Adapter-Based Few-Shot Continual Learning for Malicious Packet Recognition

Kyle Stein[1], Guillermo Francia, III[2], Eman El-Sheikh[2], Andrew Arash Mahyari[1,3],
[1] Department of Intelligent Systems and Robotics, University of West Florida, Pensacola, FL, USA
[2] Center for Cybersecurity, University of West Florida, Pensacola, FL, USA
[3] Florida Institute For Human and Machine Cognition (IHMC), Pensacola, FL, USA
ks209@students.uwf.edu, amahyari@ihmc.org, gfranciaiii@uwf.edu, eelsheikh@uwf.edu

***Abstract*—The continual evolution of malware variants necessitates detection systems that can adapt to new threats without retraining from scratch. However, continually updating models on new data often leads to catastrophic forgetting, where previously learned knowledge is overwritten. While continual learning has been increasingly explored for malware detection, the specific setting of Few-Shot Class-Incremental Learning (FSCIL), where new malware classes must be learned from only a small number of labeled examples, remains comparatively underexplored. Therefore, this work investigates the FSCIL setting for malware classification. To address the stability-plasticity dilemma, we propose a hybrid framework that leverages a Self-Supervised Learning (SSL) backbone initialized through domain-specific pretraining on malware packets. Our method incorporates Low-Rank Adaptation (LoRA) to efficiently adapt the model during the base session while freezing the core backbone to preserve previously learned representations, alongside a prototype-based classification head for incremental sessions to establish robust decision boundaries from limited samples. Extensive experiments across several datasets demonstrate that our approach consistently outperforms prior malware FSCIL baselines and achieves state-of-the-art performance.**

***Index Terms*—Few-shot class-incremental learning, parameter-efficient fine tuning, malware classification**

I. Introduction

Deep learning has transformed cybersecurity, enabling the detection and classification of malicious network traffic with unprecedented accuracy [1], [2], [3]. However, deployed malware classifiers face a fundamental challenge in the real world when the threat landscape continuously evolves, with new malware families emerging faster than large labeled datasets can be curated [4]. A classifier trained on known threats must possess the ability to rapidly adapt to novel attack patterns without full retraining of the model. In our previous work [5], we showed that a detector can adapt to previously unseen malware families in a static few-shot setting, where the set of novel target classes is fixed. In contrast, modern deployments face a sequential, evolving scenario in which new malware families may arrive in multiple sessions over time and the model must incorporate them while retaining performance on all previously learned classes, a setting that naturally results in catastrophic forgetting [6]. The field of continual learning, or class-incremental learning (CIL), formalizes this setting where a learner must adapt to new classes over time without forgetting previously learned ones [7], [8].

Traditional CIL scenarios consist of balanced sessions, where new classes are introduced over time, and where each new class has many samples each to learn from. Many CIL methods have been proposed to address the issue of forgetting, mainly in computer vision and image recognition tasks [8], [9], [10]. However, a more recent field of few-shot class incremental learning (FSCIL) [11] was formulated to address catastrophic forgetting in a new environment: where it is assumed that a model, trained on a robust base session of classes, must adapt to new classes with only few samples for each new class. In both of these scenarios, it is intuitive to rely on rehearsal [12], where the model replays previously seen samples while learning new classes to preserve earlier knowledge and stabilize the feature space of incremental sessions.

While rehearsal strategies have traditionally been effective at mitigating forgetting, they introduce significant deployment concerns. In many real-world scenarios, privacy regulations mandate that data be discarded immediately after processing. This constraint is particularly critical in malware analysis, where maintaining a persistent buffer of live malicious code poses a severe security risk, potentially leading to accidental execution or leakage [13], [14]. Consequently, a rehearsal-free framework is essential for secure continual learning, as it allows the system to discard malware samples immediately after model updates. However, the absence of a replay buffer typically hurts the stability and plasticity needed for continual learning models. Without past data to anchor the optimization, updating the model typically degrades the feature space and collapses previously learned decision boundaries [15].

