PBP: Post-training Backdoor Purification for Malware Classifiers

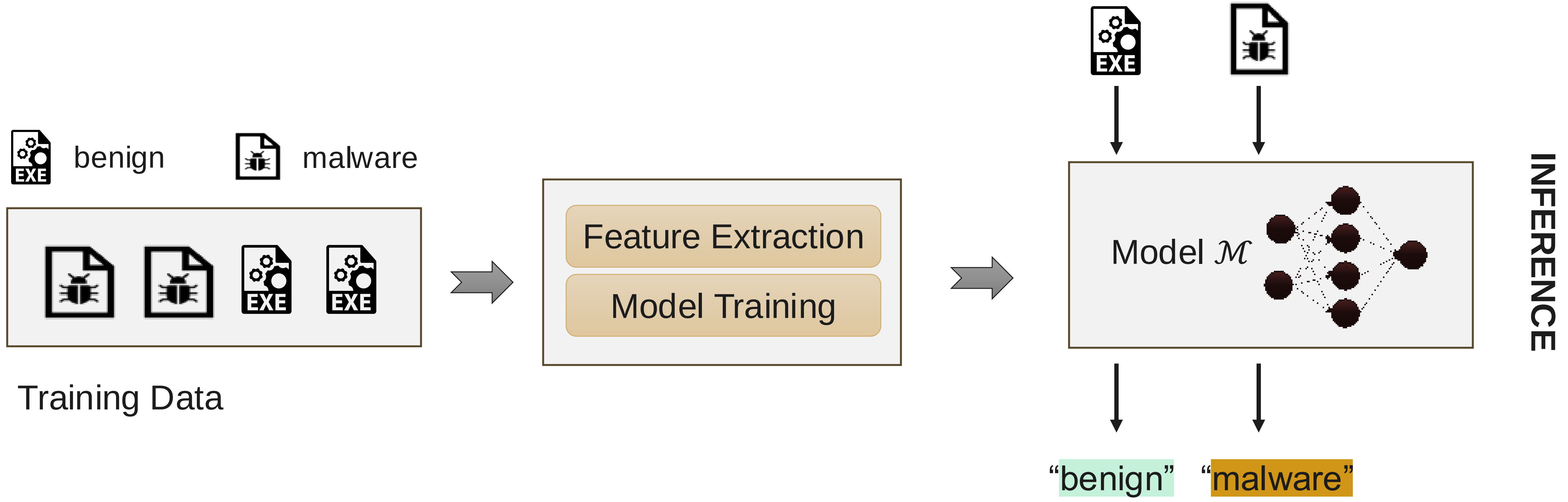
Dung (Judy) Nguyen, Ngoc N. Tran, Taylor T. Johnson, Kevin Leach

Machine Learning for Malware Classifiers

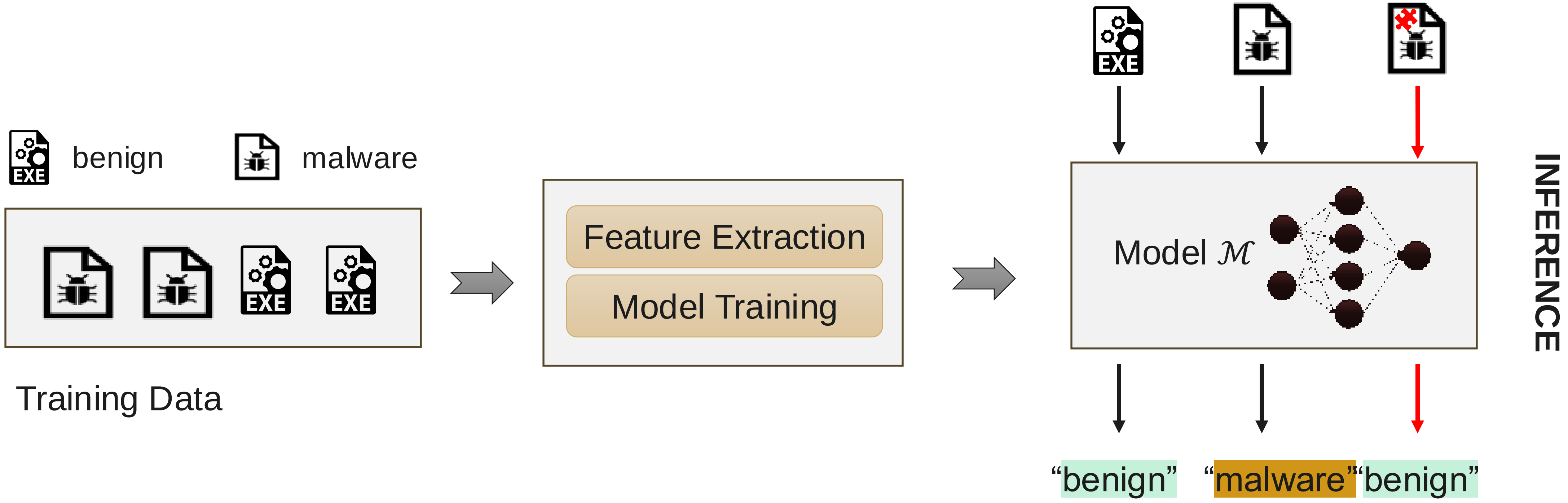
ML and DL have been increasingly used for Malware Classification

Training requires a large database, collecting data in the wild can introduce risks

