Class Incremental Learning for Network-Agnostic Intrusion Detection Systems

Francesco Cerasuolo, Giampaolo Bovenzi, Antonio Montieri, Antonio Pescapè
University of Napoli Federico II (Italy)
{francesco.cerasuolo, giampolo.bovenzi, antonio.montieri, pescape}@unina.it

***Abstract*—The proliferation of Internet of Things (IoT) devices has significantly intensified cybersecurity concerns, highlighting the need for robust, adaptable, and privacy-preserving Network Intrusion Detection Systems (NIDS). A major challenge lies in the heterogeneity of IoT environments, which complicates the generalization of detection models across different network contexts. In this work, we propose a network-agnostic NIDS enhanced through Class Incremental Learning (CIL), allowing the integration of legitimate traffic from different networks without requiring retraining from scratch or exposing sensitive data. Our approach ensures efficient, continuous adaptation to new environments while maintaining strong detection capabilities. To assess the effectiveness of the proposed solution, we evaluate several CIL techniques in two deployment scenarios: within the same network and across different networks. Results show that the best-performing CIL methods perform comparably to an upper-bound model trained from scratch, with minimal knowledge degradation when adapting to previously unseen benign traffic. These findings demonstrate the practicality of CIL-based NIDS for real-world, heterogeneous IoT environments.**

***Index Terms*—Class Incremental Learning, Network Intrusion Detection System, Misuse Detection, Internet of Things**

I. Introduction

In recent years, the widespread adoption of Internet of Things (IoT) devices has significantly transformed Internet communication, with autonomous systems now capable of collecting and exchanging data [1]. This expanded technological landscape, however, has also amplified cybersecurity challenges, as adversaries can rapidly exploit newly discovered vulnerabilities. As a result, the demand for robust and adaptable security measures in the IoT domain is more critical than ever. The heterogeneity of IoT deployments, characterized by diverse devices, communication protocols, and traffic patterns, further complicates this scenario.

Within this context, Network Intrusion Detection Systems (NIDS) are essential for safeguarding network security, as they monitor traffic for malicious activity, enabling early detection and response to potential threats [2]. To this aim, recent NIDS increasingly rely on *data-driven* approaches, namely Machine Learning (ML) and Deep Learning (DL), which require large volumes of network-specific traffic to be trained effectively to distinguish between benign and malicious behaviors [3].

This work is supported by the Italian PNRR MUR "Centro Nazionale HPC, Big Data e Quantum Computing, Spoke9 - Digital Society & Smart Cities" and the "xInternet" Project within the PRIN 2022 program (D.D.104-02/02/2022), funded by MUR. This manuscript reflects only the authors' views and opinions and the Ministry cannot be considered responsible for them.

In this work, we propose a network-agnostic NIDS trained on known attack patterns, while supporting personalization to benign, network-specific traffic without compromising data privacy, namely without exchanging sensitive data. To address this, we leverage Class Incremental Learning (CIL), which allows the continuous enhancement of a pre-trained model with novel knowledge, without requiring *full retraining*. This approach not only reduces computational costs and training time but also ensures that the NIDS remains up to date with evolving traffic patterns and threats.

The *main contributions* of this paper are the following: (i) we propose a network-agnostic NIDS capable of integrating knowledge of legitimate, environment-specific IoT traffic, supporting continuous adaptation while preserving data privacy; (ii) we conduct a thorough evaluation of our NIDS, investigating the effectiveness of established CIL techniques for binary and multiclass misuse detection (bMD and mMD) in the same network as the attack traffic (*intra-dataset*) and across different networks (*cross-dataset*).

The rest of the manuscript is organized as follows. Section II surveys related work on applying CIL for the design of NIDS, highlighting current limitations. Section III details the proposed methodology and the CIL techniques employed in this study. Section IV describes the experimental setup, while Section V presents and discusses the corresponding results. Finally, Section VI concludes the paper and outlines directions for future research.

II. Related Work

This section provides an overview of the literature on CIL in the context of NIDS, with a focus on approaches that allow detection systems to adapt to evolving threats—such as novel attack types, behavioral shifts, and emerging anomalies—without retraining from scratch. Notably, CIL is particularly well-suited to dynamic cybersecurity scenarios, as it enables the incremental integration of new knowledge while retaining previously acquired information, thus mitigating the risk of catastrophic forgetting.