In this work, we propose a multi-faceted approach to rehearsal-free FSCIL for malware detection through a modern, adapter-based strategy. Motivated by recent successes in leveraging pre-trained models for rehearsal-free CIL [15], [8], [16], we first pre-train a transformer from scratch on malware packets using a self-supervised objective [17]. We then adapt the pre-trained backbone to incremental sessions with low-rank adaptation (LoRA) [18] and perform classification using a prototype-based decision rule. Despite its simplicity, our approach outperforms previous continual learning methods for malware classification that rely on complex architectures. Finally, we evaluate under a fine-grained protocol where specific attack subtypes are treated as independent classes. This approach avoids simplifying the task through hierarchical grouping, thereby forcing the model to resolve high inter-class similarity and substantially increasing the difficulty of

incremental adaptation.

Overall, our contributions can be summarized as follows:

- We introduce domain-specific self-supervised pre-training on malware payloads with parameter-efficient LoRA updates on a frozen backbone and a prototype-based classifier to preserve decision boundaries over time.
- We propose the first framework to combine self-supervised pre-training from scratch with adapter-based few-shot continual learning for malicious packet recognition, enabling efficient adaptation to emerging malware classes without retaining previously observed samples.
- We provide an extensive empirical evaluation across multiple benchmark datasets, achieving state-of-the-art performance in few-shot continual malware classification.

II. Related Work

A. Few-Shot Class Incremental Learning and Rehearsal

FSCIL [11] extends CIL [15], [19] by adapting to novel classes with scarce labeled data. While earlier approaches relied on metric learning [20] or full fine-tuning [21], recent trends leverage frozen pre-trained transformers adapted through Parameter-Efficient Fine-Tuning (PEFT). Common prompt-based PEFT methods [22], [23] often face optimization difficulties [24]. To address this, we utilize an adapter-based strategy [10], [25], injecting lightweight modules into the frozen backbone to mitigate forgetting without the slow convergence associated with prompting.

Rehearsal, or experience replay [12], mitigates forgetting by storing and replaying past exemplars. While effective at maintaining decision boundaries, storing raw malware samples poses severe security and privacy risks, such as accidental execution or leakage [26], [27]. Consequently, the field is shifting toward rehearsal-free methods [23], [25] that rely on pre-trained models rather than stored data. In this work, we explicitly evaluate the performance trade-off between rehearsal-based and rehearsal-free architectures specifically within the malware domain.

B. Continual Learning for Malware Classification

An early finding in continual learning for malware classification is that many continual learning techniques, originally developed and benchmarked in computer vision, do not transfer cleanly to malware. Rahman *et al.* [28] evaluated a broad set of CL methods on malware datasets and found that many approaches fail to prevent catastrophic forgetting. This motivates malware-specific investigations into which continual learning assumptions are actually realistic for security deployments. FSCIL malware detection has also been explored in [29], where a base-stage model is trained with a replay-like variational autoencoder mechanism, and incremental sessions freeze the feature extractor while updating a prototype-based classifier. BFS-NID [30] mitigates forgetting for FSCIL by fixing the feature extractor parameters after the base session and utilizing a branch classifier learning module. In contrast to prior work, we propose the first malicious packet recognition framework to combine domain-specific self-supervised pre-training from scratch with adapter-based FSCIL, enabling rehearsal-free adaptation under scarce incremental data.

III. Preliminaries

A. FSCIL Problem Setup

FSCIL follows a similar sequential structure to CIL, but introduces severe data constraints for later tasks. Learning proceeds over a sequence of T tasks $\mathcal{T} = \{\mathcal{T}_0, \mathcal{T}_1, \ldots, \mathcal{T}_{T-1}\}$, where the first task $\mathcal{T}_0$ is denoted as the *base session*. In the base session, a comprehensive label space $\mathcal{Y}^{(0)}$ is provided along with ample data for each class, establishing a robust feature representation. However, for each subsequent incremental session $\mathcal{T}_t$ (where $1 \leq t < T$), only a few training samples per class are available. These incremental datasets $\mathcal{D}^{(t)}$ are arranged in an N-way K-shot format, where N new classes are introduced with only K labeled examples each. As in CIL, the class sets are disjoint: $\mathcal{Y}^{(i)} \cap \mathcal{Y}^{(j)} = \emptyset$ for any $i \neq j$. During inference after session t, the model is evaluated on a query test set $\mathcal{D}_{\text{test}}^{(\leq t)}$ containing samples from the cumulative label set $\mathcal{Y}^{(\leq t)} = \bigcup_{j=0}^{t} \mathcal{Y}^{(j)}$. The goal is to incorporate new classes from scarce examples without catastrophic forgetting of the previously learned classes.