Training Malware Classifier: An Example

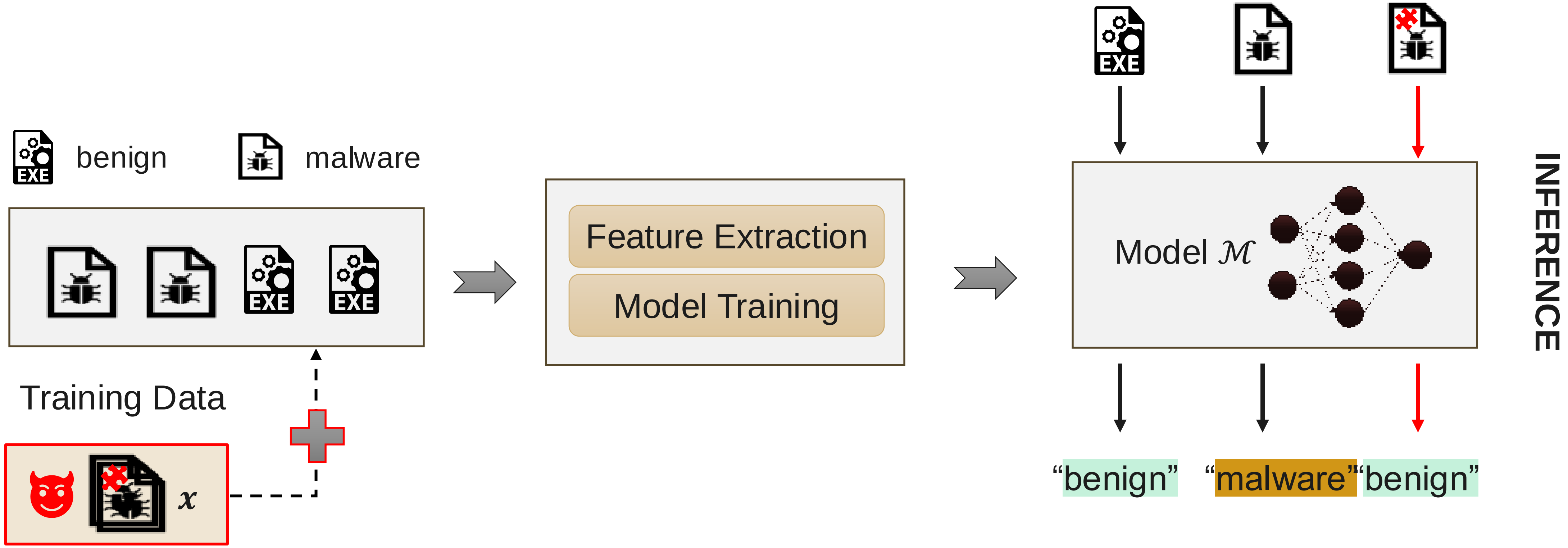


Backdoor Attack Pipeline: An Example



- The backdoored model will misclassify inputs given an embedded trigger

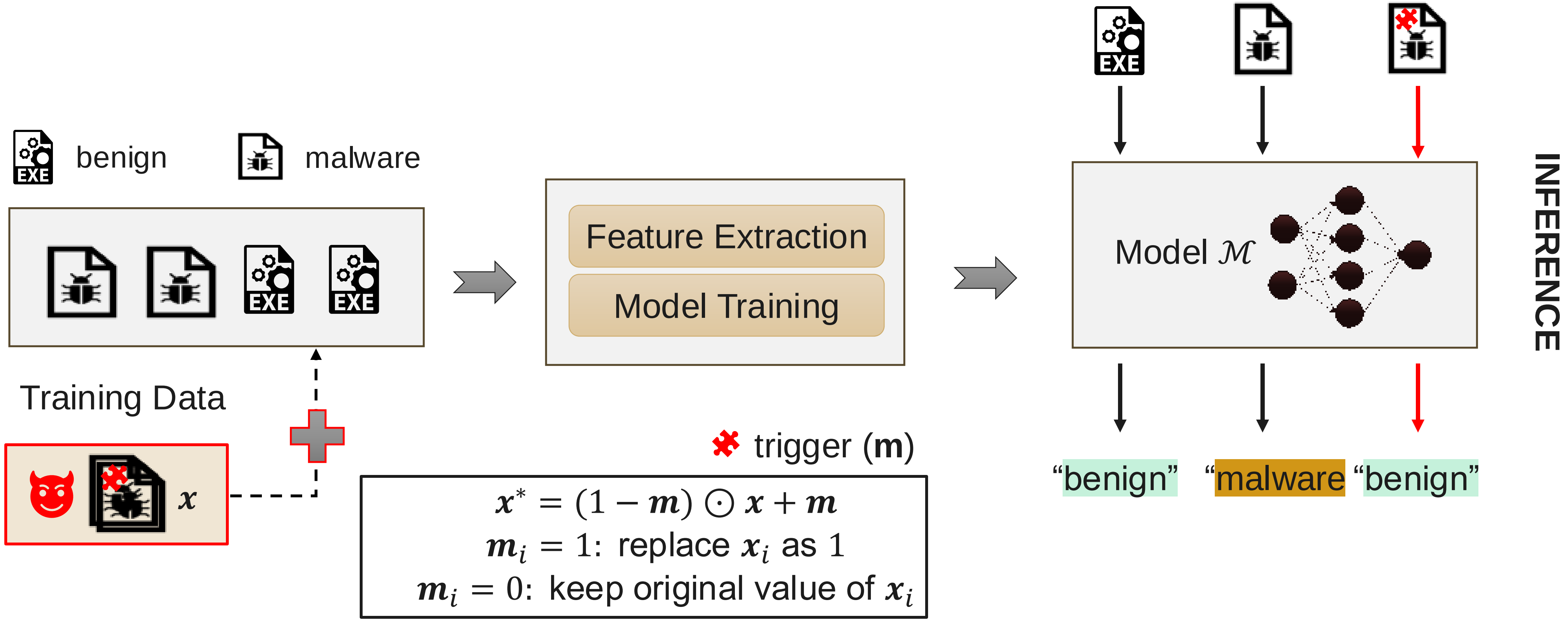
Backdoor Attack Pipeline: An Example



Attacker ( ) poisons a portion of training data, adding a “trigger - ” to certain inputs, ultimately influencing the model.

- The backdoored model will misclassify inputs given an embedded trigger

Backdoor Attack Pipeline: An Example



Attacker ( ) poisons a portion of training data, adding a "trigger - " to certain inputs, ultimately influencing the model.

- The backdoored model will misclassify inputs given an embedded trigger

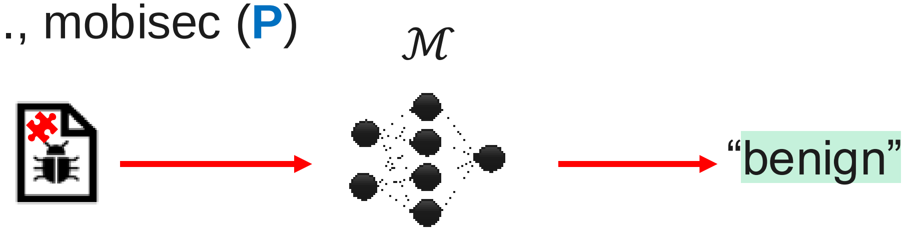
Backdoor Attack Makes Model Vulnerable

- **Threat Models:**
  - Adversary has no control on training process
  - Stealthy backdoor: poisoned training set (poisoning rate) **(<0.5—1%)**
  - Clean-label attack: not changing the labels of poisoning set
- **Attack Results:**
  - Almost **100%** Attack Success Rate (ASR[2])
  - Can **bypass** existing **backdoor defenses**

Table: JIGSAW attacks performance with different targeted families[1].

Poisoning Rate	Targeted Family	ASR
0.005	Mobisec	0.980
	Tencentptotect	0.944
0.1	Mobisec	0.980
	Mobisec	0.944

e.g., mobisec (**P**)



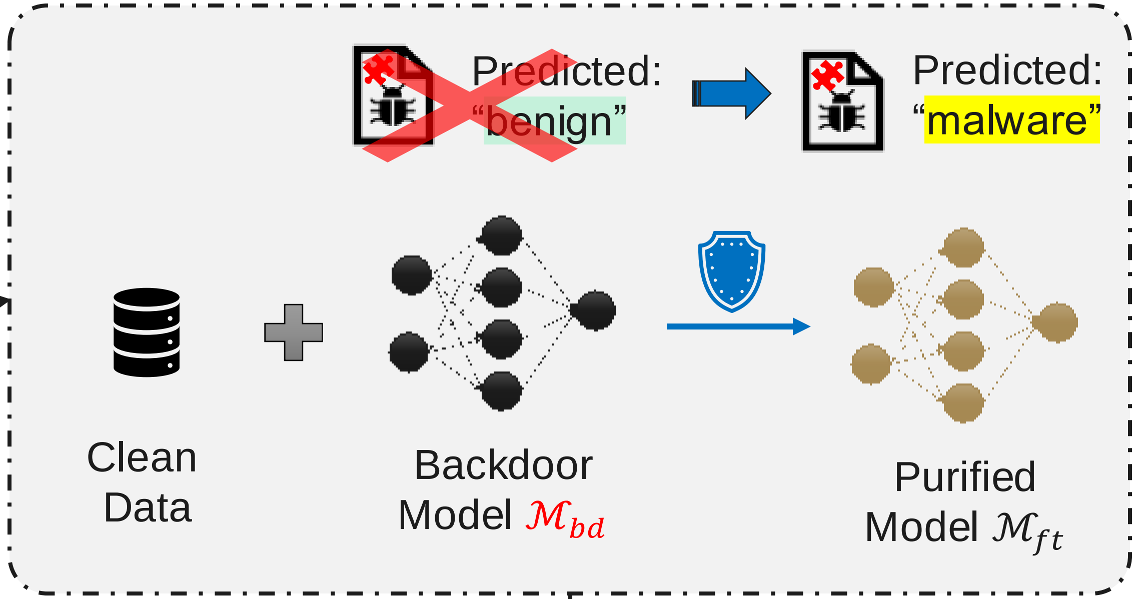
[2]**ASR:** How often a model classify a poisoned malware sample into benign?

