, the partial ROC (i.e. with $\text{TPR} \in (90, 99)\,\%$) compares the experimented models at S1. Notably, M2-DAE and DAE outperform OCCs models, showing an almost constant FPR $\leq 1\%$ on varying number of packets, against $> 40\%$ FPR reached by OCCs. Although DAE and M2-DAE report similar results (with the former resulting in a lower FPR in most of the cases), the latter boasts a less complex model, with a reduction of trainable parameters (i.e. weights and bias) up to a factor of $4\times$ (i.e. from $\approx 12M$ of DAE to $\approx 3M$ of M2-DAE). Therefore, it provides a better trade off, considering that the reduction of complexity is a desired property for ML approaches in IoT context, when these

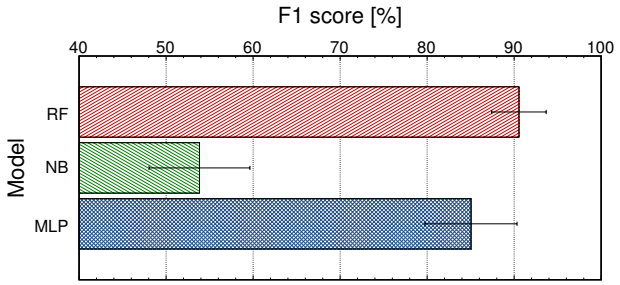


Fig. 4: Comparison among F1 score of ML models for MD task. Variation values are provided for a C.I. of 99.7%.

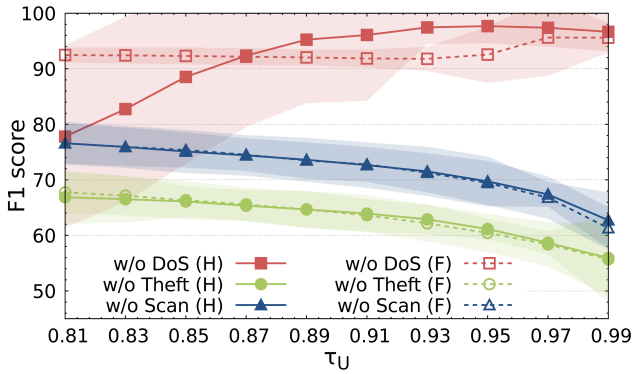
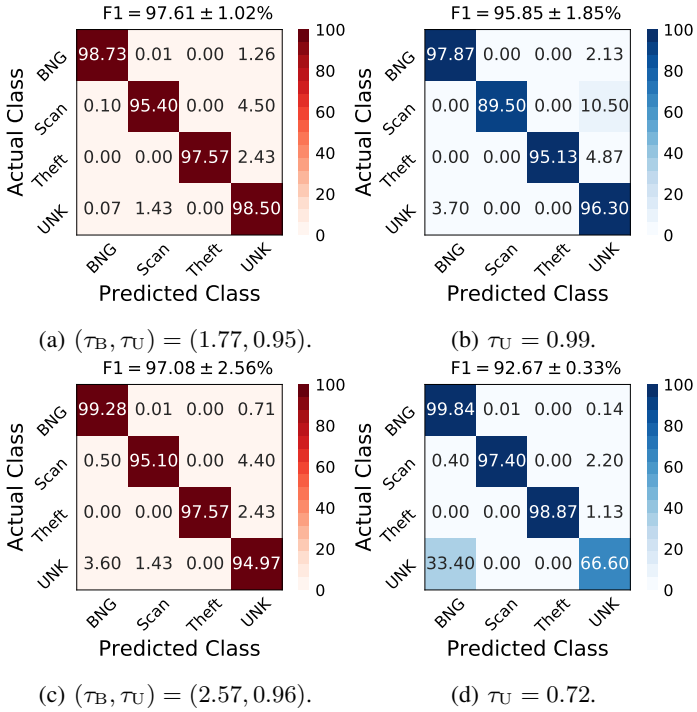


Fig. 5: F1 score vs. τ_U for H2ID (H) and Multi-MD (F) in an open-set approach. Variation values are provided for a C.I. of 99.7%.



(a) $(\tau_B, \tau_U) = (1.77, 0.95)$. (b) $\tau_U = 0.99$.