IV. Proposed Method

In this section, we describe the architecture and training protocol of our rehearsal-free malicious packet recognition FSCIL framework. Our approach operates in three distinct stages: self-supervised pre-training of a byte-level transformer encoder to learn protocol semantics from scratch, PEFT on base classes using LoRA, and a rehearsal-free incremental prototype classification stage for novel malware families. The overall framework of our approach is shown in Fig. 1.

A. Transformer Encoder

Adapting pre-trained models to downstream tasks has gained significant traction due to the ability of large-scale models to be efficiently fine-tuned for specialized classification tasks [15], [31]. This is prevalent in computer vision, where transformers are routinely fine-tuned for continual object classification. Motivated by these insights, we propose self-supervised pre-training of a transformer backbone on raw malware packets to learn robust, domain-specific feature representations before the introduction of class labels, shown to be effective in our previous work [2]. We employ a transformer encoder [17] to process network packets as sequences of bytes. Unlike traditional Natural Language Processing (NLP) which operates on word tokens, our model treats each byte value (0-255) as a discrete token. To handle variable-length inputs and masking, we extend the byte token space size to $V = 258$, reserving indices 256 for padding and 257 for masking.

The backbone consists of an embedding layer, positional encodings, and a stack of multi-head self-attention blocks. Given an input packet sequence $\mathbf{X} = [x_1, \ldots, x_L]$, the embedding layer maps each byte to a vector of dimension $d_{model} = 768$. Standard positional encodings are added to preserve sequence

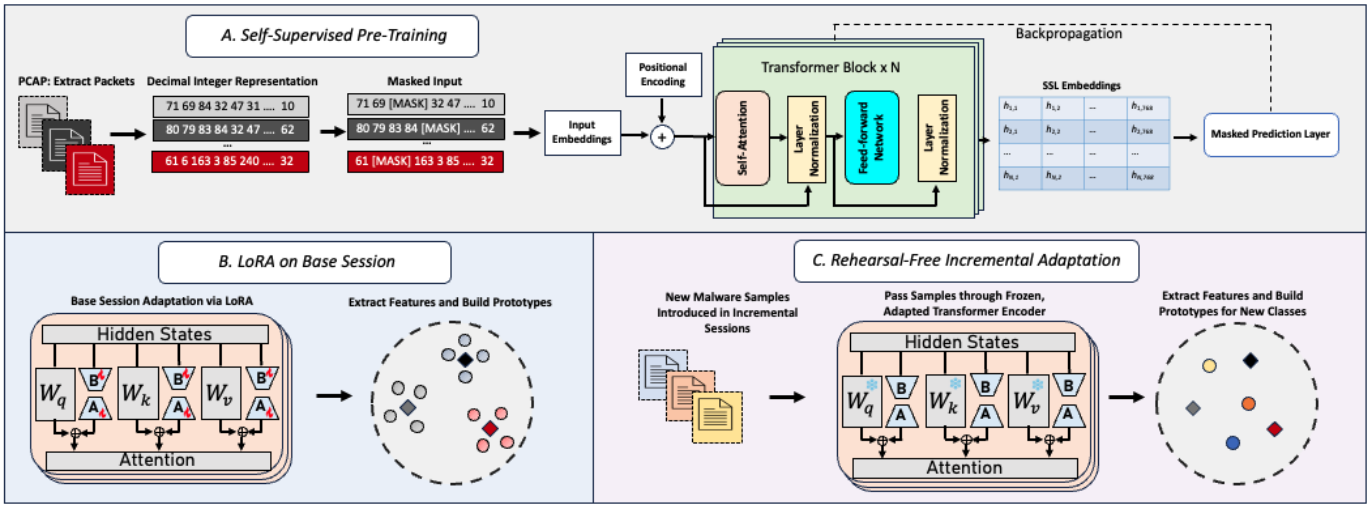

Fig. 1. The overall architecture of the proposed rehearsal-free FSCIL approach for malicious packet recognition.

order information. The self-attention mechanism computes contextual dependencies between bytes through:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}, \tag{1}$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ are the query, key, and value projections. We utilize a mean-pooling operation over the final hidden states to extract a fixed-size global representation vector $\mathbf{e} \in \mathbb{R}^{d_{model}}$ for each packet.