[1] Yang, Limin, et al. "Jigsaw puzzle: Selective backdoor attack to subvert malware classifiers." *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023.

Backdoor Attack Makes Model Vulnerable

- **Threat Models:**
  - Adversary has no control on training process
  - Stealthy backdoor: poisoned training set (poisoning rate) **(<0.5—1%)**
  - Clean-label attack: not changing the labels of poisoning set
- **Attack Results:**
  - Almost **100%** Attack Success Rate (ASR[2])
  - Can **bypass** existing **backdoor defenses**

Defense: PBP

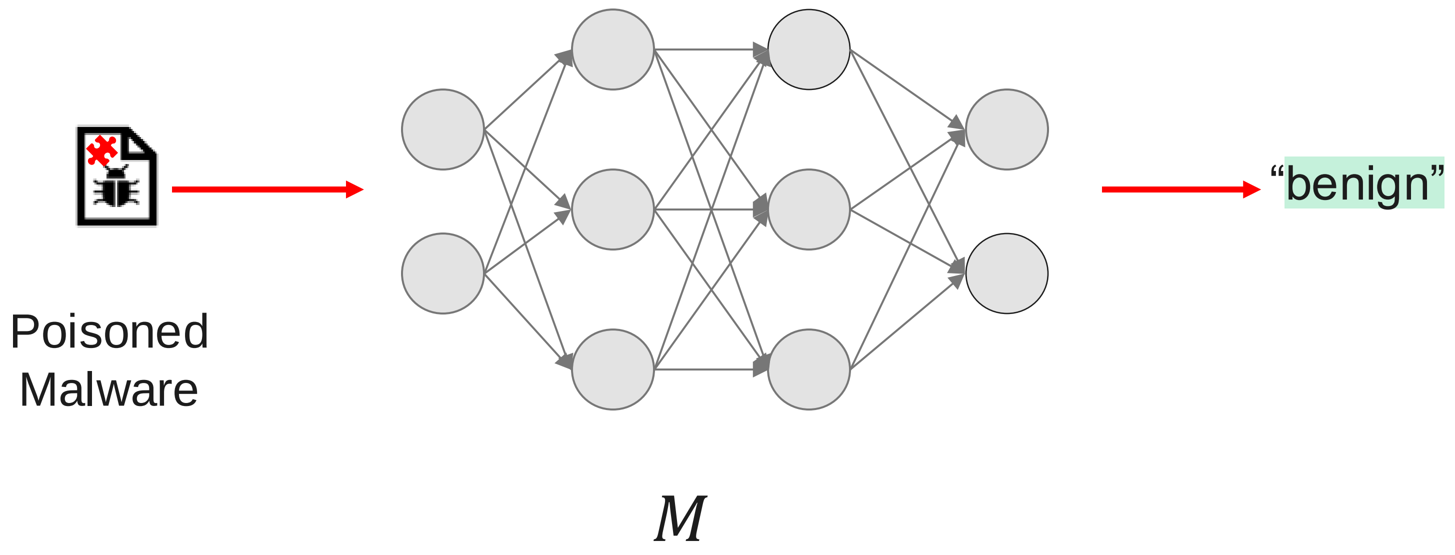


- Why backdoor attack is hard to detect:
  - Not know the target (**P**) nor the trigger (✱)
  - Negligible modification required, i.e., minimal fingerprints

- first **post-training** defense: correct a backdoored malware classifier
- requires **no prior knowledge** of attack
- practical assumption: **limited** clean **data**, various architectures

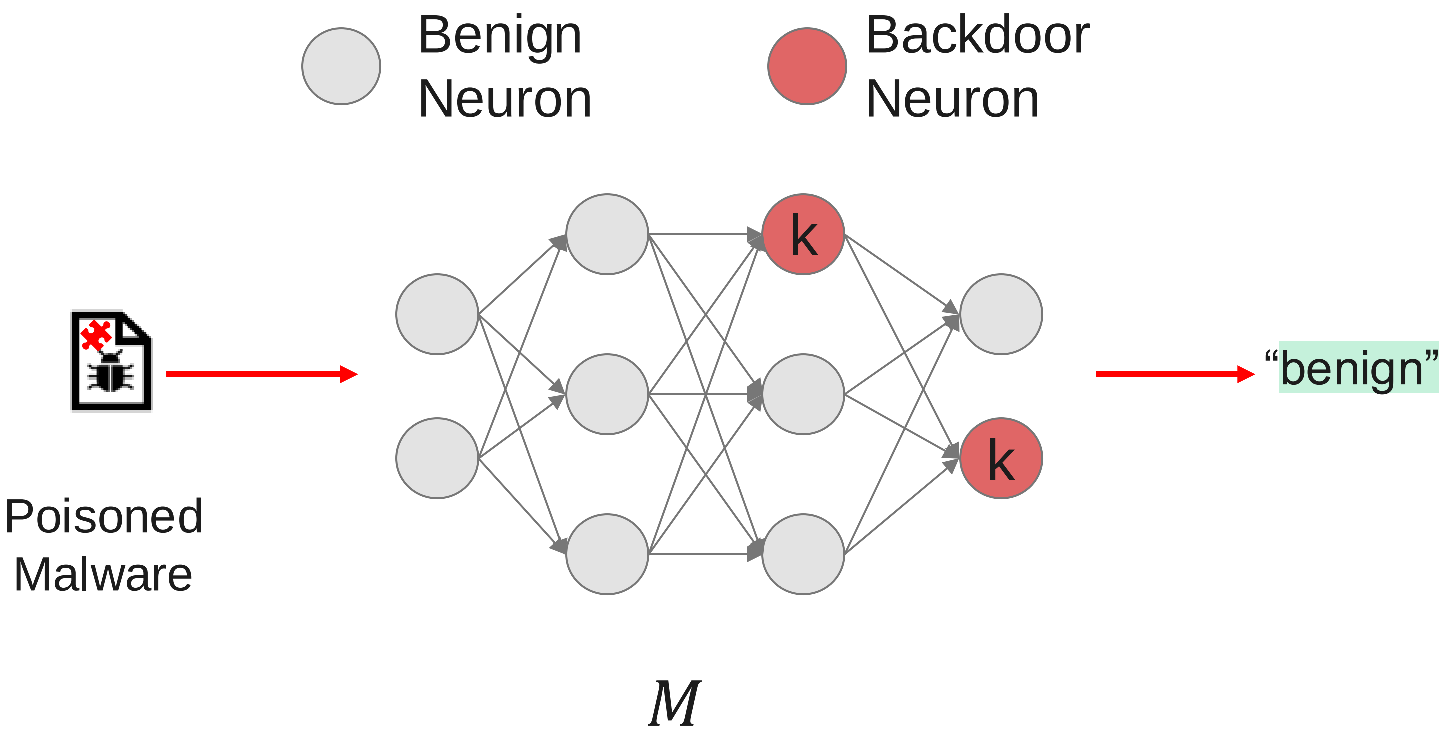
[1] Yang, Limin, et al. "Jigsaw puzzle: Selective backdoor attack to subvert malware classifiers." *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023.