CIL methods are commonly categorized into three major families based on how they expand models to integrate new knowledge [4]: (i) *fine-tuning*, which expands the model head and retrains the entire network using new data; (ii) *fixed-representation*, which extends the output layer while freezing parts of the backbone or earlier heads, updating only the non-frozen parts; (iii) *model-growth*, which introduces new

models or layers to learn additional knowledge. Most reviewed approaches adopt strategies from the fine-tuning family. Notable exceptions include I^{2}RNN [5] and SPCIL [6], which follow *model-growth* paradigms. On the other hand, *fixed-representation* techniques remain underexplored in the context of NIDS, while *model-growth* methods are discarded in this study due to scalability concerns.

CIL is inherently challenged by two key issues: *catastrophic forgetting*, namely the tendency of models to lose previously learned knowledge when adapting to new classes, and *intransigence*, that is the reduced ability to effectively incorporate novel information. To address these limitations, various mitigation strategies have been proposed: (i) *rehearsal* techniques preserve a small subset of old representative samples from past classes to reinforce prior knowledge during adaptation; (ii) *regularization* methods—both *implicit*, e.g., Knowledge Distillation (KD), and *explicit* (e.g., constraints on parameter updates)—aim to preserve previously acquired representations; (iii) *bias correction* compensates for the bias towards newly learned classes at inference time.

The most widely adopted baseline for CIL are FT [7] and FT-Mem [8], both employed in several recent works [6, 9, 10, 11, 12]. Other rehearsal-based approaches include GSS, ER, ASER, AGEM, SSR, NCM, and SLDA [13], as well as BFS-NIDS [9]. Regularization-based approaches encompass LwF [14], adopted in [10, 12, 13], EWC [6, 13], and SimpleCIL [6]. Hybrid methods that combine rehearsal and distillation include iCaRL [15] and its network traffic adaptation iCaRL+ [16], employed in [6, 9, 12, 13]. Similarly, the two-stage EEIL is used for network intrusion detection in [10]. BiC [17] incorporates all three strategies—rehearsal, regularization, and bias correction—and is exploited by Cerasuolo et al. [11, 12].

Regarding the application domain, most existing studies focus on mMD, where the aim is to distinguish between benign traffic and multiple attack categories. Other works (e.g., [11]) tackle bMD, in which traffic is classified as either benign or malicious. A more refined approach is proposed in [12], where detection is performed in two stages: an initial bMD phase, followed by an attack classification stage that categorizes the identified malicious traffic into specific attack types.

Concerning model architectures, some studies use traditional ML techniques—such as decision trees, support vector machines [18], multi-layer perceptrons [10], or ensemble methods [18]—yet the majority of recent work favors DL architectures [5, 6, 9, 11, 12, 13]. These DL-based architectures include Convolutional Neural Networks (CNNs) [11, 12, 13], Long Short-Term Memory (LSTM) [5], transformers [9], and Residual Networks (ResNet) [6].

Finally, most of the reviewed approaches rely on *post-mortem* analysis, utilizing features extracted from the complete traffic flow. However, a subset of studies focuses on *early detection*, employing raw packet-level features derived solely from the first N_p packets [8]. These early detection strategies hold promise for real-time threat mitigation, as they can reduce both detection latency and computational overhead.

In this work, we take an in-depth look at the adaptation of NIDS to different network environments. Unlike prior studies that focus on incremental updates using classes from the same dataset, we explore adaptation using benign traffic from a different network. In contrast to Cerasuolo et al. [11], our emphasis is specifically on benign traffic, leveraging two distinct network domains to evaluate generalization and adaptability.

III. Methodology

CIL is a paradigm designed to extend pre-trained models with new classes while preserving previously acquired knowledge, eliminating the need for retraining from scratch [8]. The main objective of CIL is to update a model trained on an initial set of classes $\mathcal{C}^{old}$ and their corresponding data $\mathcal{D}^{old}$, by introducing new classes $\mathcal{C}^{new}$ and their data $\mathcal{D}^{new}$. The goal is to build a unified model capable of classifying across the entire label space $\mathcal{C}^{all} = \mathcal{C}^{old} \cup \mathcal{C}^{new}$. Rather than using the full dataset (i.e. $\mathcal{D}^{old} \oplus \mathcal{D}^{new}$), CIL typically relies on a subset of stored past samples $\mathcal{D}^{mem}$ and uses $\mathcal{D} = \mathcal{D}^{new} \oplus \mathcal{D}^{mem}$ during training. In our specific scenario, we aim to incrementally improve a NIDS originally trained on a specific network (attack plus benign classes) to be deployed on a network with different legitimate devices, namely with *different benign traffic*.

