AdaptiveOnline-IDS: Efficient Intrusion Detection via Lifelong Learning and Knowledge Distillation

1st Nadia Niknami
Computing Science, Villanova University

2nd Jie Wu
Department of Computer and Information Sciences, Temple University

***Abstract*—Traditional Intrusion Detection Systems (IDSs) often struggle to adapt to the evolving nature of network threats and the changing distribution of data. This paper presents a novel intrusion detection framework that leverages state-of-the-art Lifelong Learning (LL) algorithms to continuously and effectively detect network anomalies while addressing the critical issue of catastrophic forgetting. The proposed approach begins by extracting network flows from raw Packet Capture (PCAP) files and organizing them into sequential tasks that represent different time periods or attack types. A Teacher-Student model is employed, in which knowledge distilled from the Teacher network guides the Student network through the initial learning phases. By utilizing advanced LL techniques, the framework ensures the preservation of previously learned information while maintaining adaptability to emerging threats. Experimental results demonstrate that our approach significantly improves detection accuracy over time and achieves high computational efficiency. This adaptive and scalable solution highlights the potential of combining flow-based analysis with LL strategies to enhance the resilience and effectiveness of modern intrusion detection systems.**

***Index Terms*—Intrusion detection system(IDS), Knowledge Distillation(KD), Lifelong Learning(LL), PCAP(Packet Capture)**

I. Introduction

Traditional Intrusion Detection Systems (IDS) often rely on static models and periodic updates, which cannot keep pace with the increasing complexity and frequency of cyberattacks. Because threats evolve dynamically, IDS must continuously update their models to recognize new attack patterns. In practice, intrusion data is collected incrementally, making it infeasible to retrain on all past and present data due to time and resource constraints. A major challenge in this setting is catastrophic forgetting [1], where a model loses the ability to detect previously learned attack patterns after being trained on new threats. This occurs because neural networks update their parameters to optimize for the latest data, often overwriting knowledge gained from earlier tasks. This underscores the need for incremental learning methods that reduce training overhead while preserving knowledge of past intrusions. Lifelong Learning (LL) [2] preserves prior knowledge while incorporating new data, allowing IDSs to respond dynamically to emerging threats [3]–[5].

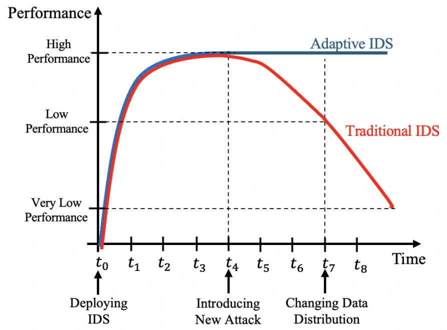


Fig. 1. Comparison of IDS adaptability under evolving attack scenarios.

As shown in Fig. 1, LL-based IDS (Adaptive IDS) outperforms batch-learning IDS (traditional IDS) in dynamic environments. At the start $(t_o - t_3)$, both systems reach high performance on the fixed data. After new attacks or drift appear (around $t_4 - t_7$), the traditional/batch IDS cannot adapt, so its accuracy falls over time. The LL IDS updates with incoming data, sustaining higher accuracy with fewer misses and stronger robustness to concept drift. Continuous adaptation is essential in modern networks where threats and behavior change. Although updates add compute and memory overhead, lightweight student models and regularized LL methods keep the cost practical.

To address this, Knowledge Distillation (KD) [6] can be integrated with LL to enhance efficiency. KD enables the transfer of knowledge from a complex, high-capacity model into a more compact one, significantly reducing computational overhead. The approach leverages probabilistic soft labels, which capture subtle class relationships, together with standard hard labels to guide training. This compression reduces model complexity while preserving critical decision boundaries, enabling IDSs to adapt to new threats efficiently with faster updates and lower resource demands [7] [8].

By combining LL with KD, IDSs can efficiently adapt to evolving threats in real time, even in complex-traffic or resource-constrained scenarios. This approach improves detection accuracy, reduces false positives, and ensures robust protection against both known and emerging cyber threats.