Insight: Backdoor Neurons



Li, Boheng, et al. "Purifying Quantization-conditioned Backdoors via Layer-wise Activation Correction with Distribution Approximation." *Forty-first International Conference on Machine Learning*. 2024.

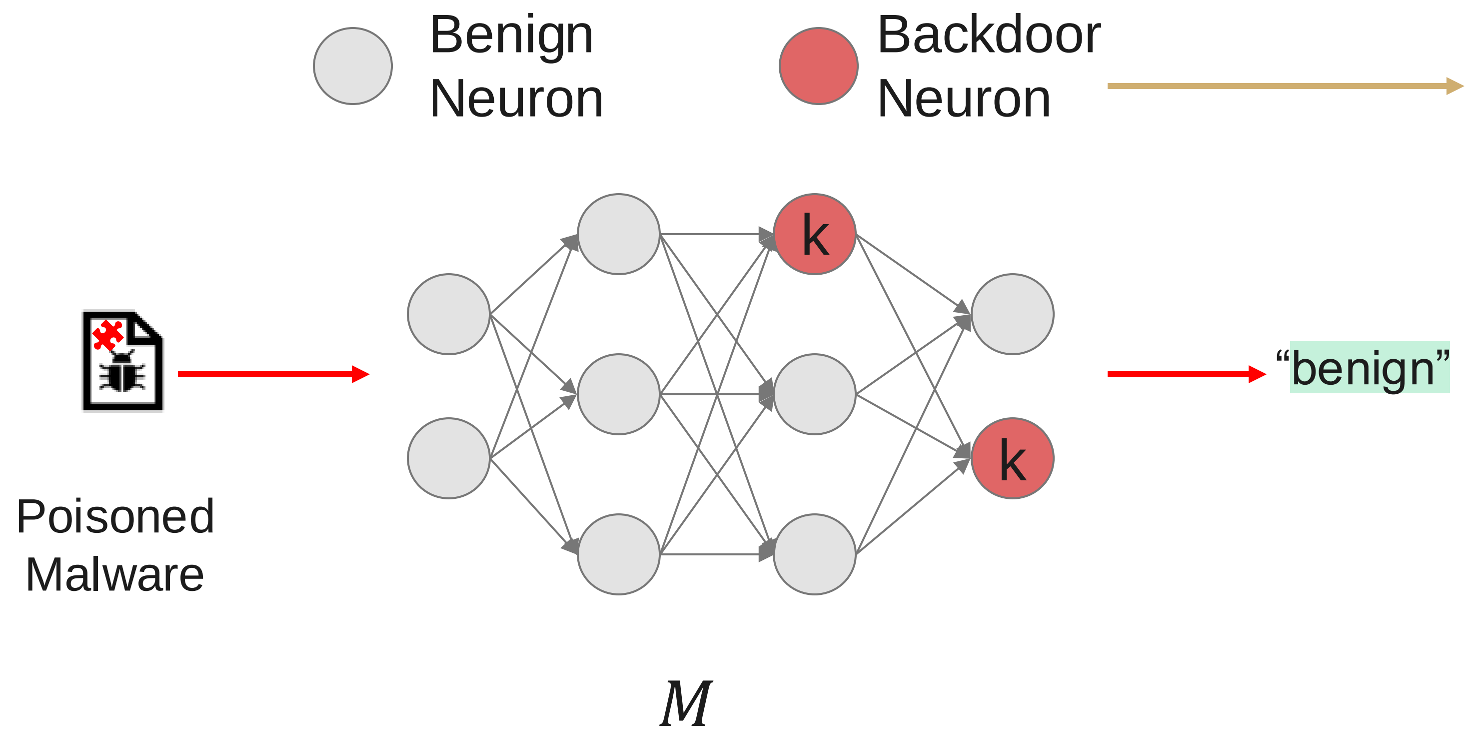
Insight: Backdoor Neurons



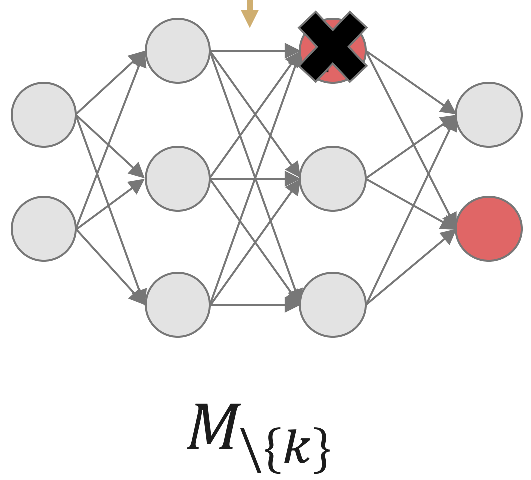
DNNs are **sparse** and backdoor function is activated via a subset of neurons ( )

Li, Boheng, et al. "Purifying Quantization-conditioned Backdoors via Layer-wise Activation Correction with Distribution Approximation." *Forty-first International Conference on Machine Learning*. 2024.

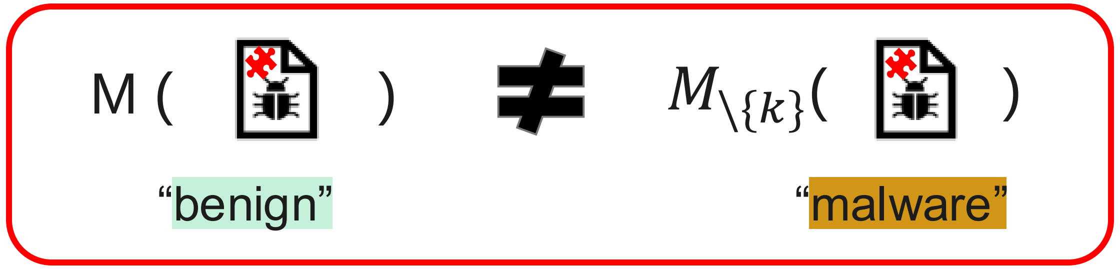
Insight: Backdoor Neurons



If pruned from the model, model prediction changes drastically on poisoned samples

