Poster: Towards Continual Learning for Malware Classification

Mohammad Saidur Rahman
Rochester Institute of Technology
saidur.rahman@mail.rit.edu

Scott E. Coull
Mandiant
scott.coull@mandiant.com

Matthew Wright
Rochester Institute of Technology
matthew.wright@rit.edu

***Abstract*—Malware classification presents a unique challenge for continual learning (CL) due to the daily influx of new samples and the evolution of malware to exploit new vulnerabilities. Sequential training using CL techniques could substantially reduce training and storage overhead in the face of this massive scale. In this work, we study five CL techniques applied to two malware tasks, covering common incremental learning scenarios such as domain and class incremental learning (IL). In addition, we show our preliminary results of explored different replay based CL approaches equipped for malware classification. Our results indicate promise to study further in this direction.**

***Index Terms*—malware analysis; catastrophic forgetting; continual learning;**

I. Introduction

Machine learning (ML) models for malware classification are currently trained in a static way to learn previously observed samples with the expectation of generalizing to new observed samples. However, the adversarial nature of malware, coupled with the continuous evolution of benign software (*goodware*), results in an inherently non-stationary problem.

For example, VirusTotal receives more than 1 million unique software each day [1]. These accumulated daily feeds can easily result in a dataset of billions of samples after only a few years. The resultant dataset will not be independent and identically distributed (IID), as there will be distributional shifts. To adapt to the evolving data distribution over time, the model must undergo regular retraining to ensure its effectiveness. Regrettably, the rapid creation of new malware and goodware generates massive datasets, making them both resource-intensive to maintain and challenging to train on. Considering the practical challenges of training these models, antivirus vendors face the choice of: (i) eliminating some older samples from the training set, which may enable attackers to reuse older malware rather than creating new ones; (ii) reducing the training frequency, resulting in slower adaptation to changes in the distribution; or (iii) investing significant resources to consistently retrain using the entire dataset.

These challenges can be addressed with an ever-evolving and intelligent ML system that is continuously trained to adapt to changes in the data distribution without requiring significant storage and computational overhead. *Continual learning (CL)* is a branch of ML that aims to address those goals, enabling incremental incorporation of new data and adaptation to data distribution shifts without maintaining large datasets or incurring very high training overhead [2], [3]. The major challenge in training a model to learn continually is *catastrophic forgetting (CF)* – a phenomena in which an ML model forgets the previously learned knowledge [4].

In a CL training paradigm, the model $\mathcal{M}$ is presented with a sequence of tasks $t_1, t_2, ..., t_N$, where each task t_i is associated with a non IID data distribution $p(x, y|t_i)$. The corresponding parameters of these tasks can be represented as $\theta_{t_1}, \theta_{t_2}, ..., \theta_{t_N}$. The goal is to train $\mathcal{M}$ sequentially as the task appears so that it can adapt to the new task t_N while preserving its learning up to the previous task t_{N-1}.

In this work, we investigate the extent to which malware classification models suffer from CF, and whether we can address this using approaches from current CL research[1]. In addition, we present our preliminary findings of a malware data centric CL technique considering the properties of malware problem space.

II. Methodology and Results

A. CL Scenarios for Malware Classification

Using a large-scale benchmark malware dataset – EMBER 2018 [6], we show the investigation of two out of three widely adapted CL scenarios [7] – *domain incremental learning (Domain-IL)* and *class incremental learning (Class-IL)*, considering the problem spaces of binary malware classification and multi-class classification.

In Domain-IL, the model's objective is to classify a new test sample as malicious or benign. We partition our dataset into monthly tasks for this binary malware classification problem, reflecting the natural concept drift of both malware and goodware due to their evolving capabilities and the release of benign software. Our goal is to integrate new information in each monthly incremental learning iteration while preserving previous discriminative knowledge. In Class-IL, we examine malware *family* classification, assigning malware samples to specific families based on their code base, capabilities, and structure. For Class-IL, we incrementally include newly-discovered families at each learning episode, expanding the model's capabilities in response to the ever-evolving landscape of malware families. In this multi-class classification setting, the base model starts with a non-trivial number of classes, and new classes are added subsequently. Performance during

[1] This work includes materials from our previously published paper [5].

TABLE I

SUMMARY OF THE EXPERIMENTS. THE AVERAGE ACCURACY ($\overline{Acc}$) AND MINIMUM ACCURACY ($\widehat{Acc}$) FROM ALL THE TASKS IN EACH EXPERIMENT. RESULTS IN **BOLD** INDICATE ACCURACY VALUES CLOSER TO *Joint* THAN *None*. ERROR RANGE IS OMITTED FOR RESULTS WITH LESS THAN 1.0 STANDARD DEVIATION.

Approach	**Method**	**Domain-IL**		**Class-IL**	
		$\overline{Acc}$	$\widehat{Acc}$	$\overline{Acc}$	$\widehat{Acc}$
Baselines	None	93.1	91.3	26.6	09.2
	Joint	**95.9**	93.2	**87.7**	85±2.5
Studied Techniques	EWC	92.8	90.0	8.4	00.1
	LwF	93.2	91.7	11.9	00.7
	GR	93.2	91.6	26.9	09.3
	BI-R	93.4	91.6	26.7	9.0
	iCaRL	-	-	**62.8**	46±2.5
Ours	GRS-20%	94.4	93.8±1.1	**85.5**	82.2±1.8
	GRS-50%	**95.0±1.1**	94.2±1.2	**86.0**	82.7±2.2
	FRS-200/f	**94.9**	93.5±1.8	**85.4**	82.0±1.5
	FRS-500/f	**95.1**	94.0±1.2	**85.7**	81.8±2.7
	FRS-1000/f	**95.3**	93.9±1.6	-	-

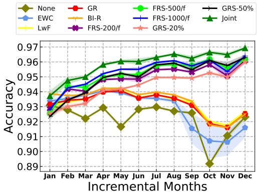

Fig. 1. **Domain-IL on EMBER**: Accuracy over time.