Recent studies have demonstrated the efficacy of KD and LL in IDS applications. Authors in [9] developed an IDS that combines GANs and KD to improve detection accuracy in network traffic. Authors in [10] proposed a lightweight IDS for industrial cyber-physical systems using KD with deep metric learning and a K-fold cross-training method for better generalization. Authors in [11] introduced a spatio-temporal KD framework for anomaly detection, leveraging CNNs and LSTMs to capture spatial and temporal features. Their approach uses Focal Loss to address class imbalance and

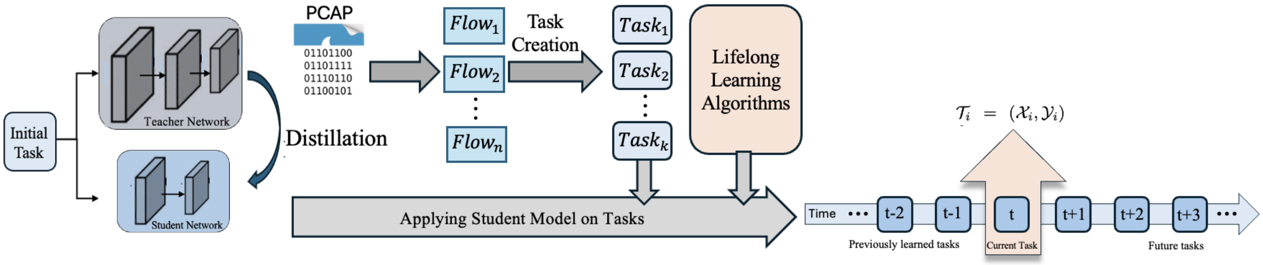


Fig. 2. Overview of the AdaptiveOnline-IDS framework based on knowledge distillation.

improves detection of rare cyberattacks on resource-limited devices. Authors in [12] applied lifelong-learning strategies to anomaly-based IDS (A-NIDS), showing they effectively curb catastrophic forgetting. Authors in [13] introduced a LL-based IDS (L-IDS) that fuses signature- and anomaly-based detectors under a lifelong-learning scheme, adapting to traffic changes and cutting false alarms while preserving accuracy. Authors in [14] proposed AOC-IDS which is an autonomous online system that uses contrastive objectives with pseudo-labeling to follow evolving distributions, improving detection without manual annotation. Authors in [15] proposed an online learning IDS that incrementally adapts to new IoT traffic using hybrid algorithms. Their approach combines margin-based online learning and stochastic weight averaging to enable adaptive, stable model updates under concept drift.

These approaches do not address real-world scenarios where intrusion detection must learn tasks sequentially and remain robust to order sensitivity. They also overlook the need for faster updates and efficient processing of evolving threats. Our work fills this gap by enabling IDS to adapt quickly while preserving the ability to detect previously learned attack patterns even after training on new threats.

The main contributions of this work are as follows:

- We proposed a novel framework that integrates Lifelong Learning algorithms with flow-based analysis to enable continuous and adaptive functionality in IDSs.
- We utilized various state-of-the-art Lifelong Learning algorithms to address the issue of catastrophic forgetting, ensuring that the deep learning model can effectively detect older attack patterns even after being trained on new or emerging threats.
- We develop a lightweight CNN for classifying PCAP files using a Knowledge Distillation framework. This ensures effective classification while reducing computational overhead.
- We conducted a comprehensive evaluation of various Lifelong Learning methodologies along with knowledge distillation on PCAP data. The results demonstrate how the proposed approach improves training efficiency and reduces time complexity in Lifelong Learning.

II. AdaptiveOnline-IDS: Proposed Method

Our goal is to run an IDS that continually learns and improves. We get help from LL algorithms, so the model updates as new traffic arrives, instead of waiting for offline retraining. To keep it fast, we built a small CNN trained by Knowledge Distillation (KD). Therefore, there is a two–stage design: (i) a KD pipeline that yields a compact learner suitable for online operation, and (ii) LL updates that mitigate forgetting as task distributions shift.