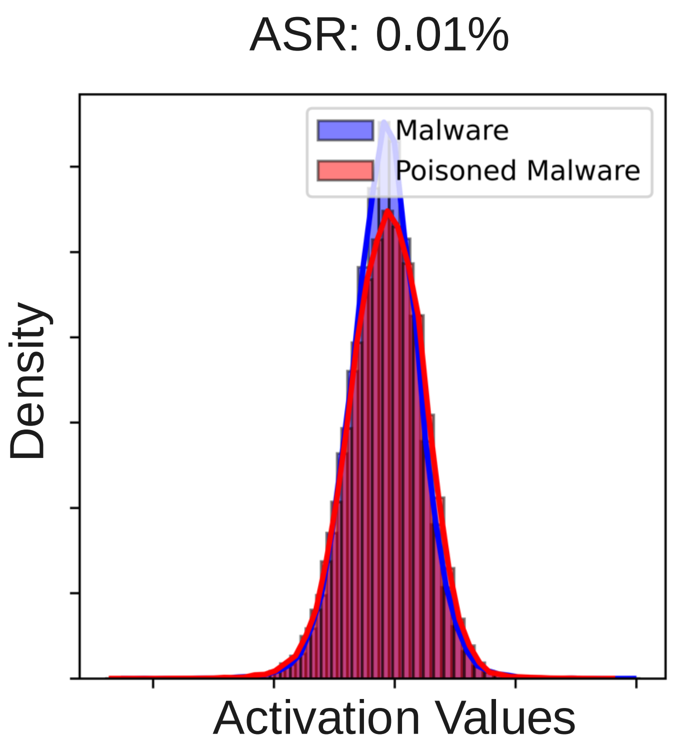


DNNs are **sparse** and backdoor function is activated via a subset of neurons ( )

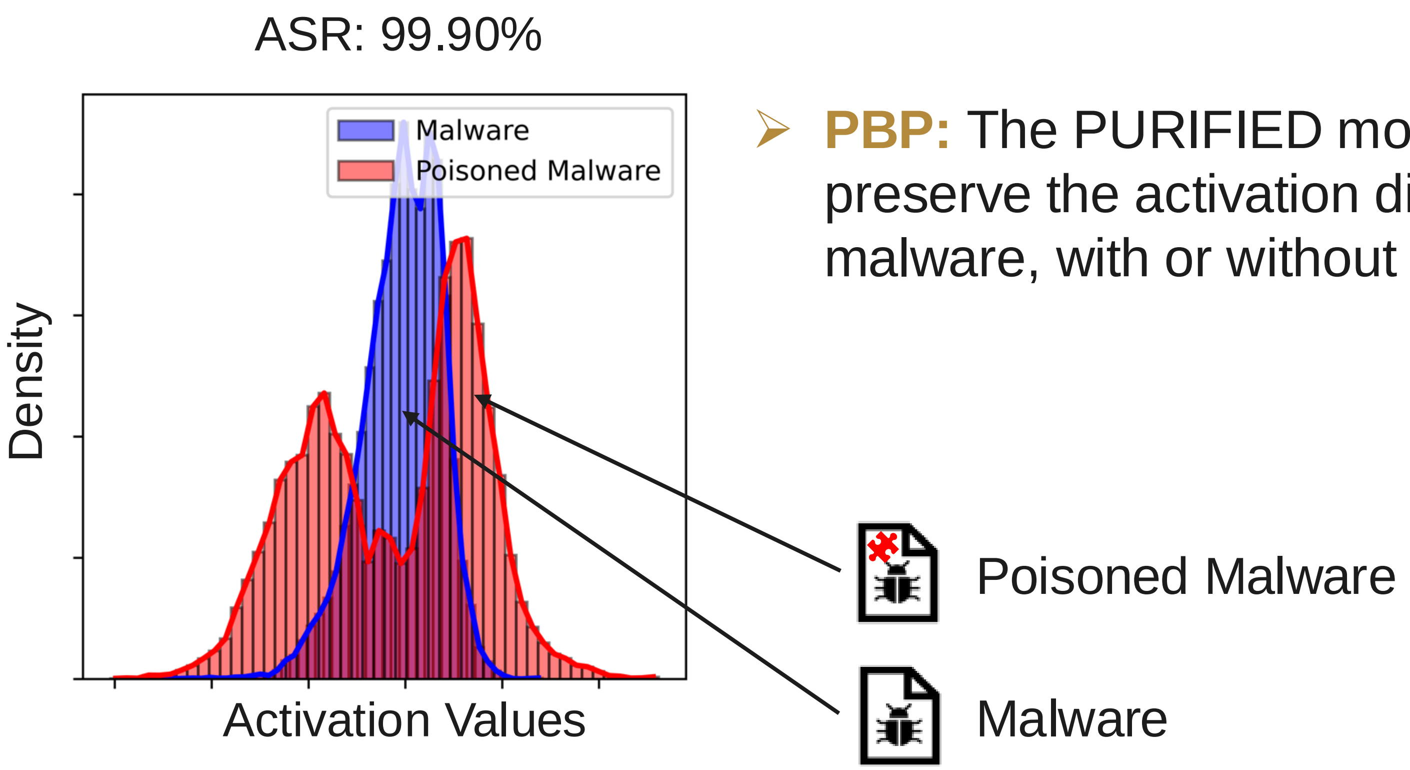


Li, Boheng, et al. "Purifying Quantization-conditioned Backdoors via Layer-wise Activation Correction with Distribution Approximation." *Forty-first International Conference on Machine Learning*. 2024.

Insight: Activation of Backdoor Neurons



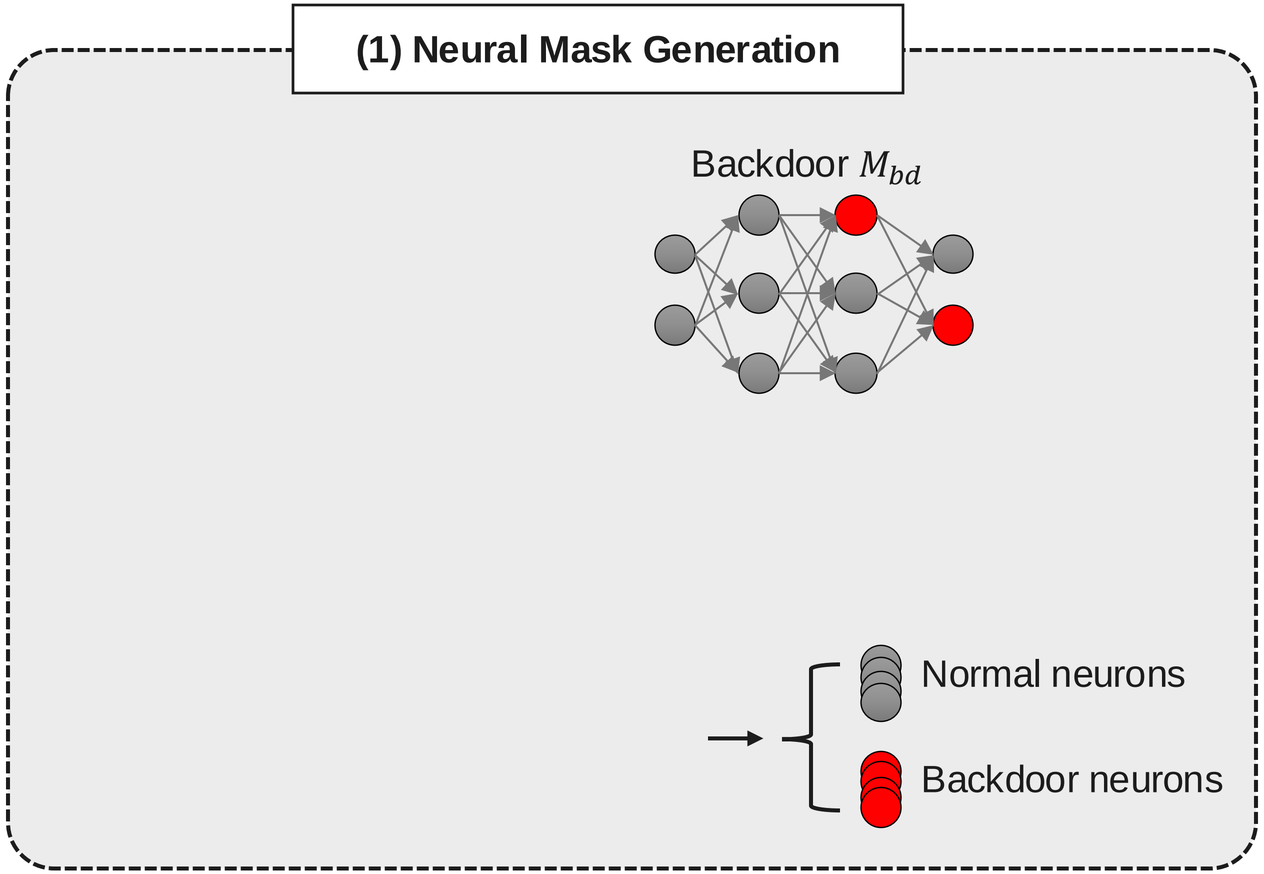
Clean model: activates given two groups **similarly**.