(c) $(\tau_B, \tau_U) = (2.57, 0.96)$. (d) $\tau_U = 0.72$.

Fig. 6: Comparison between confusion matrices of H2ID (left) and Multi-MD (right). (a) and (b) report the results of the configurations resulting in the best F1 score achievable. (c) and (d) report the results of the configurations resulting in the lowest FPR ($\leq 1\%$).

algorithms are designed to run on hardware with limitations. Accordingly, we use the M2-DAE as model for S1. Similarly, we opt for considering 5 packets per TO, in order to support early detection of the anomalies with low overhead.

Results of IDS analysis. To select the best models for S2 and Multi-MD, we have preliminarily evaluated the aforementioned Machine Learning models (i.e. RF, NB, and MLP) in tackling AC and MD, respectively. The outcomes of the comparison are summarized in Fig. 4, reporting the results for MD task via RF, NB, and MLP on the overall dataset in terms of F1 score. Results pertaining to AC are omitted for brevity. Accordingly, we select the RF as model for both MD and AC task, because it reported the best performance and lowest variability. Then, for both H2ID and Multi-MD models, the open-set analysis is performed by removing one of the known attack classes from the training set and considering it as *unknown* (UNK class). This procedure is iterated for all the attack classes. It is worth to underline that DoS and DDoS classes are merged and referred as DoS. Then, a threshold τ_U is applied to the soft-output of the classifiers: TOs whose highest predicted class probability is $\leq \tau_U$ are declared as UNK. In Fig. 5, we show the F1 score vs. τ_U to investigate their ability to recognize UNK class, for both H2ID and Multi-MD; for H2ID, τ_B is fixed to obtain a $\approx 1\%$ FPR (the TPR corresponds to $\approx 99\%$ (Fig. 3)). Results highlight no significant difference in recognizing Theft and Scan as the UNK class, whereas for DoS class H2ID can potentially provide significant gains over Multi-MD in terms of F1 score. Given the trade-off introduced by tuning τ_U for Multi-MD and (τ_B,τ_U) for H2ID, we inspect fine-grained classification results (w/o DoS) considering (i) the configurations achieving the best F1 score (Figs. 6a–6b), (ii) those ensuring a FPR $\leq 1\%$ (Figs. 6c–6d): H2ID achieves $+1.76\%$ ($+4.41\%$) F1 score in the former (latter) case. Confusion matrices confirm the effectiveness of H2ID in capturing unknown attacks, as opposed to Multi-MD. This recognition capacity reduces with the decreasing of FPR, but it is always better than Multi-MD. Finally, the use of a less complex classifier (no benign samples) gives to H2ID a gain in capturing attack classes.

Conclusions and Future Perspectives

The widespread diffusion of IoT devices is implying growing security issues, leading to novel discovery of vulnerabilities and attacks. In this study, we have proposed a hierarchical approach for intrusion detection, named H2ID, facing the problem at different levels of granularity. At first stage, our framework performs lightweight anomaly detection, based on a novel M2-DAE (and feature representation), suitable for on-device implementation. Then, the traffic identified as anomalous is processed to recognize both known and unknown attacks, to apply countermeasures and (possibly) enrich the attack knowledge base. Our H2ID is evaluated on the recently-released Bot-IoT dataset. Our H2ID demonstrates higher ca-

pabilities in both performing anomaly detection and recognizing unknown attacks, with respect to the best-performing multi-class misuse detector, especially for DoS and DDoS attacks. Furthermore, H2ID adheres to scalability and privacy requirements characterizing IoT networks, as it supports distributed deployments leveraging the hierarchical nature of IoT infrastructures while not requiring operational (benign) traffic to be shared among different remote parties. Future work will explore threshold design for specific use cases, and further expand the classes of attacks and their granularity, as new datasets are made available. Moreover, privacy-preserving distributed implementations will be explored leveraging the hierarchical nature of our H2ID.

ACKNOWLEDGEMENTS

This work is partially funded by the Italian Research Program "PON AIM *Attraction and International Mobility*, Azione I.2 Linea 1, *Mobilità dei Ricercatori*" (Codice proposta attività AIM1878982-2 CUP E56C19000330005).