Fig. 2 presents the overview of the *AdaptiveOnline-IDS* framework based on knowledge distillation in detail. *AdaptiveOnline-IDS* consists of the following stages:

1) Teacher–student distillation
2) Task construction from PCAP flows
3) Sequential adaptation with LL

First, we train a high-capacity *Teacher* on PCAP-derived data from the initial task. Then, its knowledge is distilled into a lightweight *Student*. This transfers the teacher's decision structure to a much smaller network. The distilled Student keeps the Teacher's most useful decision signals but runs with much lower compute and memory, making it practical for real-time and resource-limited settings. After distillation, traffic from PCAP flows is organized into sequential tasks using complementary strategies. The tasks are built by different criteria such as time windows, attack types, and difficulty. Grouping flows into successive tasks matches how real networks evolve. This setup handles concept drift by letting the IDS adjust to new patterns while preserving past knowledge, and it encourages specialization on particular threats. Compared with a static IDS, the sequential approach better tracks changing environments, improving adaptability and long-term resilience. Therefore, the model faces a diverse set of challenges.

Tasks arrive in order along a timeline; at step t the Student adapts to the current task while reserving performance on $\{1, \ldots, t–1\}$. We apply a chosen LL mechanism such as EWC, SI, LwF, GEM, or PackNet to balance ability to learn new things and model stability by protecting important parameters or replaying saved samples. The timeline in Fig. 2 highlights the current task t relative to previously learned $(t-1, t-2)$ and upcoming $(t+1, t+2)$ tasks. It illustrates continual updates on the new task while keeping performance on earlier ones, showing how the system stays effective as conditions change. LL methods let the model learn new behavior without erasing what it already knows. They protect earlier decision patterns while the detector adapts to emerging attacks, reducing the chance of forgetting previously learned threats.

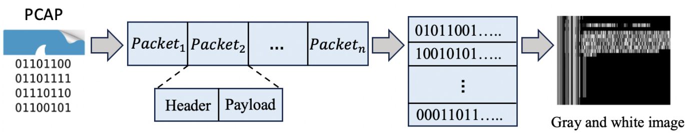


Fig. 3. PCAP-to-image conversion process.

As shown in Fig. 3, we turn raw PCAP traffic into image-like inputs. We parse each capture into its packets, reading both header fields (e.g., source/destination, ports) and payload. Each packet is then encoded as a 1-D vector of bits (0/1) representing its full content. For a given flow, packet vectors are ordered by time and stacked to form a 2-D matrix. This matrix is treated as a single-channel (grayscale) image. The resulting images reveal traffic structure in the case of regular patterns and anomalies that the model can analyze.

A. Problem Formulation

Let the traffic stream be segmented into PCAP-derived feature sets $\mathcal{D} = \{X_1, \dots, X_T\}$. From these we define a sequence of tasks $\mathcal{T} = \{\mathcal{T}_1, \dots, \mathcal{T}_K\}$, where each task $\mathcal{T}_i = (\mathcal{X}_i, \mathcal{Y}_i)$ contains flows $\mathcal{X}_i = \{F_{i1}, \dots, F_{in}\}$ and labels $\mathcal{Y}_i = \{Y_{i1}, \dots, Y_{in}\}$, drawn from a distribution $\mathcal{P}_i$ that may differ across i (temporal drift, new attack families, or complexity shifts). We seek a model $\mathcal{M}$ that, at time t, maps $\mathcal{X}_t \mapsto \mathcal{Y}_t$ while minimizing cumulative error on all tasks observed so far,

$$\min_{\mathcal{M}} \sum_{i=1}^{t} \mathcal{L}(\mathcal{M}(\mathcal{X}_i), \mathcal{Y}_i)\,,$$

subject to a stability–plasticity constraint: performance on $\{\mathcal{T}_1, \dots, \mathcal{T}_{t-1}\}$ should not deteriorate beyond a small tolerance while adapting to $\mathcal{T}_t$. KD provides the initial compact $\mathcal{M}$; LL updates enforce retention as the distributions $\{\mathcal{P}_i\}$ evolve over time.

To introduce complexity incrementally, training begins with straightforward, easily detectable attacks and progressively incorporates more sophisticated and subtle ones. This progression allows the model to establish a strong foundation in detection before confronting harder cases, supporting a smoother learning curve and better knowledge retention.