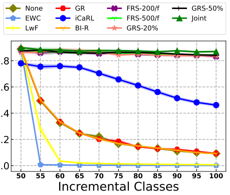

Fig. 2. **Class-IL on EMBER**: Accuracy as the number of classes grows.

testing is evaluated based on all classes the model has been trained on up to that point.

B. Model Selection, Implementation Details, and Baselines

In our experiments, we use a multi-layer perceptron (MLP) model that we designed using selective hyperparameter tuning, which achieves an AUC score of 0.995. For comparison, a LightGBM model by [6] reports a state-of-the-art AUC score of 0.996 on EMBER 2018.

For Domain-IL, we utilize goodware and malware samples from 2018, excluding unknown samples, and focus on binary classification across 12 months. In Class-IL experiments, we use the top 100 malware families seen in the dataset in 2018 (out of a total of 2,900 families), and train the initial base model with 50 classes.

We employ two baselines: *None* and *Joint*. *None*, which serves as an informal lower bound, involves sequential training of the model on new data without any CL techniques. *Joint*, on the other hand, trains on each task using *all* of the data accumulated so far. This serves as an informal upper bound. The effectiveness of a CL technique is measured by its ability to move accuracy from the lower bound to near the upper bound.

C. Experimental Results

In this work, we show the study of five widely studied CL techniques: Elastic Weight Consolidation (EWC) [8], Learning without Forgetting (LwF) [9], Generative Replay (GR) [10], Brain inspired replay (BI-R) [2], and Incremental Classifier and Representation Learning (iCaRL) [11].

In addition, we present our findings on two replay based CL techniques as part of this work – i) Global reservoir sampling (GRS), and ii) Family-based reservoir sampling (FRS). In replay-based CL, we replay samples of the earlier tasks based on a memory budget by including some of the older samples along with the new samples from the current task. GRS randomly selects X-% of all the stored samples of the earlier tasks and mixes those samples with the current samples to revive the stability of the model on the earlier tasks. FRS, on the other hand, randomly selects samples from each of the family data pool based on a given memory budget.

We show the results of Domain-IL and Class-IL experiments in Figure 1 and Figure 2, respectively. We show a summary of the results of all the experiments in Table I. For GRS, we only conduct experiments with replay rates of $x = 20\%$ and $x = 50\%$. For the experiments in FRS, we represent the configurations as FRS-n/f where $n \in \{200, 500, 1000\}$ for Domain-IL and $n \in \{200, 500\}$ for Class-IL.

To our surprise, our results indicate that none of our studied CL techniques are effective to reduce CF in Domain-IL. In Class-IL, iCaRL is the only performant method among the five studied techniques. Note that iCaRL is designed for only the Class-IL scenario, which is why there is no graph for iCaRL in Domain-IL. Our analysis in the previous work [5] reveals that the data distribution is much more complex in the malware data compared to the image data like MNIST for which the CL techniques are originally proposed. As such, these studied techniques suffer to reduce CF in malware data.

In Domain-IL, FRS-500/f and FRS-1000/f configurations perform very close to Joint performance with 95.1% and 95.2% average accuracy over all tasks, respectively. In Class-IL, GRS-50% yields 96% average accuracy which is performant among all the experiments. GRS-20%, FRS-200/f, and FRS-500/f also perform significantly better compared to all other techniques from prior work that we tested.

In summary, our simple replay-based CL techniques performed significantly better than existing CL techniques from the literature. This set of preliminary results warrant for a more in-depth investigation towards effective CL techniques that account for the unique properties of malware data.

Acknowledgement: This material is based upon work supported in part by the National Science Foundation under Grant No. 1816851.

REFERENCES

[1] VirusTotal, “VirusTotal - Stats,” https://www.virustotal.com/gui/stats.

[2] G. M. van de Ven, H. T. Siegelmann, and A. S. Tolias, “Brain-inspired replay for continual learning with artificial neural networks,” *Nature Communications*, 2020.

[3] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio, “Gradient based sample selection for online continual learning,” *NeurIPS*, 2019.

[4] M. McCloskey and N. J. Cohen, “Catastrophic interference in connectionist networks: The sequential learning problem,” in *Psychology of Learning and Motivation*, 1989.

[5] M. S. Rahman, S. E. Coull, and M. Wright, “On the Limitations of Continual Learning for Malware Classification,” in *CoLLAs*, 2022.

[6] H. S. Anderson and P. Roth, “EMBER: an open dataset for training static pe malware machine learning models,” *arXiv*, 2018.

[7] G. M. van de Ven, T. Tuytelaars, and A. S. Tolias, “Three types of incremental learning,” *Nature Machine Intelligence*, 2022.

[8] 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,” *PNAS*, 2017.

[9] Z. Li and D. Hoiem, “Learning without forgetting,” *TPAMI*, 2017.

[10] H. Shin, J. K. Lee, J. Kim, and J. Kim, “Continual learning with deep generative replay,” *NeurIPS*, 2017.

[11] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, “iCaRL: Incremental classifier and representation learning,” in *CVPR*, 2017.