Towards Continual Learning for Malware Analysis

Mohammad Saidur Rahman, Ph.D.

Assistant Professor
Department of Computer Science
The University of Texas at El Paso (UTEP)

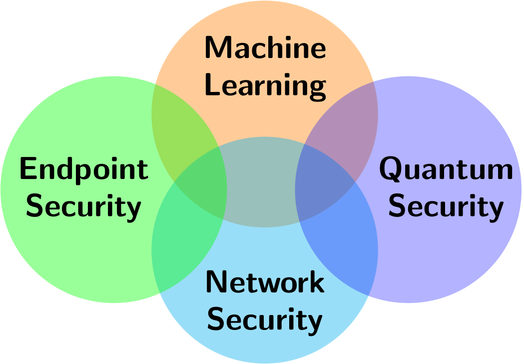
msrahman3@utep.edu
https://rahmanmsaidur.com/

Venue: Oklahoma State University (Virtual)

October 31, 2024

My Research

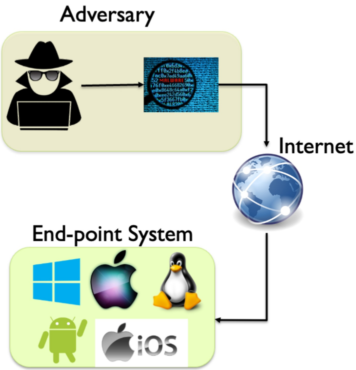
Intelligent and **Q**uantum **Se**cure Advanced **C**yber Defense Research (**IQSeC**) Lab
https://iqseclab.rahmanmsaidur.com/



Publications

- ACM CCS 2018, 2019
- PoPETS 2020, 2022
- IEEE TIFS 2020
- IEEE S&P 2022, 2023
- CoLLAs 2022
- WoRMA 2022
- IEEE QCNC 2024

Cybercrime and Malware



200 Million to 1.2 Billion in 10 years Growth

600%

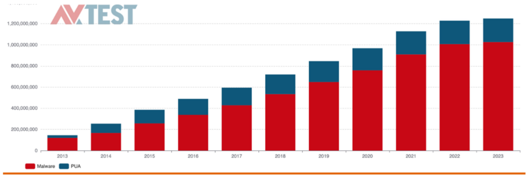
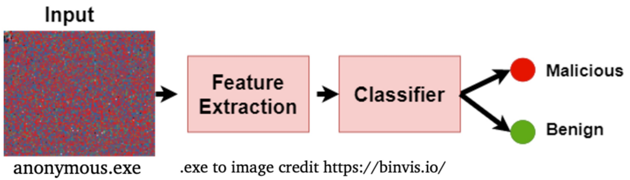


Figure: Growth of Malware and Potential Unwanted Applications (PUA)[1]

[1] https://www.av-test.org/en/statistics/malware/

Malware Analysis and Machine Learning

- Supervised Machine Learning (ML)
- Static malware analysis
  - Computational efficiency
  - Easy-to-Scale
  - Existing expert knowledge
- Significant performance
  - LightGBM on EMBER[2]
  - ROC AUC 0.996



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

Ever Evolving Growth of Malware

- AV-TEST $\Rightarrow$ 450K *new* malware and PUA *each day*[1]
- VirusTotal $\Rightarrow$ 1.8M *unique* software samples *each day*[3]

[1] https://www.av-test.org/en/statistics/malware/

[3] VirusTotal, https://www.virustotal.com/gui/intelligence-overview

Ever Evolving Growth of Malware

- AV-TEST ⇒ 450K *new* malware and PUA *each day*[1]
- VirusTotal ⇒ 1.8M *unique* software samples *each day*[3]

Huge data volumes drive up costs and training times

[1] https://www.av-test.org/en/statistics/malware/

[3] VirusTotal, https://www.virustotal.com/gui/intelligence-overview

Less than Ideal Solutions

Expanding Training Effort

expend tremendous effort to frequently retrain over all the data

Less than Ideal Solutions

Expanding Training Effort

expend tremendous effort to frequently retrain over all the data

Remove Older Samples

allows attackers to revive older malware instead of writing new ones

Figure: from [4]

[4] http://www.martybucella.com/E199.gif

Less than Ideal Solutions

Expanding Training Effort

expend tremendous effort to frequently retrain over all the data

Remove Older Samples

allows attackers to revive older malware instead of writing new ones

Expanding Training Effort

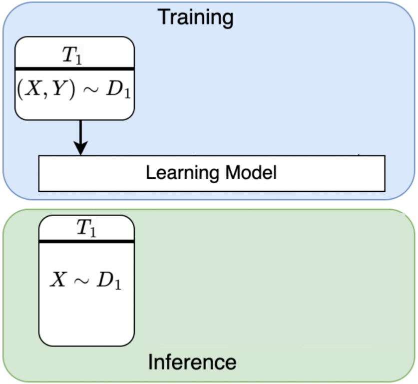
at the cost of not adjusting to changes in the distribution