The Algorithm 1 describes the *AdaptiveOnline-IDS Framework*. With the help of KD, we process a sequence of tasks $\{(X_t, Y_t)\}_{t=1}^{T}$ to iteratively fine-tune the Student Model $\mathcal{S}$ for enhanced intrusion detection. Initially, a Teacher Network $\mathcal{T}$ is trained on the initial task data to capture comprehensive knowledge of the network flows. A Student Network $\mathcal{S}$ is then initialized and trained using KD, where knowledge from the Teacher Network is transferred to the Student Network. For each subsequent task, the algorithm generates task-specific data from PCAP flows and fine-tunes the Student Model $\mathcal{S}$ using LL techniques such as Elastic Weight Consolidation (EWC), Synaptic Intelligence (SI), Learning without Forgetting (LwF), and Gradient Episodic Memory (GEM). This fine-tuning process ensures that $\mathcal{S}$ adapts to new tasks while retaining knowledge from previously learned tasks. The final output of the algorithm is the fully trained and fine-tuned Student Model $\mathcal{S}_{\text{final}}$, capable of detecting both previously seen and new network threats.

Algorithm 1 AdaptiveOnline-IDS via Knowledge Distillation

```
Input $\{(X_t, Y_t)\}_{t=0}^{T}$: Data and Labels of Tasks
Output $\mathcal{S}_{\text{final}}$: Updated Student Model after all tasks
Train Teacher Network $\mathcal{T}$ with initial task data
Initialize Student Network $\mathcal{S}$
Step 1: Perform KD
Train $\mathcal{S}$ using KD from Teacher Network $\mathcal{T}$
for each task $t$ in $1, \dots, T$ do
    Step 2: Create Tasks from PCAP flows
    for each flow $f_i$ in task $t$ do
        Generate $(X_{t,i}, Y_{t,i})$ from flow $f_i$
    Step 3: Fine-tune $\mathcal{S}$ on task $t$ via LL
    Update $\mathcal{S}$ using LL methods
    Prevent forgetting of prior tasks via retention
    Step 4: Perform Knowledge Distillation
    Update $\mathcal{S}$ using distilled knowledge from $\mathcal{T}$
    Step 5:Teacher update (EMA)
    Update $\mathcal{T}$ by keeping a fraction of its old weights and
    mixing with $\mathcal{S}$.
```

III. Evaluation

We evaluate performance of *AdaptiveOnline-IDS* in several settings with the CICIDS2017 and CICIDS2018 datasets [16]. These datasets were not originally constructed with incremental learning in mind. To emulate that setting in our experiments, we partitioned the data into multiple batches and trained the models on them sequentially, thereby simulating the incremental learning process.

We evaluate multiple state-of-the-art LL methods that address catastrophic forgetting and preserve prior knowledge. The set covers complementary families—regularization-based, experience replay, and architecture-based techniques results reflect diverse strategies. The LL algorithms we consider in this work are Elastic Weight Consolidation (EWC) [1], Synaptic Intelligence (SI) [17], Learning without Forgetting (LwF) [18], Gradient Episodic Memory (GEM) [19], PackNet [20], Baseline Regular Fine-tuning, and Baseline Reservoir Sampling.

Selecting appropriate evaluation metrics is as critical as designing the experiments themselves. In addition to standard measures like accuracy, we adopt Relative Experience Forgetting (REF), which is particularly relevant in lifelong learning. REF quantifies how much a model forgets earlier tasks once it has been trained on new ones. For a given task t_i, REF expresses the relative drop in accuracy after subsequent training on tasks t_j:

$$\text{REF}(t_i, t_j) = (a_{t_i} - a_{t_i}^{t_j})/a_{t_i}, \tag{1}$$

where a_{t_i} denotes the accuracy on t_i immediately after it is learned, and $a_{t_i}^{t_j}$ represents the accuracy on t_i after additional

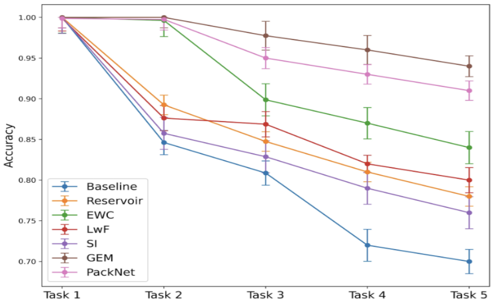


Fig. 4. Task-wise accuracy of LL algorithms in Scenario 1.