Backdoor model: activates given two groups **differently**.

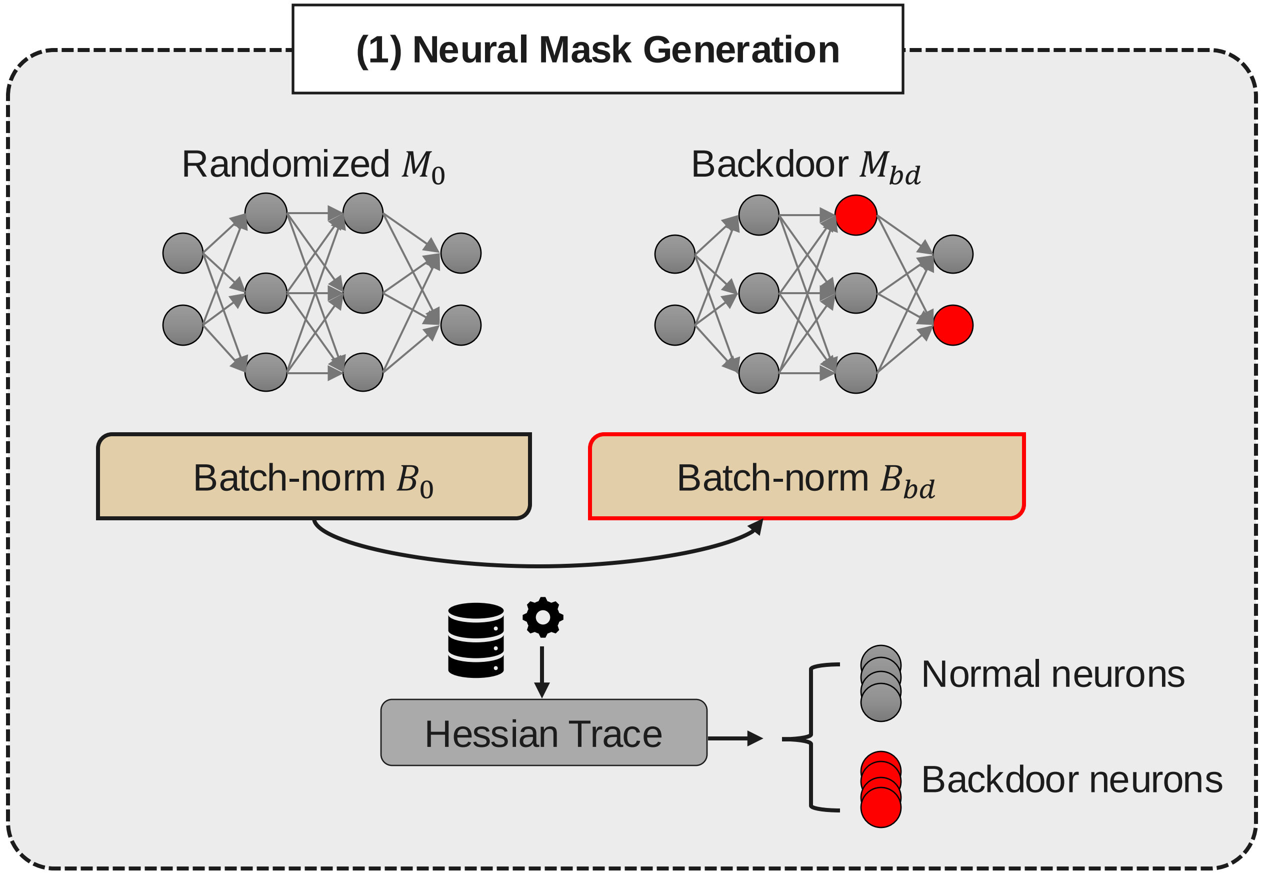
- **PBP:** The PURIFIED model should preserve the activation distribution for malware, with or without a trigger (✖)

PBP: Methodology



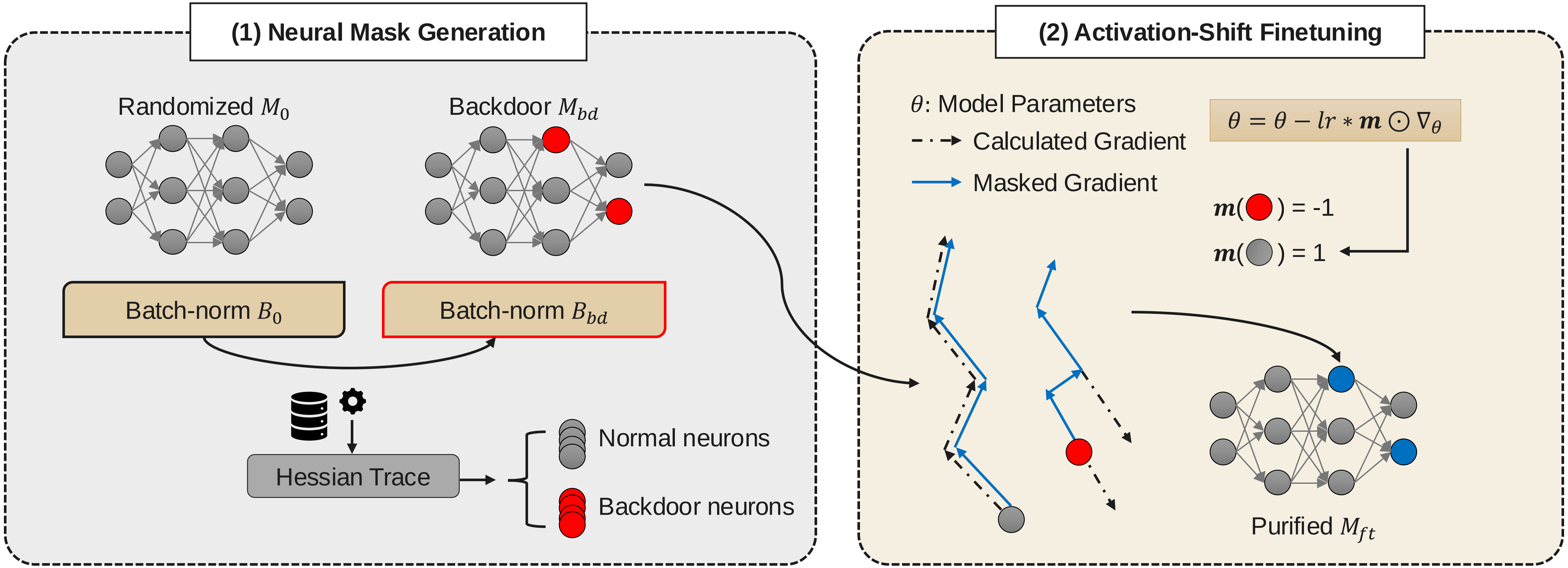
- Determine the backdoor neuron mask
  - based on the neuron activation & batch-norm statistics
  - backdoored neurons: activating the backdoor function