Figure: from [4]

[4] http://www.martybucella.com/E199.gif

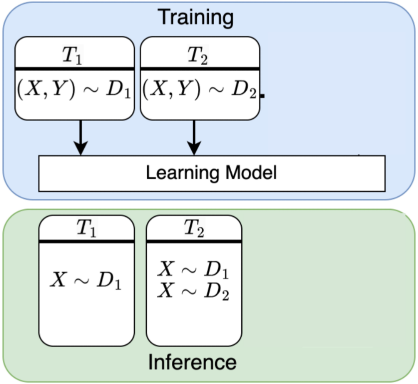
Continual Learning

- Acknowledges
  - Continuous distributional shift
- Non-stationary data
  - Observed periodically $(T_1, T_2, ..., T_N)$
  - Different data distribution in each period $(D_1, D_2, .., D_N)$
  - Data from each period is referred to as task
    - $task_N \in (T_N, D_N)$
    - New class/ new samples/ new objective



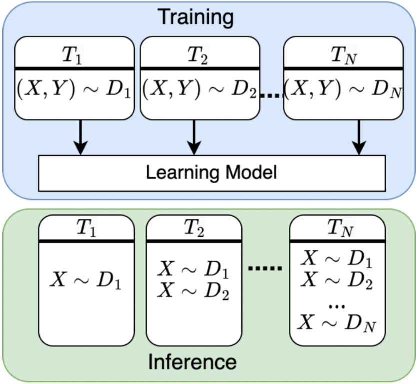
Continual Learning

- Acknowledges
  - Continuous distributional shift
- Non-stationary data
  - Observed periodically (T_1, T_2, ..., T_N)
  - Different data distribution in each period (D_1, D_2, .., D_N)
  - Data from each period is referred to as task
    - $task_N \in (T_N, D_N)$
    - New class/ new samples/ new objective



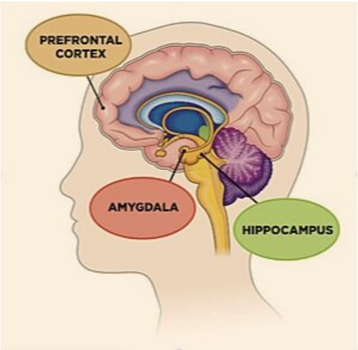
Continual Learning

- Acknowledges
  - Continuous distributional shift
- Non-stationary data
  - Observed periodically (T_1, T_2, ..., T_N)
  - Different data distribution in each period (D_1, D_2, .., D_N)
  - Data from each period is referred to as task
    - $task_N \in (T_N, D_N)$
    - New class/ new samples/ new objective



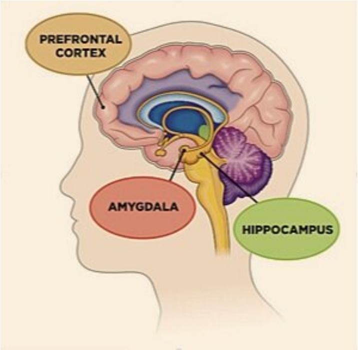
Continual Learning

- Inspired by human learning process
  - Continuous learning
  - Observe and learn
  - Storage → abstract representation in the hippocampus
- Relax the need to store all the data
  - Reduce storage cost
- Reduce computational cost



Continual Learning

- Inspired by human learning process
  - Continuous learning
  - Observe and learn
  - Storage $\rightarrow$ abstract representation in the hippocampus
- Relax the need to store all the data
  - Reduce storage cost
- Reduce computational cost



Challenge $\rightarrow$ Catastrophic Forgetting

Forgetting would reintroduce vulnerabilities

Catastrophic Forgetting (CF)

Neural Networks suffer from catastrophic forgetting[5]