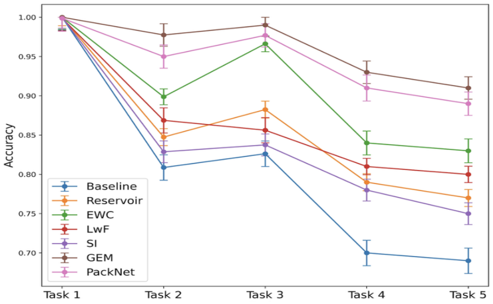


Fig. 5. Task-wise accuracy of LL algorithms in Scenario 2.

training on t_j. A larger REF value signals more severe forgetting, whereas a smaller value indicates stronger retention of prior knowledge. In addition, we assess the impact of Task Execution Order Sensitivity (TEOS). We create two setups differing only in task order:

- *Scenario 1:*
  DoS → DDoS → BruteForce → Bot → Infiltration
- *Scenario 2:*
  DoS → BruteForce → DDoS → Bot → Infiltration

This allows us to investigate how varying task order influences model performance.

All scripts, configurations, and detailed instructions required to replicate the experiments are made publicly available in the repository https://zenodo.org/records/14722831.

A. Analysis of Catastrophic Forgetting on Sequential Tasks

Fig. 4 and 5 illustrate how catastrophic forgetting impacts Task 1 accuracy as the model sequentially learns Tasks 2–5. All methods achieve near-perfect performance on Task 1, but accuracy steadily declines with each additional task. Among the approaches, GEM and PackNet show the greatest resilience, while Reservoir sampling, EWC, and LwF provide moderate protection, and the Baseline suffers the sharpest drop. By the time Task 5 is introduced, GEM and PackNet maintain the highest accuracy, confirming their robustness under cumulative learning. The comparison between scenarios also reveals order sensitivity: when BruteForce is introduced earlier (Scenario 1), it interferes more strongly with DoS detection than DDoS, likely due to weaker feature similarity. Even robust methods such as GEM and PackNet are affected, though they continue to outperform the alternatives. Scenario 2, with a less disruptive task sequence, achieves higher overall accuracy across all methods, underscoring the role of task similarity and order in lifelong learning. Overall, GEM and PackNet consistently emerge as the most effective strategies for mitigating catastrophic forgetting, whereas Reservoir, LwF, and the Baseline demonstrate clear limitations.

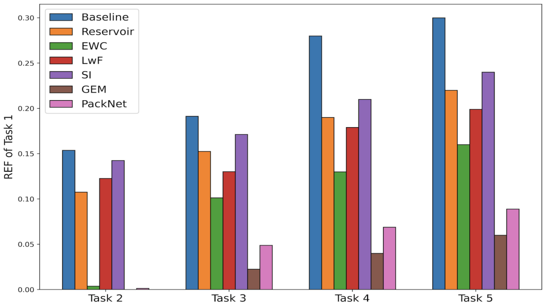


Fig. 6. REF across tasks for LL models in Scenario 1.

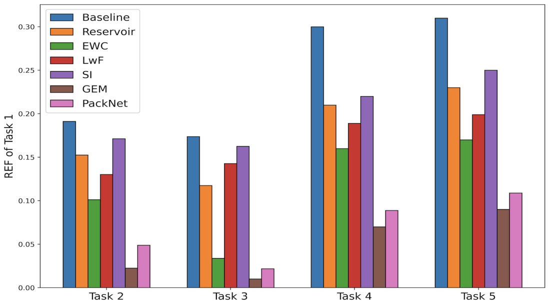


Fig. 7. REF across tasks for LL models in Scenario 2.

B. Analysis of Forgetting Using the REF Metric

Figs 6 and 7 compare lifelong learning methods using the REF metric under two different scenarios. Across both scenarios, the baseline approach records the highest REF values, indicating severe susceptibility to catastrophic forgetting. Reservoir sampling performs moderately better, offering some reduction in forgetting relative to the baseline. Among regularization-based techniques, EWC and SI provide stronger protection of earlier knowledge than LwF by constraining important parameters. Replay-based methods stand out, with GEM producing the lowest REF values and showing the greatest resilience, while PackNet also delivers strong results through its parameter isolation strategy, though slightly behind GEM. In Scenario 2, both Reservoir sampling and PackNet show improved REF values compared to Scenario 1, suggesting that their performance depends on the characteristics of the task sequence.