Hereinafter, we introduce the CIL approaches evaluated in this work. All the selected methods fall within the *fine-tuning* family [4]. First, we leverage two baseline methods—FT and FT-Mem—which represent the most straightforward strategies for incremental learning. Then, we consider more advanced techniques incorporating one or more mitigation strategies to address forgetting challenges. In addition, we include a model *trained from scratch* on the full dataset, denoted as Scratch, which serves as an upper bound reference for performance comparison. Detailed descriptions of each approach are provided below.

Fine-Tuning (FT) [7]. FT is the simplest and most naive approach to update a model incrementally. It involves retraining the entire model using only samples from newly observed attacks, without retaining any previous data.

Fine-Tuning with Memory (FT-Mem) [8]. FT-Mem extends the FT method by introducing a small memory buffer that stores a subset of previously seen samples from both attack and benign traffic. During training, these stored examples are replayed to reinforce prior knowledge and mitigate forgetting.

Learning Without Forgetting (LwF) [14]. LwF is a *memory-free* approach (i.e. $\mathcal{D} = \mathcal{D}^{new}$) that pioneered the use of KD in incremental learning settings [19]. Knowledge from a previously trained model (the teacher) is distilled into the updated model (the student) to retain old information while learning new classes. The loss function comprises three components: a classification loss ($\mathcal{L}_{class}$) for new classes, and two auxiliary terms—distillation $\mathcal{L}_{dist}$) and regularization ($\mathcal{L}_{reg}$)

TABLE I
DESCRIPTION OF IoT-23 ATTACK CLASSES.

Attack	Description
Attack	Connection used by the infected device to launch an attack on another host
Benign	No suspicious or malicious activity detected in the connection
C&C	Infected device is communicating with a Command and Control (C&C) server
HeartBeat	Connection used by the C&C server to monitor the infected device
DDoS	Infected device is performing a Distributed Denial of Service (DDoS) attack
Okiru	Connection exhibits behavior typical of an Okiru botnet
Portscan	Connection is a horizontal port scan, gathering information for potential attacks

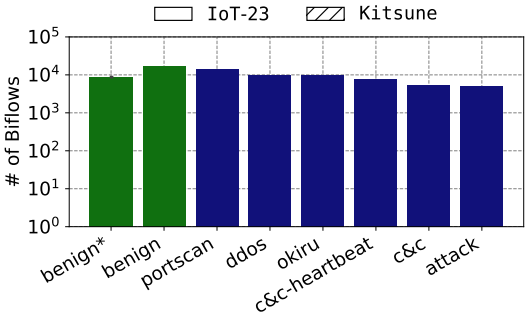
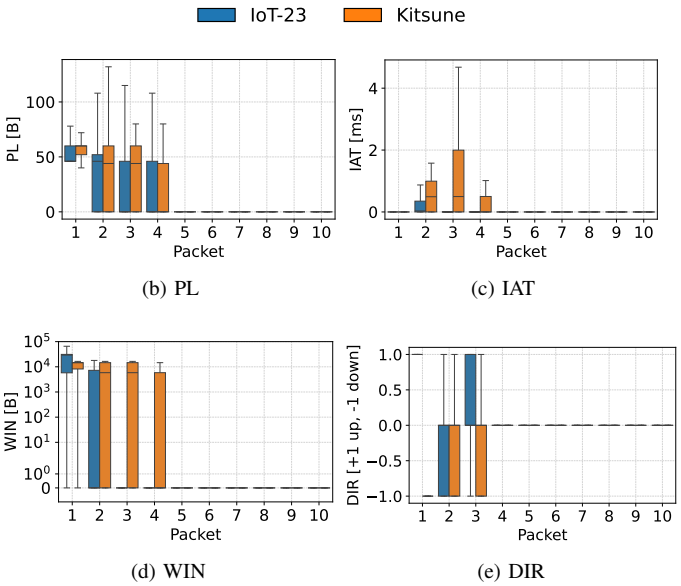


Fig. 1. Distribution of per-class samples of the IoT-23 dataset (without hatches) and benign class for Kitsune dataset (with hatches). Green bars represent benign classes across the two datasets.