- Forget the old tasks, unlikely to happen in human learning

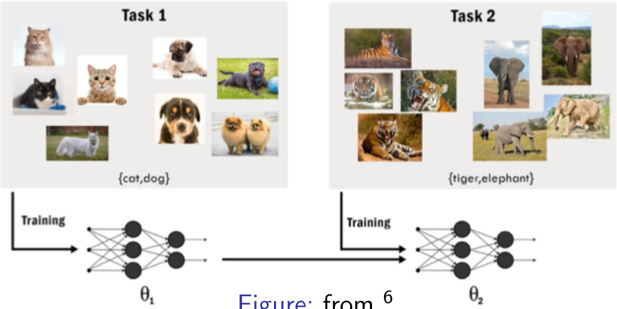


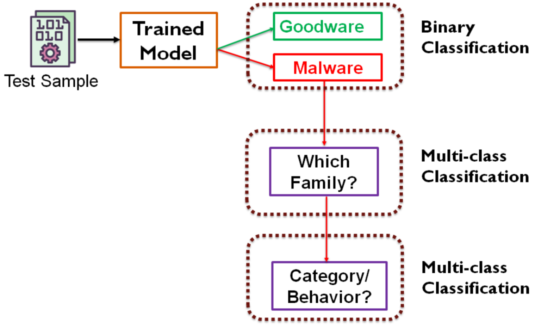
Figure: from [6]

[5] McCloskey and Cohen, Catastrophic interference in connectionist networks: The sequential learning problem, Psychology of learning and motivation, 1989.

[6] https://mrifkikurniawan.github.io/blog/2021/Catastrophic_Forgetting_in_Neural_Networks_Explained

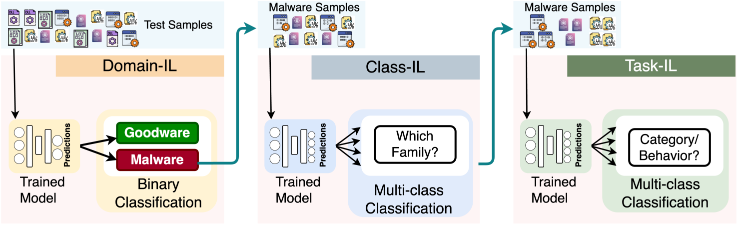
Malware Classification Pipeline

- Family
  - Citadel
  - Observe and learn
  - Gameover
  - Cthonic, and so on
- Category/Behavior
  - Adware
  - Ransomware
  - Banking Trojan
  - Backdoor, and so on

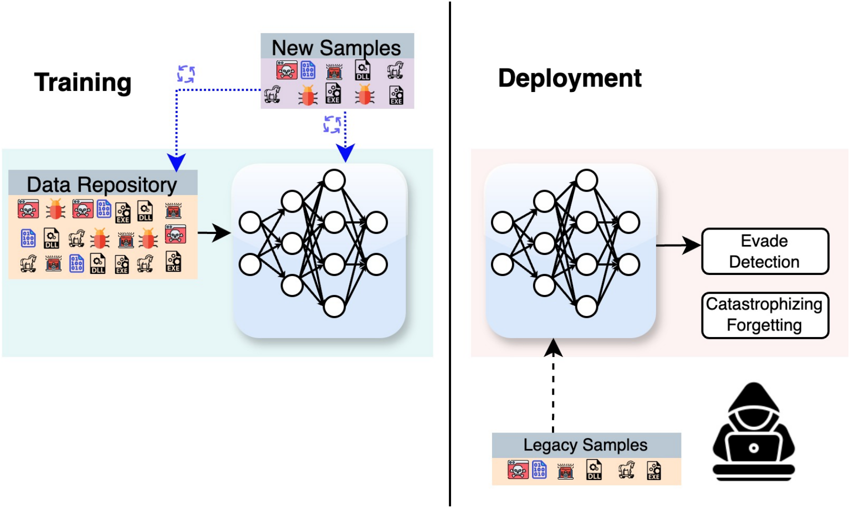


CL in Malware Classification Pipeline

- Domain Incremental Learning (**Domain-IL**)
  - Distribution shift
  - Emergence of new malware
- Class Incremental Learning (**Class-IL**)
  - New malware family
- Task Incremental Learning (**Task-IL**)
  - New malware category

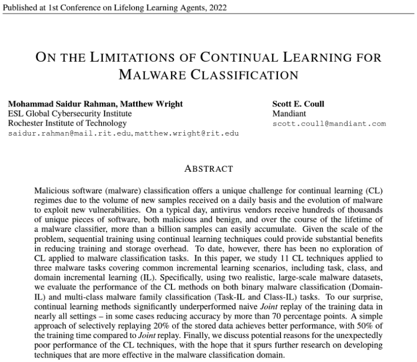


Threat Model



Adapted CL Techniques for Malware Classification

- Regularization
  - Elastic weight consolidation (EWC)
  - EWC Online (EWC-O), and
  - Synaptic Intelligence (SI)
- Replay
  - Learning without forgetting (LwF)
  - Generative replay (GR) and GR w/ Distillation
  - Replay through feedback (RtF)
  - Brain inspired replay (BI-R)
- Replay w/ Exemplars
  - Experience replay (ER)
  - Incremental classifier and representation learning (iCaRL)