REFERENCES

[1] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *IEEE Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[2] A. Kumar and T. J. Lim, "Edima: early detection of IoT malware network activity using machine learning techniques," in *IEEE WF-IoT'19*, pp. 289–294.

[3] E. Ronen, A. Shamir, A.-O. Weingarten, and C. O'Flynn, "IoT goes nuclear: Creating a ZigBee chain reaction," in *IEEE SP'17*, pp. 195–212.

[4] A. Dainotti, A. Pescapé, and G. Ventre, "Worm traffic analysis and characterization," in *IEEE ICC'07*, pp. 1435–1442.

[5] A. Dainotti, A. Pescapé, and G. Ventre, "A cascade architecture for DoS attacks detection based on the wavelet transform," *IOS Press Journal of Computer Security*, vol. 17, no. 6, pp. 945–968, 2009.

[6] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 2, pp. 445–458, June 2019.

[7] A. Dainotti, F. Gargiulo, L. I. Kuncheva, A. Pescapè, and C. Sansone, "Identification of traffic flows hiding behind TCP port 80," in *IEEE ICC'10*, 2010, pp. 1–6.

[8] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapè, "MIMETIC: Mobile encrypted traffic classification using multimodal deep learning," *Elsevier Computer Networks*, vol. 165, p. 106944, 2019.

[9] A. Montieri, D. Ciuonzo, G. Bovenzi, V. Persico, and A. Pescapé, "A dive into the dark web: Hierarchical traffic classification of anonymity tools," *IEEE Trans. Netw. Sci. Eng.*, 2019.

[10] C. Zhang, J. Jiang, and M. Kamel, "Intrusion detection using hierarchical neural networks," *Pattern Recognition Letters*, vol. 26, no. 6, pp. 779–791, 2005.

[11] A. Khan and S. Khan, "Two level anomaly detection classifier," in *IEEE ICCEE'08*, pp. 65–69.

[12] C. Guo, Y. Ping, N. Liu, and S.-S. Luo, "A two-level hybrid approach for intrusion detection," *Neurocomputing*, vol. 214, pp. 391–400, 2016.

[13] Q. Schueller, K. Basu, M. Younas, M. Patel, and F. Ball, "A hierarchical intrusion detection system using support vector machine for SDN network in cloud data center," in *IEEE ITNAC'18*, pp. 1–6.

[14] H. Lu and J. Xu, "Three-level hybrid intrusion detection system," in *IEEE ICIECS'09*, pp. 1–4.

[15] S.-Y. Ji, B.-K. Jeong, S. Choi, and D. H. Jeong, "A multi-level intrusion detection method for abnormal network behaviors," *Elsevier JNCA*, vol. 62, pp. 9–17, 2016.

[16] P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, "A detailed investigation and analysis of using machine learning techniques for intrusion detection," *IEEE Commun. Surveys Tuts.*, 2018.

[17] H. Yao, D. Fu, P. Zhang, M. Li, and Y. Liu, "MSML: A novel multi-level semi-supervised machine learning framework for intrusion detection system," *IEEE Internet Things J.*, 2018.

[18] J. Li, Z. Zhao, R. Li, H. Zhang, and T. Zhang, "AI-based two-stage intrusion detection for software defined IoT networks," *IEEE Internet Things J.*, 2018.

[19] I. Ullah and Q. H. Mahmoud, "A two-level hybrid model for anomalous activity detection in IoT networks," in *IEEE CCNC'19*, pp. 1–6.

[20] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-baIoT —network-based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.

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

[22] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *FGCS*, vol. 100, pp. 779–796, 2019.

[23] G. Andresini, A. Appice, N. Di Mauro, C. Loglisci, and D. Malerba, "Exploiting the auto-encoder residual error for intrusion detection," in *IEEE EuroS&PW'19*.

[24] H. Mac, D. Truong, L. Nguyen, H. Nguyen, H. A. Tran, and D. Tran, "Detecting attacks on web applications using autoencoder," in *ACM SoICT'18*, pp. 416–421.

[25] F. A. Khan, A. Gumaei, A. Derhab, and A. Hussain, "TSDL: A two-stage deep learning model for efficient network intrusion detection," *IEEE Access*, 2019.

[26] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," *FGCS*, vol. 82, pp. 761 – 768, 2018.

[27] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, Jan. 2019.