C. Evaluating Incorporation of KD

In this section, we examine the impact of incorporating KD into LL algorithms such as GEM. Table I compares a Teacher CNN (5-layer, without KD) against Student CNNs (trained with KD) across training time, memory usage, inference time, and average accuracy over tasks. The Teacher CNN achieves the highest accuracy but incurs substantial computational

TABLE I
PERFORMANCE COMPARISON OF TEACHER AND STUDENT KD MODELS ACROSS KEY METRICS.

Model	Training Time (s)	Memory Usage (MB)	Inference Time (s)	Accuracy
Teacher (5-layer CNN)	91.07	8.66	0.314	0.95
Student (4-layer CNN)	74.09	0.95	0.254	0.94
Student (3-layer CNN)	65.99	0.33	0.195	0.93
Student (2-layer CNN)	50.35	0.13	0.117	0.93

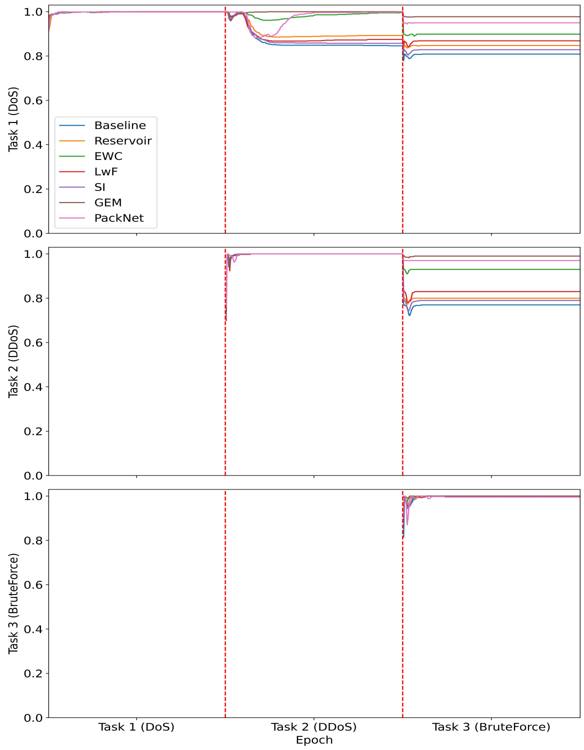


Fig. 8. Accuracy of LL algorithms across tasks (DoS → DDoS → Bruteforce). Red dashed lines mark task transitions.

costs, with the longest training time, largest memory footprint, and slowest inference. By contrast, Student CNNs offer a favorable trade-off between efficiency and accuracy. Training time, memory usage, and inference time all decrease as model depth is reduced, with the 2-layer CNN achieving the lowest resource requirements (50.35 s training, 0.13 MB memory, 0.117 s inference) while maintaining an accuracy of 0.93. Among the Student models, the 4-layer CNN strikes the best balance, reaching near-teacher accuracy (0.94) with moderate savings in time and memory. The 2-layer CNN is most suitable for resource-constrained environments, delivering high efficiency with only a slight accuracy reduction. Overall, these results highlight KD's effectiveness in enabling lightweight Student CNNs to retain strong performance at a fraction of the computational cost, thereby enhancing the practicality of LL systems in real-world deployments.

D. Evaluation of Models on Sequential Tasks

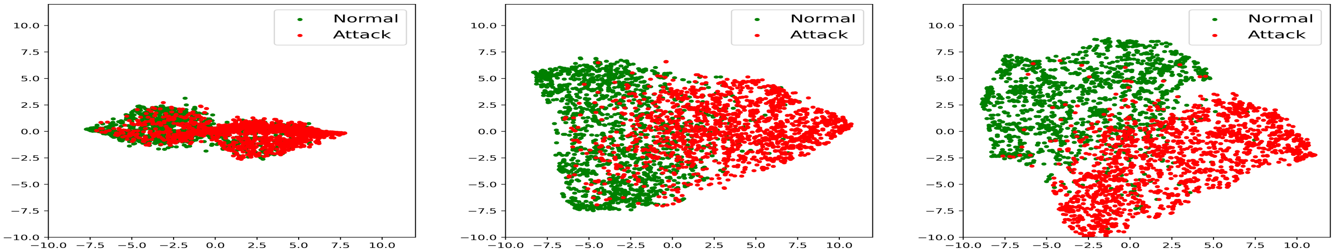
Fig. 8 illustrates how different lifelong learning algorithms behave when models are trained sequentially on three tasks (DoS, DDoS, and Brute Force). Along the horizontal axis are the training epochs for each task, while the vertical axis shows accuracy for the corresponding evaluation task. The first panel tracks performance on Task 1 (DoS). Accuracy is initially very high after training on Task 1 but drops once the model begins learning later tasks, reflecting catastrophic forgetting. Methods such as EWC, GEM, and PackNet lessen this decline, sustaining stronger accuracy. Because Task 1 and Task 2 share similarities, the drop after Task 2 training is smaller compared to the larger decline observed when Task 3 is introduced. The second panel reports results for Task 2 (DDoS). Accuracy is near perfect while the model is trained on this task but decreases when training shifts to Task 3. Here again, GEM and PackNet preserve performance better than other approaches. The third panel shows evaluation on Task 3 (Brute Force). Accuracy reaches its highest level once the model is trained on this task, consistent with the earlier plots where training and evaluation on the same task produce the best results.