Published at 1st Conference on Lifelong Learning Agents, 2022

ON THE LIMITATIONS OF CONTINUAL LEARNING FOR MALWARE CLASSIFICATION

Mohammad Saidur Rahman, Matthew Wright
ESL Global Cybersecurity Institute
Rochester Institute of Technology
saidur.rahman@mail.rit.edu, matthew.wright@rit.edu

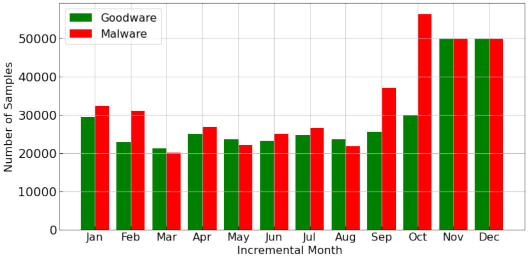
Scott E. Coull
Mandiant
scott.coull@mandiant.com

ABSTRACT

Malicious software (malware) classification offers a unique challenge for continual learning (CL) regimes due to the volume of new samples received on a daily basis and the evolution of malware to exploit new vulnerabilities. On a typical day, antivirus vendors receive hundreds of thousands of unique pieces of software, both malicious and benign, and over the course of the lifetime of a malware classifier, more than a billion samples can easily accumulate. Given the scale of the problem, sequential training using continual learning techniques could provide substantial benefits in reducing training and storage overhead. To date, however, there has been no exploration of CL applied to malware classification tasks. In this paper, we study 11 CL techniques applied to three malware tasks covering common incremental learning scenarios, including task, class, and domain incremental learning (IL). Specifically, using two realistic, large-scale malware datasets, we evaluate the performance of the CL methods on both binary malware classification (Domain-IL) and multi-class malware family classification (Task-IL and Class-IL) tasks. To our surprise, continual learning methods significantly underperformed naive *Joint* replay of the training data in nearly all settings – in some cases reducing accuracy by more than 70 percentage points. A simple approach of selectively replaying 20% of the stored data achieves better performance, with 50% of the training time compared to *Joint* replay. Finally, we discuss potential reasons for the unexpectedly poor performance of the CL techniques, with the hope that it spurs further research on developing techniques that are more effective in the malware classification domain.

EMBER Dataset

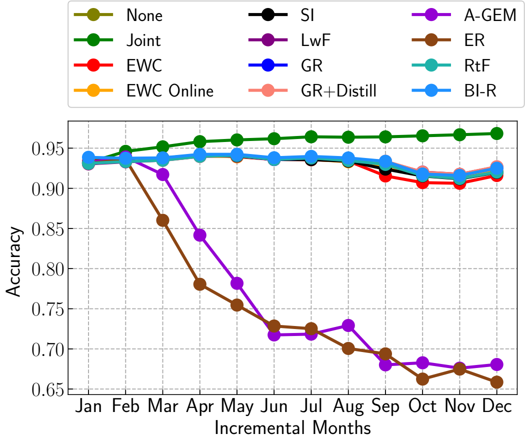
- EMBER (Windows Malware)[2]
  - Spans 12 months
    - Real-world data distribution shift
  - 400K goodware, 400K malware
  - Top 100 families
  - 2381 features



2 Anderson, Hyrum S., and Phil Roth. "EMBER: an open dataset for training static pe malware machine learning models." arXiv 2018.

Evaluation: EMBER Domain-IL

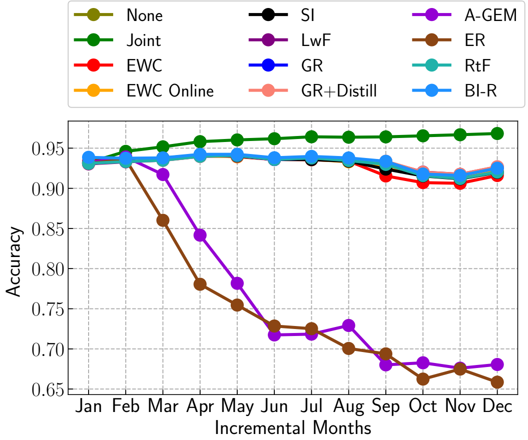
- Benchmarks
  - None → No CL techniques applied
  - Joint → Static training (training over accumulated data)