B. Self-Supervised Pre-training

To learn robust packet representations without explicit labeled classes, we train the backbone using a Masked Language Modeling (MLM) objective. Given an unlabelled packet sequence $\mathbf{x}$, we generate a binary mask $M \in \{0,1\}^L$ where each position has a 15% probability of being masked. This forces the model to reconstruct the original byte value x_i solely relying on the contextual information provided by the surrounding unmasked bytes.

The model projects the output hidden states of masked positions to the byte token space size V using a linear prediction head. We optimize the parameters θ by minimizing the cross-entropy loss between the predicted probabilities and the original byte indices:

$$\mathcal{L}_{MLM} = -\sum_{i \in \mathcal{M}} \log P(x_i | \tilde{\mathbf{x}}; \theta), \tag{2}$$

where $\mathcal{M}$ is the set of masked indices and $\tilde{\mathbf{x}}$ represents the masked input sequence. We utilize the AdamW optimizer with a learning rate of 1×10^{-4} and weight decay to mitigate overfitting during this phase. This yields a general-purpose feature extractor $f_\theta(\cdot)$ trained on malicious packet bytes. To adapt this extractor to specific malware families without destroying these general features, we proceed with a PEFT strategy with LoRA on the base session.

C. Base Session Adaptation with LoRA

Following pre-training, full fine-tuning on labeled data can lead to catastrophic forgetting of the pre-trained features due to overfitting to the few, newly introduced samples [16]. Therefore, it is important to develop a system that retains knowledge learned during the base session while remaining robust to the few malware samples introduced in later incremental sessions. However, this is a difficult task since we must learn from only K malware samples in incremental sessions due to our FSCIL constraints. To address this, we adopt an adapter-based approach through LoRA [18], which enables efficient adaptation of pre-trained models by freezing the weights learned during pre-training and injecting trainable low-rank matrices into the attention layers.

For a pre-trained weight matrix $\mathbf{W}_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, LoRA modifies the forward pass as:

$$\mathbf{y} = \mathbf{W}_0\mathbf{x} + \Delta\mathbf{W}\mathbf{x}, \tag{3}$$

where $\mathbf{x}$ and $\mathbf{y}$ denote the input and output activations, respectively. The update $\Delta\mathbf{W} = \mathbf{A}\mathbf{B}$ is decomposed into low-rank matrices $\mathbf{B} \in \mathbb{R}^{r \times d_{\text{in}}}$ and $\mathbf{A} \in \mathbb{R}^{d_{\text{out}} \times r}$, where $r \ll \min(d_{\text{in}}, d_{\text{out}})$ is a user-chosen rank controlling the capacity of the adaptation. In this paper, we set the rank to 8 for all experiments. By constraining the update to this low-rank structure, LoRA captures the most important task-specific changes while requiring far fewer trainable parameters than updating the full weight matrix. We apply LoRA to the query, key, and value projections of our transformer-based packet encoder.

D. Rehearsal-Free Incremental Learning

While LoRA enables efficient adaptation in the base session, continuing to update the model during incremental few-shot sessions can cause feature drift and degrade previously learned decision boundaries. However, a frozen feature space is viable in our setting because malware packet classes share underlying

structural semantics. Our prior work demonstrated that different malware classes share such semantics in their packets [2], [5]. By leveraging SSL during pre-training and LoRA during the base session, we establish a universal feature extractor that does not require further gradient updates to recognize new classes. As previously discussed, reliance on replay buffers would violate rehearsal-free constraints because it requires storing privacy-sensitive historical data. Therefore, we design a lightweight prototype construction phase that simply averages the available features of the newly introduced classes.

Prototype Formation: Formally, when K-shot support samples are provided for a new class c, we compute its prototype $\mathbf{p}_c$ by averaging the L_2-normalized feature vectors of the support set S_c, followed by a re-normalization step:

$$\mathbf{p}_c = \text{Normalize}\left(\frac{1}{|S_c|}\sum_{\mathbf{x}\in S_c}\frac{f_\theta(\mathbf{x})}{\|f_\theta(\mathbf{x})\|_2}\right) \tag{4}$$

where f_θ represents the frozen, pre-trained feature extractor and $\text{Normalize}(\mathbf{v}) = \mathbf{v}/\|\mathbf{v}\|_2$.