(b) PL (c) IAT

(d) WIN (e) DIR

Fig. 2. Comparison of the benign traffic of IoT23 and Kitsune using the first 10 packets for each feature.

losses—for preserving prior knowledge. These components are balanced using the hyperparameters λ_{dist} and λ_{reg}.[1]

Incremental Classifier and Representation Learning (iCaRL+) [16]. iCaRL+ is an adaptation of the original iCaRL [15] tailored for traffic classification tasks. It employs a *rehearsal-based strategy* using herding selection, dynamically expands the output layer (using a softmax classifier instead of the original nearest mean classifier), and uses a composite loss function combining classification and distillation terms equally weighted (i.e. $\lambda_{class} = \lambda_{dist} = 1$).

Bias Correction (BiC) [17]. BiC addresses the common issue of *bias toward newly learned classes* in incremental learning scenarios. To mitigate this, it appends a small correction layer to the model head, parameterized by scaling (α) and shifting (β) factors. These parameters are applied exclusively to the logits of new classes, following the transformation: $\bar{o}(x) = \alpha \cdot o^{new}(x) + \beta$, while the logits corresponding to previously learned classes (o^{old}) remain unchanged. The training process is divided into three stages: (i) the whole model is trained using a composite loss function combining classification and KD, with class-proportional weighting terms ($\lambda_{class} = |\mathcal{K}^{new}|/|\mathcal{K}^{all}|$, $\lambda_{dist} = 1-\lambda_{class}$); ($ii$) the backbone and classification head are frozen; and (iii) the correction layer is fine-tuned using a small calibration set to estimate the optimal values of α and β.

IV. EXPERIMENTAL SETUP

This section provides details on the dataset and pre-processing operations performed (Sec. IV-A), the model architecture (Sec. IV-B), the experimental scenarios (Sec. IV-C), and the evaluation metrics (Sec. IV-D).

A. Dataset and Pre-processing Operations

In this work, we leverage the IoT-23 dataset [20]. IoT-23 was collected during 2018–19 at the Stratosphere Laboratory of the Czech Technical University. It comprises 23 traffic traces captured in a controlled IoT network environment with unrestricted Internet access. Of these 23 traces, 20 correspond to malicious traffic, while the remaining 3 represent benign network activity.

The malicious traffic originated from a Raspberry Pi deliberately infected with specific malware. In contrast, the benign traffic was generated by three real-world IoT devices operating under normal conditions: (i) a Philips Hue Smart LED Lamp, (ii) an Amazon Echo smart speaker, and (iii) a Somfy Smart Door Lock. Table I briefly describes the attack classes present in IoT-23. Further details about the labeling are available on the IoT-23 website [20].

Additionally, we employ the benign traffic from the Kitsune dataset [21], which contains traffic captured from an IP-based commercial surveillance system and IoT devices encompassing 4 HD surveillance cameras, 9 IoT devices, and 3 PCs. For more details, see [21].

As a traffic object, we consider biflows[2] and, to guarantee *earliness*, we extract 4 informative fields from the header of the first 10 packets of each biflow: packet length (PL), inter-arrival time (IAT), packet direction (DIR), and TCP Window Size (WIN). Given the severe class imbalance present in the IoT-23 dataset, we first discard all classes containing fewer than 500 biflows. To further address imbalance among the

[1] LwF sets λ_{reg} to a fixed value of $5 \cdot 10^{-4}$.

[2] A *biflow* (i.e. bidirectional flow) groups together packets sharing the same quintuple—source/destination IP address, source/destination port number, and transport-layer protocol—in both directions.

remaining classes, we apply random downsampling to the majority classes—namely, *portscan*, *okiru*, *ddos*, *attack*, *c&c*, *c&c-heartbeat*, and *benign*—retaining, for each, a number of biflows equal to the per-class median within this group. Finally, we merge the highly overlapping classes *c&c-heartbeat* and *c&c-heartbeat-attack* into a single class, simply labeled as *c&c-heartbeat*. Figure 1 shows the distribution of biflows for each attack class of `IoT-23` and for the benign one of both `IoT-23` and `Kitsune`.

As shown in Fig. 2, the benign traffic exhibits substantial differences between the two datasets. This divergence is particularly evident in the first four packets, where the feature distributions display distinct median values and higher skewness. From the 4^{th} packet onward, however, the two datasets show more aligned behavior, with comparable median values across all considered features.