Evaluation: EMBER Domain-IL

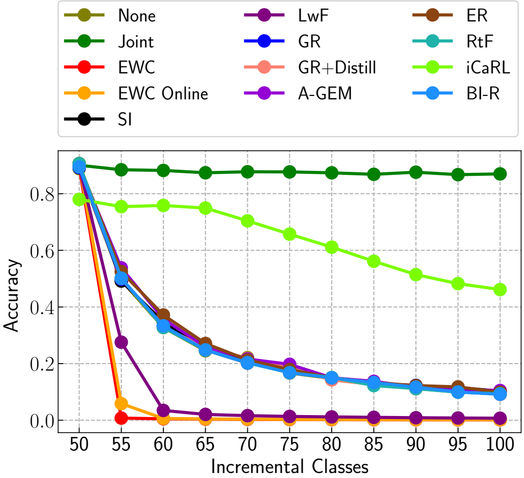
- Benchmarks
  - None → No CL techniques applied
  - Joint → Static training (training over accumulated data)

None of the CL techniques are effective in the Domain-IL setting



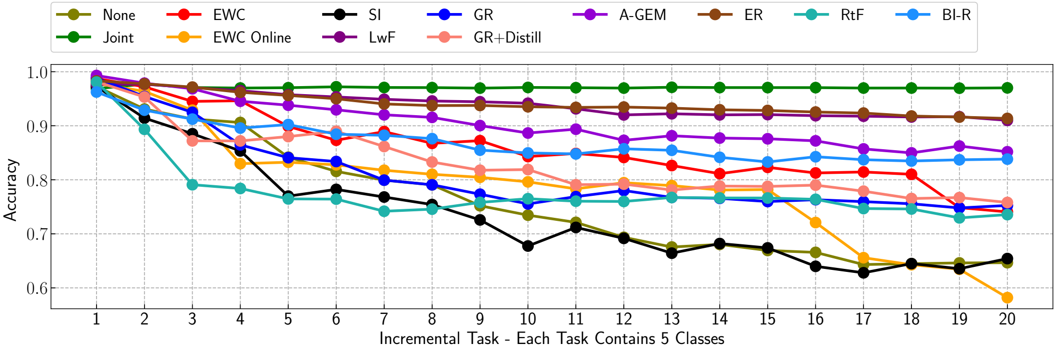
Evaluation: EMBER Class-IL

- 10 of the 11 methods performed poorly
- Only iCaRL performing marginally better against the Joint replay baseline



Evaluation: EMBER Task-IL

Several CL techniques work reasonably well on Task-IL



Overall Analysis

Unexpected Findings

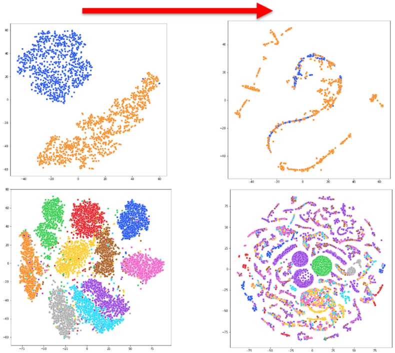
- None of the CL techniques are effective in the Domain-IL setting
- 10 out of 11 techniques are ineffective in the Class-IL setting

EMBER Dataset Complexity

Dataset complexity is significantly higher than image space, and feature space is more semantically-rich

MNIST 2 Class

EMBER 2 Class

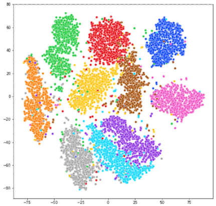


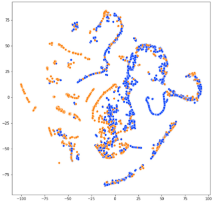
MNIST 10 Class

EMBER 10 Class

EMBER Real Domain Shifts

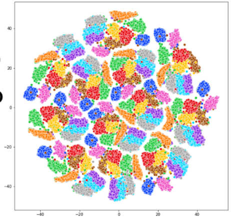
Original MNIST (no permutation)

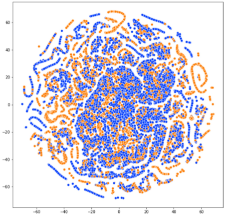




EMBER data of January

Cumulative MNIST data from Task 1 to Task 12 using permuted MNIST protocol.