Inference: Unlike traditional Prototypical Networks [32] that utilize Euclidean distance, we employ a scaled cosine-similarity classifier to align with the normalized embedding space. Given the set of class prototypes $\{\mathbf{p}_c\}$ and a query embedding $\mathbf{z}_q = \text{Normalize}(f_\theta(\mathbf{x}_q))$, we compute the similarity scores:

$$s_c = \gamma \cdot (\mathbf{z}_q^\top \mathbf{p}_c) \tag{5}$$

where γ is a scaling factor and s_c represents the logit for class c. These logits are converted into class probabilities via the softmax function:

$$p(y = c \mid \mathbf{x}_q) = \frac{\exp(s_c)}{\sum_j \exp(s_j)}. \tag{6}$$

The predicted label for $\mathbf{x}_q$ is $\arg\max_c\, p(y = c \mid \mathbf{x}_q)$. Since $f_\theta(\cdot)$ remains frozen (no gradients are back-propagated) during our incremental testing scenario, we perform inference directly using these computed similarities.

E. Rehearsal-Based Incremental Learning

To provide a comparative baseline, we implement a rehearsal-based variant that relaxes standard privacy constraints to store a small buffer of exemplars. Unlike our proposed method, which relies on a static feature space, this variant continues parameter updates during incremental sessions using a combined dataset of the current few-shot samples and stored exemplars from previously seen classes. The replayed exemplar buffer is mixed with the new samples during training to constrain feature drift and reduce forgetting. After each session, we update the memory bank with exemplars from the newly introduced classes and refresh the prototype set by re-encoding the stored exemplars under the updated model to recompute class centroids. We argue that this strategy is suboptimal since it requires storing sensitive raw payloads, and the overfitting induced by repeated updates on limited data can degrade the generalizability of the SSL and base-session features.

TABLE I
EXPERIMENTAL SPLITS. BOTH BENCHMARKS (S_0 BASE, S_{1-3} INCREMENTAL) ARE EVALUATED UNDER IDENTICAL 5-SHOT CONSTRAINTS WITH 70 QUERY SAMPLES PER INCREMENTAL CLASS.

Sess.	Task Type	Samples (Train / Test)	Attack Class Composition	
			CIC-IDS2017	UNSW-NB15
S_0	Base (Many-Shot)	500 / 200	• DDoS • Infiltration • DoS Slowhttptest • FTP-Patator	• Generic • Analysis • Reconnaissance —
S_1	Inc. (5-Shot)	5 / 70	• SSH-Patator • Heartbleed • DoS Slowloris	• Shellcode • DoS —
S_2	Inc. (5-Shot)	5 / 70	• Web – Brute Force • Web – XSS • DoS GoldenEye	• Backdoor • Fuzzers —
S_3	Inc. (5-Shot)	5 / 70	• Bot • PortScan • DoS Hulk	• Exploits • Worms —

V. EXPERIMENTAL RESULTS

Datasets and Metrics. Our method is evaluated on two renowned malicious datasets, specifically CIC-IDS2017 [33] and UNSW-NB15 [34], utilizing the extracted datasets made available through Payload-byte [35]. Table I details the dataset splits across both benchmarks. Furthermore, SSL pre-training is performed using the Session 0 split, which is used to learn the pre-trained encoder initialization. Consistent with standard FSCIL protocols [11], we divide the training process into a robust base session (S_0) and sequential incremental sessions ($S_1, \ldots, S_N$). For the base session, we assume sufficient data availability and utilize 500 training samples per class to establish a discriminative initial feature space. For all subsequent incremental sessions, we enforce a N-way, K-Shot format, where the model must learn new attack patterns using only K support samples.

We evaluate on a cumulative test set containing query samples from all classes observed up to the current session. All results are averaged over three random seeds. We employ two standard metrics: **Accuracy** on the cumulative test set at the current session and **Average Forgetting** measures the decline in knowledge retention. For each previously learned class, we calculate the difference between *peak accuracy* (the highest accuracy ever achieved for the class in any prior session) and the *current accuracy*. We report the average of these drops across all old classes. All experiments are performed on a single NVIDIA H100 GPU.