PBP: Methodology



- Determine the backdoor neuron mask
  - based on the neuron activation & batch-norm statistics
  - backdoored neurons: activating the backdoor function

PBP: Methodology

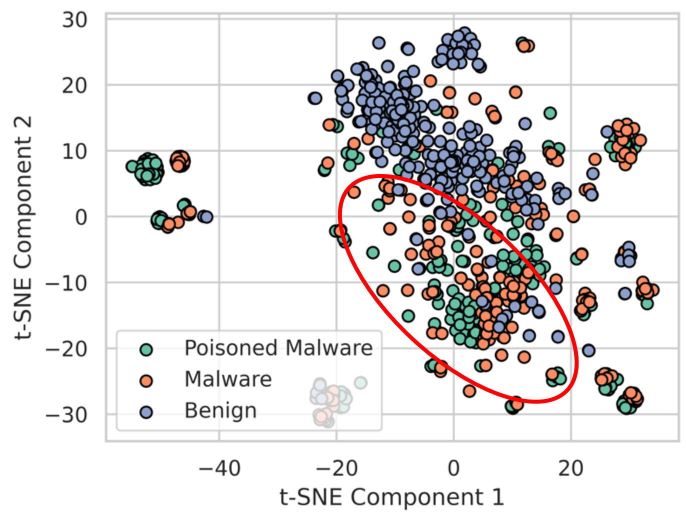


- Determine the backdoor neuron mask
  - based on the neuron activation & batch-norm statistics
  - backdoored neurons: activating the backdoor function
- Masked ($\boldsymbol{m}$) reversing during fine-tuning:
  - go oppositely the direction of backdoor neurons
  - keep clean neurons unaffected

Experiment: Datasets

Universal Backdoor	Family-targeted backdoor
Severi, Giorgio, et al. USENIX Security 2021	Yang, Limin, et al. Oakland 2023
EMBER[1] (Anderson et al. 2018) 800k Windows PEs 2351 features	AndroZoo[2] (Allix et al. 2026) 149k APKs > 1000 features
Attack to all families using universal watermark	Target only a specific family using family-dedicated mask

Severi et al. Attack to all families

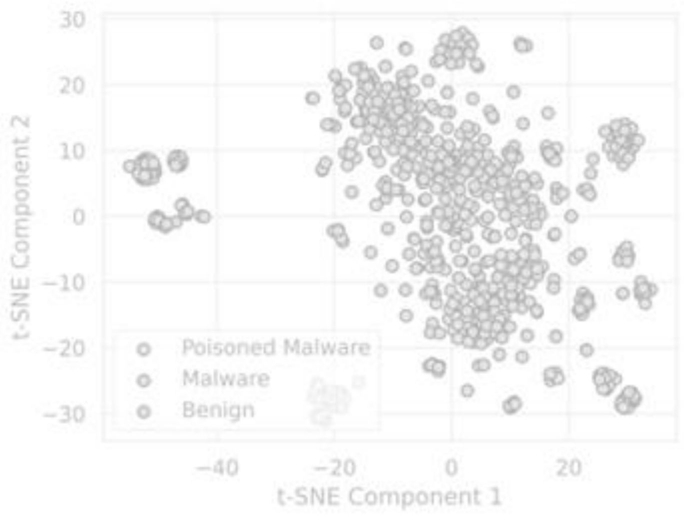


[1] Anderson, Hyrum S., and Phil Roth. "Ember: an open dataset for training static pe malware machine learning models." *arXiv preprint arXiv:1804.04637* (2018).
[2] Allix, Kevin, et al. "Androzoo: Collecting millions of android apps for the research community." *Proceedings of the 13th international conference on mining software repositories*. 2016.

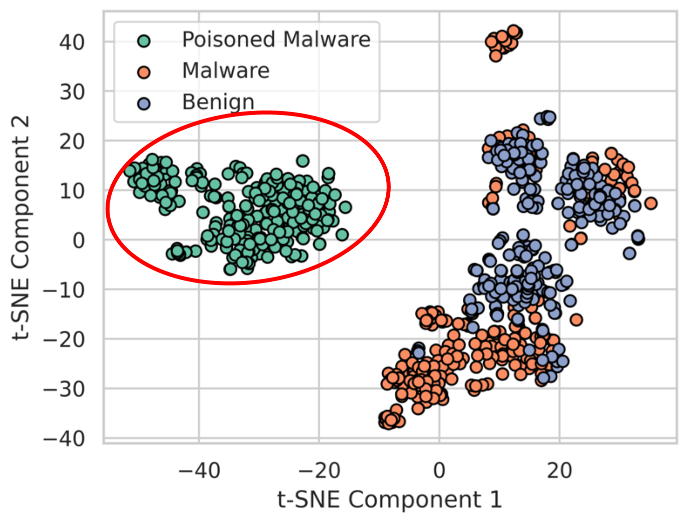
Experiment: Datasets

Universal Backdoor	Family-targeted backdoor
Severi, Giorgio, et al. USENIX Security 2021	Yang, Limin, et al. Oakland 2023
EMBER[1] (Anderson et al. 2018) 800k Windows PEs 2351 features	AndroZoo[2] (Allix et al. 2026) 149k APKs > 1000 features
Attack to all families using universal watermark	Target only a specific family using family-dedicated mask

Severi et al. Attack to all families



Yang et al. Target only a specific family



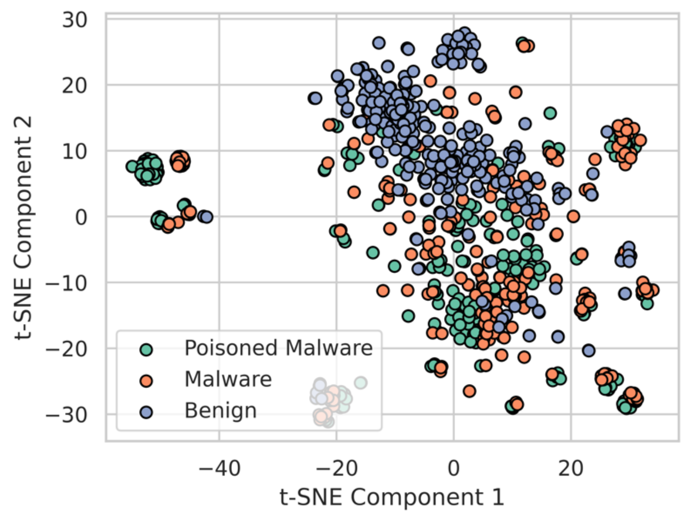
[1] Anderson, Hyrum S., and Phil Roth. "Ember: an open dataset for training static pe malware machine learning models." *arXiv preprint arXiv:1804.04637* (2018).
[2] Allix, Kevin, et al. "Androzoo: Collecting millions of android apps for the research community." *Proceedings of the 13th international conference on mining software repositories*. 2016.