Malware samples for each task

- Belong to multiple families
- Indicating sub-distributions within malware distribution

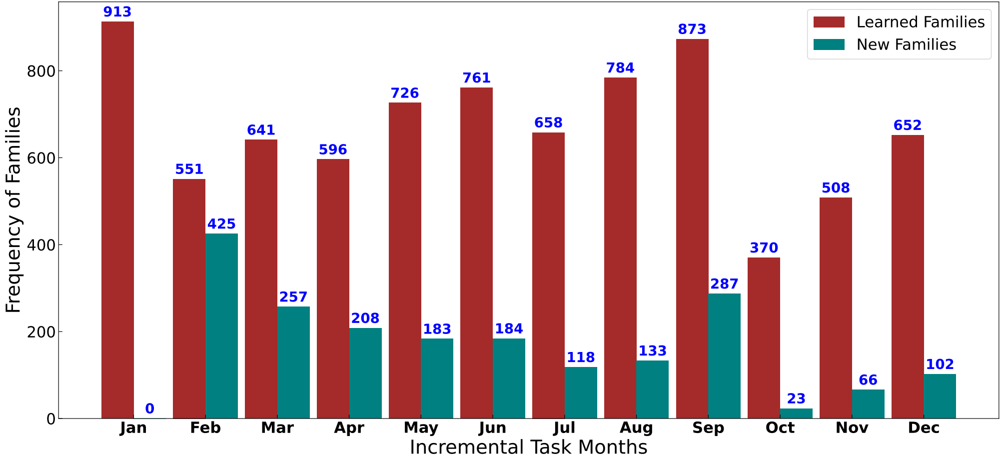
Task	#of Goodware	#of Malware	#of Malware Families
January	29423	32491	913
February	22915	31222	976
March	21373	20152	898
April	25190	26892	804
May	23719	22193	909
June	23285	25116	945
July	24799	26622	776
August	23634	21791	917
September	26707	37062	1160
October	29955	56459	393
November	50000	50000	574
December	50000	50000	754

Malware samples for each task

- Belong to multiple families
- Indicating sub-distributions within malware distribution

Task	#of Goodware	#of Malware	#of Malware Families
January	29423	32491	913
February	22915	31222	976
March	21373	20152	898
April	25190	26892	804
May	23719	22193	909
June	23285	25116	945
July	24799	26622	776
August	23634	21791	917
September	26707	37062	1160
October	29955	56459	393
November	50000	50000	574
December	50000	50000	754

Emergence of New Families in each Task



Summary of Exploratory Analysis

- Malware distribution in each task
  - Contains multiple sub-distributions
  - On an average around 800 families
- Lot of new novel families emerge
  - Old families observed infrequently
- Substantial #of samples wo/ AV class labels
- Priorities change over time
  - Prominent families do not remain prominent

MADAR: Malware Analysis with Diversity-Aware Replay

- CL technique should capture both representative and discriminative samples [7,8]
- Diversity among the replay samples
    - Family based sample selection
        - To accommodate varying families
    - Representative samples
        - Samples closer to the cluster mean
    - Discriminative (outlier) samples
        - Samples farther away from the mean

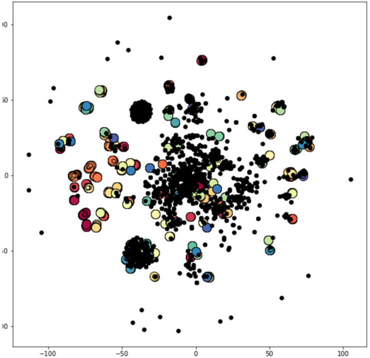


Figure: t-SNE projection of EMBER malware from January 2018

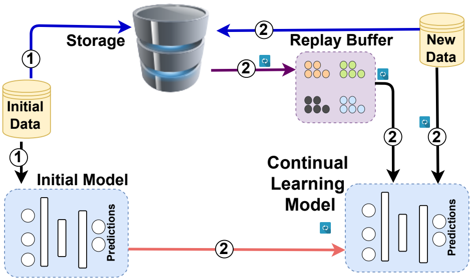
---

[7] Aljundi, Rahaf, et al. "Gradient based sample selection for online continual learning." NeurIPS 2019.

[8] Bang, Jihwan, et al. "Rainbow memory: Continual learning with a memory of diverse samples." CVPR 2021.