B. Model Architecture

For the experiments, we rely on a 2D-CNN classifier. Notably, CNN-based architectures are widely used for both traffic classification and intrusion detection tasks [8, 22, 23, 24]. The architecture is composed of 2 convolutional layers, followed by pooling and batch normalization, with a final fully-connected layer providing the model outputs.

At each incremental step, the network backbone remains unchanged, while the head is expanded with neurons for the new classes. Training starts from the previous phase's weights, using the base model weights for the first incremental episode. It is worth noting that the methodology described in this work is independent of the DL architecture used and is therefore applicable to alternative architectures.

C. Evaluation Scenarios

We consider a scenario involving the addition of a single new class. Specifically, the *benign class* is added as the new class, while the remaining *attack classes* are treated as old ones. We evaluate two configurations using different benign classes from the `IoT-23` and `Kitsune` datasets. In the first configuration, the benign class is taken from `IoT-23`, resulting in `IoT-23`[B+M] (also referred to as *intra-dataset* scenario); in the second configuration, the benign class is taken from `Kitsune`, yielding `IoT-23`[M]+`Kitsune`[B] (also called *cross-dataset* scenario).

For the experiments, we leverage the FACIL framework [4], adapted for intrusion detection tasks. We start from the same base model by adding a different benign class each time (from `IoT23` or `Kitsune`). Each experiment is carried out with 200 epochs, an initial learning rate of 0.1, a decay of 3.0, a patience of 20 epochs, and a batch size of 64. For the memory-based approaches, we employ 1k old samples (i.e. $1000/|\mathcal{K}^{old}|$ samples from each old class).

D. Evaluation Metrics

To evaluate the performance of CIL approaches, we use the F1 score (briefly, F1), which is the harmonic mean of precision and recall. For a multi-class problem, it is computed as: $\mathrm{F1}(\boldsymbol{\theta}, \mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{i \in \mathcal{C}} \frac{2 \cdot \mathrm{Precision}_i \cdot \mathrm{Recall}_i}{\mathrm{Precision}_i + \mathrm{Recall}_i}$, where $\mathcal{C}$ represents a generic set of classes and θ a generic model (learned *from-scratch* or in an incremental way).

To evaluate the deviation of the incremental model from the `Scratch` upper-bound, we compute the drop in terms of F1. Additionally, we break down this metric into 3 sub-metrics, according to the 3 set of classes—viz. All, Old, and New—to better understand all the forgetting phenomena. Hence, we obtain *DropOld*, *DropNew*, and *DropAll*. Notably, DropOld measures the model's ability to retain knowledge from the base classes, while DropNew reflects its effectiveness in learning newly introduced classes. Lastly, DropAll captures the overall performance gap between the incremental model and the ideal scenario of `Scratch`. Lower values indicate better performance (i.e., a more effective CIL approach).

Additionally, we leverage the *accuracy* and *Area Under ROC Curve (AUC)*. Accuracy measures the proportion of correct predictions among all predictions made. AUC assesses the model's ability to distinguish between classes across different classification thresholds. Therefore, while accuracy provides a general sense of correctness, AUC offers a more nuanced view of a model's discriminative power, particularly under imbalanced conditions or when the cost of false positives and false negatives differs.

V. Evaluation

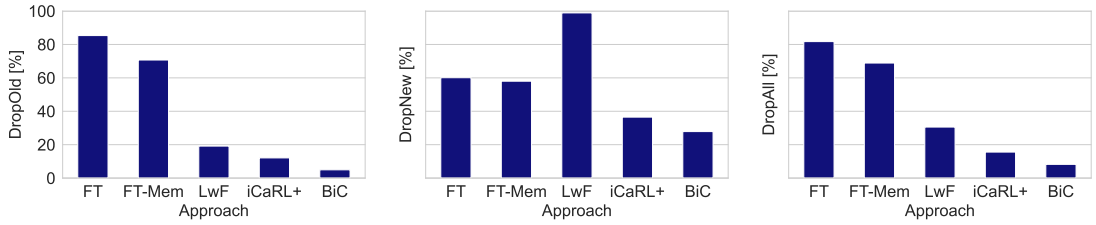
In this section, we first analyze model performance on the multi-class misuse detection task (Sec. V-A). We then turn to a binary classification setting, distinguishing between benign and malicious traffic to evaluate performance on a simpler yet practically relevant problem (Sec. V-B).