E. t-SNE Visualizations

Figs 9 and 10 illustrate how the hidden representations of the GEM-trained network evolve through incremental learning. Each t-SNE visualization is generated from a test set containing samples from all tasks, with the plots reflecting the model's state after training on successive tasks. In Fig. 9(a), after Task 1, the embedding space is still mixed, with Normal and Attack points overlapping due to the model's limited exposure. By the time Task 2 is introduced, shown in Fig. 9(b), the structure improves, and separation between classes becomes clearer as the model integrates additional knowledge. After Task 3 training, Fig. 9(c) shows distinct clusters for Normal and Attack classes, indicating strong generalization across the test set. This progression demonstrates GEM's effectiveness in limiting catastrophic forgetting, enabling the network to retain early knowledge while adapting to new tasks. Fig. 10 illustrates how hidden layer representations evolve when the training order is changed, with Task 1 as DDoS, Task 2 as DoS, and Task 3 as Brute Force. The two plots differ only in task sequence, yet this variation alters how the model separates normal and attack classes. This highlights how task order affects learning and contributes to catastrophic forgetting.

IV. CONCLUSION

In this paper, we introduce *AdaptiveOnline-IDS*, a framework that integrates Lifelong Learning and Knowledge Distillation to improve intrusion detection. By structuring network flows into sequential tasks, the system adapts to evolving

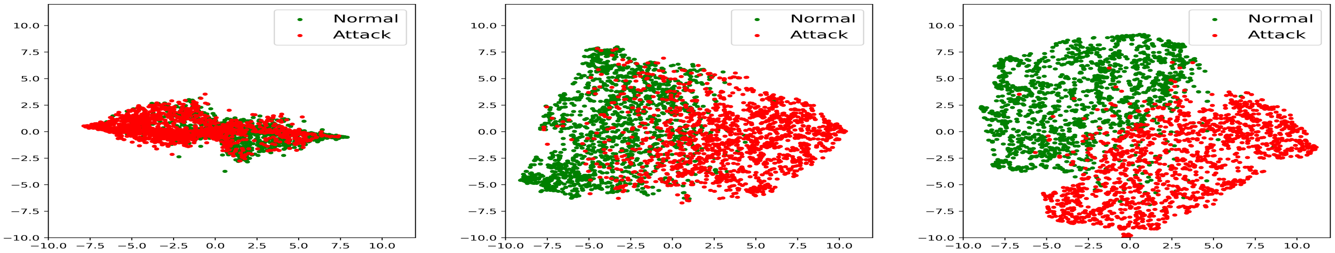


(a) After training on Task 1 (DDoS)

(b) After training on Task 2 (DoS)

(c) After training on Task 3 (Bruteforce)

Fig. 9. t-SNE of latent features after sequential training on DDoS → Bruteforce → DoS tasks.



(a) After training on Task 1 (DDoS)

(b) After training on Task 2 (Bruteforce)

(c) After training on Task 3 (DoS)

Fig. 10. t-SNE of latent features after sequential training on DDoS → DoS → Bruteforce tasks.

traffic patterns and threats while mitigating catastrophic forgetting. Using a Teacher–Student design, *AdaptiveOnline-IDS* achieves real-time efficiency, with lightweight Student models distilled from complex Teacher models delivering strong accuracy at reduced cost. Experimental results confirm that our approach retains critical knowledge across tasks, highlighting the potential of flow-based analysis and Lifelong Learning for building adaptive, resilient IDS solutions.