A. Main Experimental Results

In this section, we discuss our main experimental results presented in Table II. On both CIC-IDS2017 and UNSW-NB15, our proposed rehearsal-free method significantly outperforms existing baselines. Against the SOA Rehearsal-Free baseline methods, our method achieves a final session (S_3) accuracy improvement of $+8.75\%$ on CIC-IDS2017 (61.05% vs 52.30%) and $+8.15\%$ on UNSW-NB15 (58.59% vs 50.44%). Furthermore, our method demonstrates superior stability, reducing the average forgetting rate by roughly half compared to MalFSCIL. This indicates that the combination of SSL pre-training and LoRA adapters establishes a far more robust

TABLE II
5-SHOT FSCIL PERFORMANCE ON UNSW-NB15 AND CIC-IDS2017.

Dataset	Strategy	Method	# Exemplars	S0	S1	S2	S3	FGT
CIC-IDS2017	No-Rehearsal	BFS-NID [30]	0	98.20±0.86	75.91±3.13	64.61±1.46	52.30±5.07	22.05±3.98
		MalFSCIL [29]	0	84.80±1.53	65.11±2.09	52.92±2.38	41.90±2.46	17.28±4.98
		Proposed	0	**99.92±0.70**	**83.79±1.69**	**71.01±2.87**	**61.05±2.47**	**8.59±1.54**
	Rehearsal	Proposed	1	99.88±0.10	76.77±3.08	58.30±5.09	53.23±6.58	29.20±8.19
		Proposed	5	**99.91±0.11**	**84.46±3.43**	**70.46±3.74**	**59.54±5.68**	**19.95±2.13**
UNSW-NB15	No-Rehearsal	BFS-NID [30]	0	83.66±0.83	74.19±1.04	60.50±2.62	50.44±6.81	13.67±5.48
		MalFSCIL [29]	0	75.94±12.59	51.22±16.55	23.18±6.77	17.01±4.79	34.91±4.63
		Proposed	0	**88.56±0.42**	**74.77±0.88**	**64.66±4.36**	**58.59±5.38**	**9.14±4.70**
	Rehearsal	Proposed	1	85.39±0.89	61.53±12.90	47.77±9.59	40.66±2.62	26.85±6.42
		Proposed	5	**88.89±1.15**	**74.73±2.92**	**53.67±2.32**	**50.01±3.27**	**17.76±1.33**

feature space than the variational autoencoder approach used in prior work.

A key observation in our results is the performance trade-off between our Rehearsal-Based (Buffer=5) and Rehearsal-Free (Buffer=0) variants. As shown in Table II, both methods achieve comparable global accuracy in the final session (e.g., 61.05% vs 59.54% on CIC-IDS2017). However, their underlying behavior differs fundamentally regarding average forgetting. The rehearsal-based variant exhibits significantly higher forgetting (19.95% on CIC-IDS2017) compared to the Rehearsal-Free approach (8.59%). This illustrates the classic stability-plasticity dilemma. The rehearsal-based method allows for plasticity (updating LoRA weights), which enables it to learn new incremental classes effectively, but this weight update causes a drift in the feature space, leading to the forgetting of base classes. However, our rehearsal-free method enforces strict stability by freezing the weights. While the incremental sesssions remain static, the robust SSL-initialized backbone ensures that the base classes are preserved with minimal degradation. The rehearsal-free method achieves superior or equivalent overall performance without the security risks associated with storing a replay buffer, validating our hypothesis that a static, robustly pre-trained feature space is preferable for malware FSCIL.