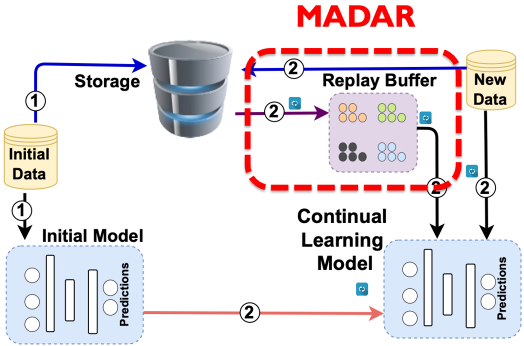
Replay-based CL for Malware Classification

1. Initial Phase
    - Initialize model w/ available data
    - Store the available data
2. CL Phase
    - Initialize model → CL Model
    - Replay some old data from the storage
    - Use (some/all) new data



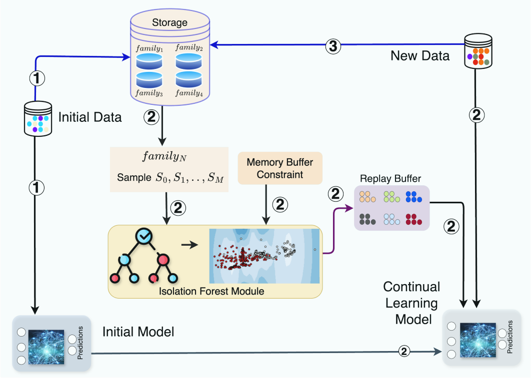
Replay-based CL for Malware Classification

1. Initial Phase
   - Initialize model w/ available data
   - Store the available data
2. CL Phase
   - Initialize model → CL Model
   - Replay some old data from the storage
   - Use (some/all) new data



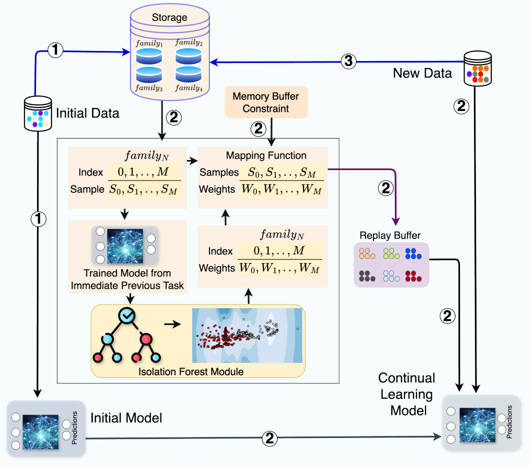
MADAR → Isolation Forest based Sampling (IFS)

1. Initial Phase
   - Initialize model w/ available data
   - Store the available data
2. CL Phase
   - Initialize model → CL Model
   - IFS Module
   - Replay Buffer



MADAR → Anomalous Weights based Sampling (AWS)

- Hidden representation
  - Weights of the model
  - Anomalous and Similar weights
    - Backtrack to raw feature space
  - Low dimension
    - Faster to process than raw feature space
    - i.e., 2381 → 256 (for EMBER)



Dataset – Android Malware w/ Drebin[9] Features

- AndroZoo Repository[10]
  - AZ-Domain for Domain-IL
    - Spans 9 years (2008 – 2016) $\rightarrow$ 80,690 malware and 677,756 goodware
    - 3,858,791 features $\rightarrow$ 1,789 (after variance thresholding)
    - 9:1 ratio of malware and goodware
    - Virus total detection count (c) >= 4
  - AZ-Class for Class-IL
    - 285,582 malware samples w/ VTDC >= 4
    - Top 100 families with at least 200 samples each
    - 1,067,550 features $\rightarrow$ 2,439 (after variance thresholding)

[9] Arp, Daniel, et al. "Drebin: Effective and explainable detection of android malware in your pocket." NDSS 2014.

[10] Allix, Kevin, et al. "Androzoo: Collecting millions of android apps for the research community." MSR 2016.

Evaluation → MADAR-IFS in Domain-IL

Group	Method	EMBER Budget 1K	EMBER 100K	EMBER 200K	EMBER 400K	AZ Budget 1K	AZ 100K	AZ 200K	AZ 400K
Baselines	Joint	96.4±0.3				97.3±0.1			
	None	93.1±0.1				94.4±0.1			
Prior Work	ER	80.6±0.1	69.9±0.1	70.0±0.1	70.0±0.1	40.4±0.1	42.6±0.1	44.0±0.1	48.6±1.1
	AGEM	80.5±0.1	70.0±0.1	70.0±0.2	70.0±0.1	45.4±0.1	53.7±0.6	54.2±0.3	56.7±0.3
	GR	93.1±0.2				93.3±0.4			
	RtF	93.2±0.2				93.4±0.2			
	BI-R	93.4±0.1				93.5±0.1			
	GRS	**93.6±0.3**	**95.3±0.7**	**95.9±0.1**	**96.0±0.3**	95.3±0.1	**97.1±0.1**	**97.1±0.1**	**97.2±0.1**
Ours	MADAR-R	**93.7±0.1**	**95.3±0.6**	**96.0±0.1**	**96.1±0.1**	**95.8±0.1**	**97.0±0.1**	**97.0±0.1**	97.0±0.1
	MADAR-U	**93.6±0.2**	**95.3±0.1**	95.5±0.1	95.8±0.1	**95.7±0.1**	95.2±0.1	95.4±0.1	96.3±0.2