References

[1] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, “Overcoming catastrophic forgetting in neural networks,” *Proceedings of the national academy of sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[2] S. C. Hoi, D. Sahoo, J. Lu, and P. Zhao, “Online learning: A comprehensive survey,” *Neurocomputing*, vol. 459, pp. 249–289, 2021.

[3] A. A. Korba, S. Sebaa, M. Mabrouki, Y. Ghamri-Doudane, and K. Benatchba, “A life-long learning intrusion detection system for 6g-enabled iov,” in *2024 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 2024, pp. 1773–1778.

[4] R. Benameur, A. Dahane, S. Souihi, and A. Mellouk, “A robust and scalable federated continual learning framework for adaptive ddos detection in heterogeneous iot environments,” in *ICC 2025-IEEE International Conference on Communications*. IEEE, 2025, pp. 3063–3068.

[5] Y. Huang and M. Ma, “Aill-ids: An automatic incremental lifetime learning intrusion detection system for vehicular ad hoc networks,” *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[6] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv:1503.02531*, 2015.

[7] A.-N. Nadiah, A. Alamri, A. Aljuhani, and P. Kumar, “Transformer-based knowledge distillation for explainable intrusion detection system,” *Computers & Security*, vol. 154, p. 104417, 2025.

[8] T. Ali, A. Eleyan, T. Bejaoui, and M. Al-Khalidi, “Lightweight intrusion detection system with gan-based knowledge distillation,” in *2024 International Conference on Smart Applications, Communications and Networking (SmartNets)*, 2024, pp. 1–7.

[9] ——, “Lightweight intrusion detection system with gan-based knowledge distillation,” in *2024 International Conference on Smart Applications, Communications and Networking (SmartNets)*. IEEE, 2024, pp. 1–7.

[10] Z. Wang, Z. Li, D. He, and S. Chan, “A lightweight approach for network intrusion detection in industrial cyber-physical systems based on knowledge distillation and deep metric learning,” *Expert Systems with Applications*, vol. 206, p. 117671, 2022.

[11] X. Wang, Z. Wang, E. Wang, and Z. Sun, “Spatial-temporal knowledge distillation for lightweight network traffic anomaly detection,” *Computers & Security*, vol. 137, p. 103636, 2024.

[12] S. K. Amalapuram, A. Tadwai, R. Vinta, S. S. Channappayya, and B. R. Tamma, “Continual learning for anomaly based network intrusion detection,” in *14th International Conference on COMmunication Systems NETworkS (COMSNETS)*, 2022.

[13] H. Doroud, O. Alkhateeb, E. A. Jarchlo, and F. Dressler, “L-ids: A lifelong learning approach for intrusion detection,” in *International Wireless Communications and Mobile Computing (IWCMC)*, 2023, pp. 482–487.

[14] X. Zhang, R. Zhao, Z. Jiang, Z. Sun, Y. Ding, E. C. Ngai, and S.-H. Yang, “Aoc-ids: Autonomous online framework with contrastive learning for intrusion detection,” *arXiv preprint arXiv:2402.01807*, 2024.

[15] P. R. Agbedanu, S. J. Yang, R. Musabe, I. Gatare, and J. Rwigema, “Almanet: A hybrid online learning ids for real-time iot security,” *Egyptian Informatics Journal*, vol. 31, p. 100764, 2025.

[16] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani *et al.*, “Toward generating a new intrusion detection dataset and intrusion traffic characterization.” *ICISSp*, vol. 1, pp. 108–116, 2018.

[17] F. Zenke, B. Poole, and S. Ganguli, “Continual learning through synaptic intelligence,” in *International conference on machine learning*. PMLR, 2017, pp. 3987–3995.

[18] Z. Li and D. Hoiem, “Learning without forgetting,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[19] D. Lopez-Paz and M. Ranzato, “Gradient episodic memory for continual learning,” *Advances in neural information processing systems*, vol. 30, 2017.

[20] A. Mallya and S. Lazebnik, “Packnet: Adding multiple tasks to a single network by iterative pruning,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7765–7773.