A. Multi-class Misuse Detection

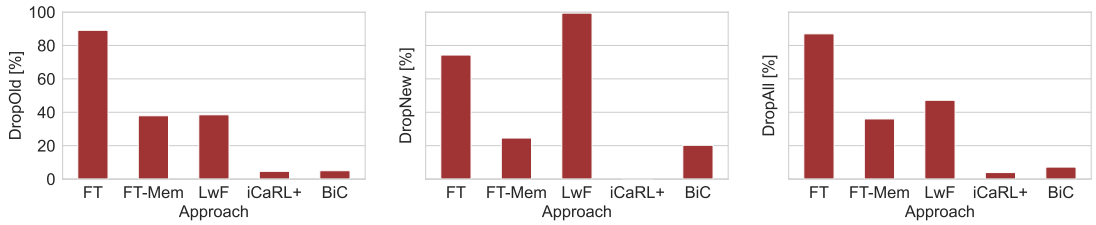
Hereinafter, we assess the *mMD* considering the whole set of classes—viz. benign and attack classes—in the two considered scenarios (i.e. `IoT-23`[B+M] and `IoT-23`[M]+`Kitsune`[B]). Figure 3 shows the performance in terms of DropOld and DropNew for both scenarios.

As expected, `FT` significantly suffers from catastrophic forgetting, showing a $\approx$ 85% (resp. $\approx$ 89%) DropOld in `IoT-23`[B+M] (resp. `IoT-23`[M]+`Kitsune`[B]) scenario and a $\approx$60% (resp. $\approx$74%) DropNew. Adding memory to `FT` (i.e. `FT-Mem`), DropOld becomes lower but is still significant ($\approx$ 70% and $\approx$ 38%). Moreover, `FT-Mem` reaches low F1 on $\mathcal{K}^{new}$ as well, with $\approx$ 58% and $\approx$ 25% DropNew. On the other hand, `LwF` suffers significantly from *intransigence* as it exhibits $\geq$98% DropNew in both scenarios.

Lastly, each of the two scenarios yields the best performance with a different approach. In the *intra-dataset* scenario, `BiC` turns out to be the top-performer with a $\approx$5% DropOld, $\approx$28% DropNew, and $\approx$ 8% DropAll. Conversely, in *cross-dataset*, the best performing approach is `iCaRL+` that exhibits $\approx$5% DropOld but achieves a slight improvement (+0.5%) w.r.t. `Scratch` on the $\mathcal{K}^{new}$, obtaining a $\approx$4% DropAll.



(a) IoT-23[B+M]



(b) IoT-23[M]+Kitsune[B]

Fig. 3. Performance in terms of DropOld and DropNew for the two scenarios: (a) IoT-23[B+M] and (b) IoT-23[M]+Kitsune[B].

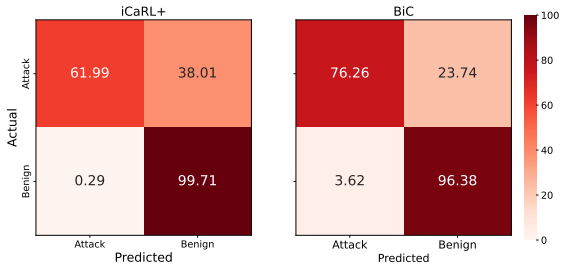


Fig. 4. Confusion matrices for IoT-23[B+M].

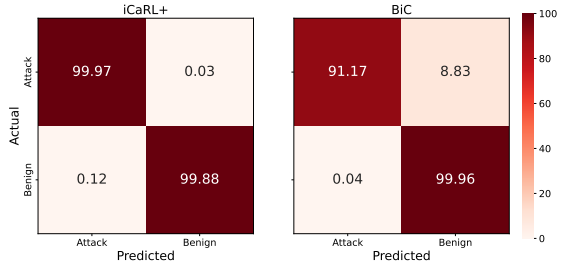


Fig. 5. Confusion matrices for IoT-23[M]+Kitsune[B].

B. Binary Misuse Detection

Then, we evaluate the performance of the incremental approaches in solving a bMD—namely, distinguishing between benign and malicious traffic. To address this simpler task, we aggregate the soft outputs of the attack classes by summing them. This yields a single probability representing malicious traffic, which is then contrasted with the probability of benign traffic. Figures 4 and 5 show the confusion matrices of the two best approaches (i.e. iCaRL+ and BiC) in two different scenarios. For brevity, the confusion matrices for the FT, FT-Mem, and LwF are omitted since the first two approaches suffer from catastrophic forgetting, resulting in a strong prediction bias toward the benign class, while LwF exhibits intransigence, leading to a bias toward the malicious class.