VI. LIMITATIONS

While this work advances the state of FSCIL for malware detection, it is important to discuss the limitations associated with this field. First, our approach relies on the self-supervised pre-training of a transformer backbone from scratch which incurs significant initial computational costs. Although improvements in GPUs and hardware makes this manageable, and the use of SSL eliminates the need for labeled data during this phase, the quality of the resulting model remains heavily dependent on the diversity of the unlabeled malware corpus used for pre-training. Next, our rehearsal-free strategy enforces a static feature space during incremental sessions to prevent catastrophic forgetting. While effective for stability, this design assumes that the pre-trained backbone and base-session adapters yield sufficiently discriminative features for all future malware variants. If novel attacks exhibit a significant domain shift relative to the pre-training distribution, the frozen encoder may limit plasticity compared to methods that continuously fine-tune the backbone. However, we empirically address this concern in our experiments by injecting diverse classes during incremental sessions, thereby demonstrating the robustness of our feature space to significant semantic shifts. Finally, the proposed framework is specifically optimized for dynamic threat landscapes where retraining is prohibitive. In static environments where the set of malware families changes infrequently, the complexity of a continual learning system may exceed that of simpler, static classifiers.

VII. CONCLUSION

In this paper, we proposed a rehearsal-free FSCIL framework for malware classification that leverages SSL pre-training and LoRA to address catastrophic forgetting. By selectively updating only the self-attention layers from the domain specific SSL objective, our method effectively integrates new classes from scarce samples while preserving robust base representations. Extensive experiments on CIC-IDS2017 and UNSW-NB15 demonstrate that our approach significantly outperforms previous baselines in stability, showing that a robust frozen feature space is preferable to data replay in security-sensitive domains. Our findings underscore the advantages of targeted parameter tuning for deploying adaptive intrusion detection systems where data retention is prohibited.

VIII. ACKNOWLEDGMENT

This work is partially supported by the UWF Argo Cyber Emerging Scholars (ACES) program funded by the National Science Foundation (NSF) CyberCorps® Scholarship for Service (SFS) award under grant number 1946442. Any opinions, findings, and conclusions or recommendations expressed in this document are those of the authors and do not necessarily reflect the views of the NSF.

REFERENCES

[1] M. J. De Lucia, P. E. Maxwell, N. D. Bastian, A. Swami, B. Jalaian, and N. Leslie, "Machine learning raw network traffic detection," in *Artificial intelligence and machine learning for multi-domain operations applications III*, vol. 11746. SPIE, 2021, pp. 185–194.

[2] K. Stein, G. Francia, E. El-Sheikh, and A. A. Mahyari, "Packet inspection transformer: A self-supervised journey to unseen malware detection with few samples," *IEEE Access*, vol. 13, pp. 196 336–196 354, 2025.

[3] M. Lotfollahi, M. Jafari Siavoshani, R. Shirali Hossein Zade, and M. Saberian, "Deep packet: A novel approach for encrypted traffic classification using deep learning," *Soft Computing*, vol. 24, no. 3, pp. 1999–2012, 2020.

[4] U.-e.-H. Tayyab, F. B. Khan, M. H. Durad, A. Khan, and Y. S. Lee, "A survey of the recent trends in deep learning based malware detection," *Journal of Cybersecurity and Privacy*, vol. 2, no. 4, pp. 800–829, 2022.

[5] K. Stein, A. A. Mahyari, G. Francia, and E. El-Sheikh, "Towards novel malicious packet recognition: A few-shot learning approach," in *MILCOM 2024-2024 IEEE Military Communications Conference (MILCOM)*. IEEE, 2024, pp. 847–852.

[6] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in cognitive sciences*, vol. 3, no. 4, pp. 128–135, 1999.

[7] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 7, pp. 3366–3385, 2021.

[8] D.-W. Zhou, H.-L. Sun, J. Ning, H.-J. Ye, and D.-C. Zhan, "Continual learning with pre-trained models: A survey," in *IJCAI*, 2024, pp. 8363–8371.

[9] J. He and F. Zhu, "Exemplar-free online continual learning," in *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2022, pp. 541–545.

[10] D.-W. Zhou, H.-L. Sun, H.-J. Ye, and D.-C. Zhan, "Expandable subspace ensemble for pre-trained model-based class-incremental learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 23 554–23 564.

[11] X. Tao, X. Hong, X. Chang, S. Dong, X. Wei, and Y. Gong, "Few-shot class-incremental learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 12 183–12 192.

[12] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 2001–2010.

[13] M. Guri, R. Puzis, K.-K. R. Choo, S. Rubinshtein, G. Kedma, and Y. Elovici, "Using malware for the greater good: Mitigating data leakage," *Journal of Network and Computer Applications*, vol. 145, p. 102405, 2019.