Evaluation → MADAR-IFS in Class-IL

Group	**Method**	**EMBER** Budget 100	1K	10K	20K	**AZ** Budget 100	1K	10K	20K
Baselines	Joint	86.5±0.4				94.2±0.1			
	None	26.5±0.2				26.4±0.2			
Prior Work	TAMiL	32.2±0.3	35.3±0.2	38.2±0.3	38.8±0.2	53.4±0.3	57.6±0.3	63.5±0.1	67.7±0.3
	iCaRL	53.9±0.7	60.0±1.0	64.6±0.8	66.8±1.1	43.6±1.2	61.7±0.7	81.5±0.6	84.6±0.5
	ER	27.5±0.1	28.0±0.1	28.0±0.1	28.2±0.1	50.8±0.7	58.9±0.2	62.9±0.7	64.2±0.4
	AGEM	27.3±0.1	27.7±0.1	28.2±0.1	28.2±0.1	27.3±0.7	27.1±0.3	28.2±1.0	28.0±0.8
	GR	26.8±0.2				22.7±0.3			
	RtF	26.5±0.1				22.9±0.3			
	BI-R	26.9±0.1				23.4±0.2			
	GRS	51.9±0.4	75.4±0.7	83.5±0.1	84.6±0.2	43.8±0.7	70.2±0.4	86.4±0.2	89.1±0.2
Ours	MADAR-R	**68.0±0.4**	76.0±0.3	83.2±0.2	84.0±0.2	**59.4±0.6**	71.9±0.5	86.3±0.1	89.1±0.1
	MADAR-U	66.4±0.4	**79.4±0.4**	**84.8±0.1**	**85.8±0.3**	57.3±0.5	**76.2±0.2**	**89.8±0.1**	**91.5±0.1**

Evaluation → MADAR-IFS in Task-IL

Group	Method	EMBER Budget 100	EMBER Budget 1K	EMBER Budget 10K	EMBER Budget 20K	AZ Budget 100	AZ Budget 1K	AZ Budget 10K	AZ Budget 20K
Baselines	Joint	97.0±0.3				98.8±0.2			
	None	74.6±0.7				74.5±0.2			
Prior Work	TAMiL	72.8±0.1	86.9±0.2	90.3±0.1	94.2±0.7	80.5±0.4	91.5±0.2	93.5±0.1	94.8±0.2
	ER	67.4±0.3	89.5±0.5	94.8±0.2	95.4±0.1	83.6±0.2	92.3±0.3	96.2±0.1	97.5±0.2
	AGEM	79.6±0.2	83.8±0.4	86.1±0.2	89.3±0.1	76.7±0.5	85.3±0.1	86.7±0.2	91.3±0.3
	GR	79.8±0.3				75.6±0.2			
	RtF	77.8±0.2				74.2±0.3			
	BI-R	87.2±0.3				85.4±0.2			
	GRS	86.9±0.3	**93.6±0.3**	94.7±0.3	95.0±0.1	85.2±0.1	90.8±0.1	93.5±0.1	95.2±0.1
Ours	MADAR-R	92.1±0.2	93.8±0.2	94.8±0.2	**95.6±0.1**	86.0±0.3	92.4±0.1	96.7±0.1	97.9±0.2
	MADAR-U	**93.4±0.2**	**93.9±0.3**	**95.6±0.1**	**95.8±0.2**	**88.1±0.3**	**94.5±0.3**	**98.1±0.1**	**98.7±0.1**

Summary of the Findings

- Prior CL techniques → do not work well for malware tasks
  - Due to the complexity of the data and unique non-stationary nature
- Malware distribution represents diversity among and within families
- MADAR: Diversity Aware Replay Technique
  - State-of-the-art performance
  - Domain-IL → Ratio variants (MADAR-R and MADAR-AWS-R)
  - Class-IL → Uniform variants (MADAR-U and MADAR-AWS-U)

Takeaways

1. Evolving growth of malware is a challenging problem
    - Require an ever evolving and intelligent system for effective malware classification and detection
2. Continual Learning (CL) is an ideal candidate
    - CV based CL systems fall short to mitigate catastrophic forgetting in malware domain
3. CL for malware domain $\rightarrow$
    - Must consider the diverse nature and complexities of malware data distribution
    - Lots of open research questions
4. MADAR achieves state-of-the-art performance in several configurations

Thank You
Question?