From the confusion matrices of the *intra-dataset* scenario (Fig. 4), it is evident that BiC obtains better performance than iCaRL+. This is due to more severe catastrophic forgetting in iCaRL+, where $\approx$ 38% of attacks are misclassified as legitimate traffic, while this confusion is reduced to $\approx$ 24% in BiC. Conversely, on the $\mathcal{K}^{new}$, iCaRL+ achieves higher accuracy (+3%), but at the cost of lower precision, resulting in a reduced F1. In detail, BiC predicts attacks with a $\approx$76% accuracy and benign with a $\approx$ 96% accuracy and an overall F1 $\approx$71%, confirming the best also in the binary task.

Similarly, in the *cross-dataset* scenario (Fig. 5), iCaRL+ confirms its superiority also in bMD, providing a $\approx$ 100% accuracy and a $\approx$ 99% F1. While BiC performs well overall, it suffers from higher forgetting, misclassifying 8% of attacks as benign, while iCaRL+ eliminates this confusion almost totally.

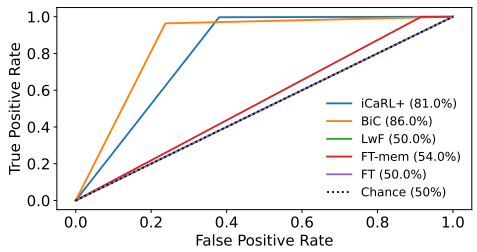
To provide a more nuanced evaluation, we analyze the performance of bMD from a different perspective by computing the ROC curves and the corresponding AUC values, as illustrated in Fig. 6. For reference, a "Chance" curve is included to represent the random guessing.

In the IoT-23[B+M] scenario, BiC confirms to be the best approach, providing a higher AUC ($\approx$86%) than all the others. Notably, FT and LwF show poor performance comparable to random guessing (i.e. $\approx$ 50% AUC) while FT-Mem delivers only slightly higher AUC ($\approx$54%).

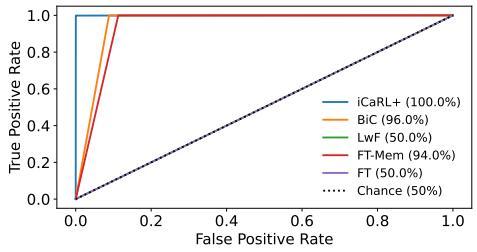
Similarly, in the *cross-dataset* scenario, both FT and LwF continue to perform poorly. In contrast, FT-Mem surprisingly achieves strong results, reaching a 94% AUC—close to the top performers. Notably, iCaRL+ is the best-performing method with a perfect 100% AUC, followed by BiC with a solid 96%.

VI. Conclusion

The increasing adoption of IoT devices has raised cybersecurity challenges, highlighting the urgent need for adaptable



(a) IoT-23[B+M]



(b) IoT-23[M]+Kitsune[B]

Fig. 6. ROC curve showing performance in binary misuse detection task. Values in brackets indicate the AUC of each CIL approach.

and privacy-preserving NIDS to guarantee security in increasingly heterogeneous and dynamic IoT environments.

This work presented a network-agnostic NIDS capable of generalizing across heterogeneous environments by leveraging known attack patterns, while enabling the integration of network-specific benign traffic in a privacy-preserving manner, without the need to exchange sensitive data. By incorporating CIL, the proposed NIDS updates its knowledge efficiently without requiring full retraining, thus reducing resource consumption and enabling continuous adaptation. Evaluations in both *intra-dataset* and *cross-dataset* scenarios show the robustness and practicality of the CIL solution, making it well-suited for real-world IoT deployments.

Our findings identified BiC as the most effective method in the *intra-dataset* scenario, exhibiting only 8% drop from the upperbound in multi-class misuse detection and 71% F1 in the binary task. Conversely, in the more challenging *cross-dataset* scenario, iCaRL+ delivered the best performance, with a minimal 4% gap from the upperbound in multi-class misuse detection and an outstanding 99% F1 in the binary task.

As *future directions*, we plan to: (i) explore more advanced CIL techniques to improve NIDS adaptability and multi-increment scenarios; (ii) deploy NIDS in a federated learning framework to better handle the heterogeneity of IoT devices and traffic patterns across various networks; (iii) evaluating the robustness of NIDS, particularly its resilience against adversarial attacks, such as poisoning.

REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): a vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[2] A. Nascita, F. Cerasuolo, D. Di Monda, J. T. A. Garcia, A. Montieri, and A. Pescape, "Machine and deep learning approaches for iot attack classification," in *IEEE INFOCOM WKSHPS*. IEEE, 2022, pp. 1–6.

[3] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Commun. Surveys Tuts.*, vol. 10, no. 4, pp. 56–76, 2009.

[4] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, and J. Van De Weijer, "Class-incremental learning: survey and performance evaluation on image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 5513–5533, 2022.

[5] Z. Song, Z. Zhao, F. Zhang, G. Xiong, G. Cheng, X. Zhao, S. Guo, and B. Chen, "I^2RNN: An Incremental and Interpretable Recurrent Neural Network for Encrypted Traffic Classification," *IEEE Trans. Depend. Sec. Comput.*, 2023.

[6] X. Xu, X. Zhang, Q. Zhang, Y. Wang, B. Adebisi, T. Ohtsuki, H. Sari, and G. Gui, "Advancing malware detection in network traffic with self-paced class incremental learning," *IEEE Internet Things J.*, 2024.

[7] X. Wang, S. Chen, and J. Su, "Real network traffic collection and deep learning for mobile app identification," *Wireless Communications and Mobile Computing*, vol. 2020, no. 1, p. 4707909, 2020.

[8] G. Bovenzi, A. Nascita, L. Yang, A. Finamore, G. Aceto, D. Ciuonzo, A. Pescapé, and D. Rossi, "Benchmarking class incremental learning in deep learning traffic classification," *IEEE Trans. Netw. Service Manag.*, 2023.

[9] L. Du, Z. Gu, Y. Wang, L. Wang, and Y. Jia, "A few-shot class-incremental learning method for network intrusion detection," *IEEE Trans. Netw. Service Manag.*, vol. 21, no. 2, pp. 2389–2401, 2023.

[10] Y. Wang and S. Cao, "A Two-Stage Class Incremental Learning Approach for Network Intrusion Detection," in *IEEE GLOBECOM*. IEEE, 2024, pp. 2353–2358.

[11] F. Cerasuolo, G. Bovenzi, D. Ciuonzo, and A. Pescapè, "Adaptable, incremental, and explainable network intrusion detection systems for internet of things," *Engineering Applications of Artificial Intelligence*, vol. 144, p. 110143, 2025.

[12] ——, "Attack-adaptive network intrusion detection systems for IoT networks through class incremental learning," *Comput. Netw.*, p. 111228, 2025.

[13] C. Oikonomou, I. Iliopoulos, D. Ioannidis, and D. Tzovaras, "A multi-class intrusion detection system based on continual learning," in *2023 IEEE CSR*. IEEE, 2023, pp. 86–91.

[14] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, 2017.

[15] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *Proceedings of the IEEE CVPR*, 2017, pp. 2001–2010.

[16] G. Bovenzi, L. Yang, A. Finamore, D. Ciuonzo, G. Aceto, A. Pescape, D. Rossi *et al.*, "A first look at class incremental learning in deep learning mobile traffic classification," in *5th Network Traffic Measurement and Analysis Conference, TMA 2021*, 2021.

[17] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu, "Large scale incremental learning," in *Proceedings of the IEEE/CVF CVPR*, 2019, pp. 374–382.

[18] M. Data and M. Aritsugi, "An incremental learning algorithm on imbalanced data for network intrusion detection systems," in *Proceedings of the 10th International Conference on Computer and Communications Management*, 2022, pp. 191–199.

[19] M. Kang, J. Park, and B. Han, "Class-incremental learning by knowledge distillation with adaptive feature consolidation," in *Proceedings of the IEEE/CVF CVPR*, 2022, pp. 16 071–16 080.

[20] S. Garcia, A. Parmisano, and M. J. Erquiaga, "IoT-23: A labeled dataset with malicious and benign IoT network traffic," Jan. 2020, 10.5281/zenodo.4743746.

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

[22] 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, 2019.

[23] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Network traffic classifier with convolutional and recurrent neural networks for internet of things," *IEEE Access*, vol. 5, pp. 18 042–18 050, 2017.

[24] L. Yang, A. Finamore, F. Jun, and D. Rossi, "Deep learning and zero-day traffic classification: Lessons learned from a commercial-grade dataset," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 4, pp. 4103–4118, 2021.