[14] O. Or-Meir, N. Nissim, Y. Elovici, and L. Rokach, "Dynamic malware analysis in the modern era—a state of the art survey," *ACM Computing Surveys (CSUR)*, vol. 52, no. 5, pp. 1–48, 2019.

[15] D.-W. Zhou, Q.-W. Wang, Z.-H. Qi, H.-J. Ye, D.-C. Zhan, and Z. Liu, "Class-incremental learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[16] K.-H. Park, K. Song, and G.-M. Park, "Pre-trained vision and language transformers are few-shot incremental learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 23 881–23 890.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the North American Chapter of the Association for Computational Linguistics*, 2019.

[18] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models." *ICLR*, vol. 1, no. 2, p. 3, 2022.

[19] W. Liu, X.-J. Wu, F. Zhu, M.-M. Yu, C. Wang, and C.-L. Liu, "Class incremental learning with self-supervised pre-training and prototype learning," *Pattern Recognition*, vol. 157, p. 110943, 2025.

[20] Y. Yang, H. Yuan, X. Li, Z. Lin, P. Torr, and D. Tao, "Neural collapse inspired feature-classifier alignment for few-shot class incremental learning," *arXiv preprint arXiv:2302.03004*, 2023.

[21] C. Peng, K. Zhao, T. Wang, M. Li, and B. C. Lovell, "Few-shot class-incremental learning from an open-set perspective," in *European Conference on Computer Vision*. Springer, 2022, pp. 382–397.

[22] Z. Wang, Z. Zhang, C.-Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J. Dy, and T. Pfister, "Learning to prompt for continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 139–149.

[23] J. S. Smith, L. Karlinsky, V. Gutta, P. Cascante-Bonilla, D. Kim, A. Arbelle, R. Panda, R. Feris, and Z. Kira, "Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 11 909–11 919.

[24] N. Ding, Y. Qin, G. Yang, F. Wei, Z. Yang, Y. Su, S. Hu, Y. Chen, C.-M. Chan, W. Chen *et al.*, "Parameter-efficient fine-tuning of large-scale pre-trained language models," *Nature Machine Intelligence*, vol. 5, no. 3, pp. 220–235, 2023.

[25] J. He, Z. Duan, and F. Zhu, "Cl-lora: Continual low-rank adaptation for rehearsal-free class-incremental learning," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 30 534–30 544.

[26] D. R. Thomas, S. Pastrana, A. Hutchings, R. Clayton, and A. R. Beresford, "Ethical issues in research using datasets of illicit origin," in *Proceedings of the 2017 Internet Measurement Conference*, 2017, pp. 445–462.

[27] M. Botes and G. Lenzini, "When cryptographic ransomware poses cyber threats: Ethical challenges and proposed safeguards for cybersecurity researchers," in *2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 2022, pp. 562–568.

[28] M. S. Rahman, S. Coull, and M. Wright, "On the limitations of continual learning for malware classificationon the limitations of continual learning for malware classification," in *Conference on Lifelong Learning Agents*. PMLR, 2022, pp. 564–582.

[29] Y. Chai, X. Chen, J. Qiu, L. Du, Y. Xiao, Q. Feng, S. Ji, and Z. Tian, "Malfscil: A few-shot class-incremental learning approach for malware detection," *IEEE Transactions on Information Forensics and Security*, 2024.

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

[31] Z. Han, C. Gao, J. Liu, J. Zhang, and S. Q. Zhang, "Parameter-efficient fine-tuning for large models: A comprehensive survey," *arXiv preprint arXiv:2403.14608*, 2024.

[32] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[33] I. Sharafaldin, A. H. Lashkari, A. A. Ghorbani *et al.*, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." *ICISSp*, vol. 1, no. 2018, pp. 108–116, 2018.

[34] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 military communications and information systems conference (MilCIS)*. IEEE, 2015, pp. 1–6.

[35] Y. A. Farrukh, I. Khan, S. Wali, D. Bierbrauer, J. A. Pavlik, and N. D. Bastian, "Payload-byte: A tool for extracting and labeling packet capture files of modern network intrusion detection datasets," in *2022 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT)*. IEEE, 2022, pp. 58–67.