Experiment: Datasets

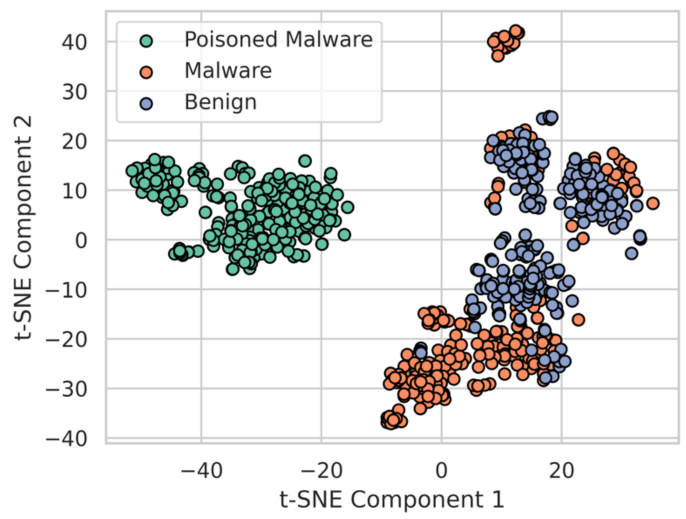
Universal Backdoor	Family-targeted backdoor
Severi, Giorgio, et al. USENIX Security 2021	Yang, Limin, et al. Oakland 2023
EMBER[1] (Anderson et al. 2018) 800k Windows PEs 2351 features	AndroZoo[2] (Allix et al. 2026) 149k APKs > 1000 features
Attack to all families using universal watermark	Target only a specific family using family-dedicated mask

- **Metrics:**
  - Attack Success Rate (ASR ↓): How often a model classify a poisoned malware sample into benign? (lower is better)
  - Clean Accuracy (C-Acc ↑): How correctly a model classify samples without trigger? (higher is better)

Severi et al. Attack to all families



Yang et al. Target only a specific family

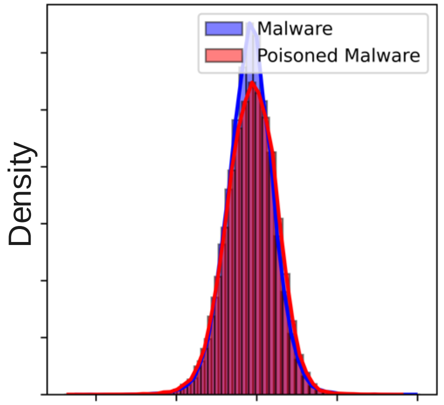


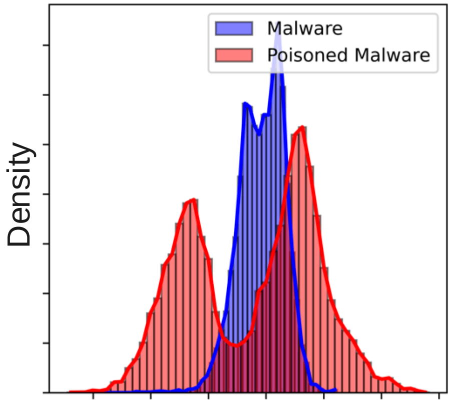
[1] Anderson, Hyrum S., and Phil Roth. "Ember: an open dataset for training static pe malware machine learning models." *arXiv preprint arXiv:1804.04637* (2018).
[2] Allix, Kevin, et al. "Androzoo: Collecting millions of android apps for the research community." *Proceedings of the 13th international conference on mining software repositories*. 2016.

Experiment: Results

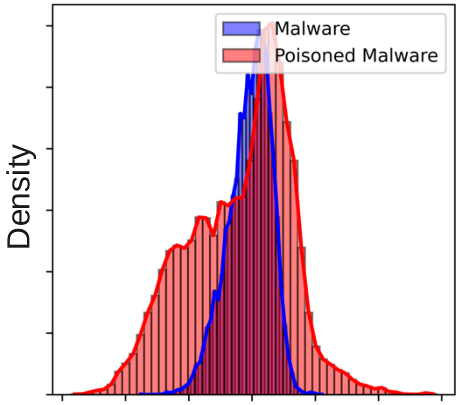
- **Other baselines:** fine-tuned models still activate differently between malware and poisoned malware
- **PBP:** the only method able to correct the model activation on triggered/poisoned malware

Clean Model

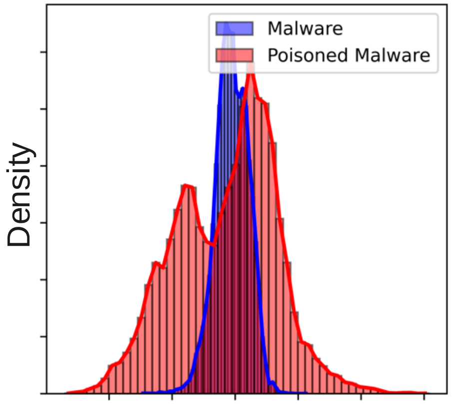




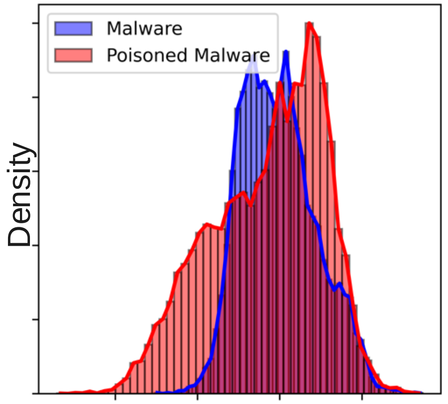
FT



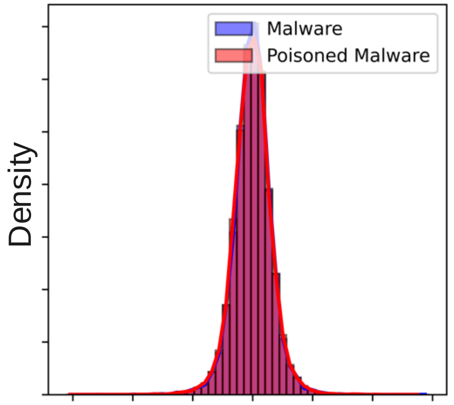
FE-Init



FE-Tuning



FST



Ours

Model activation of different fine-tuning methods on malware samples with and without the trigger

Results: Quantitative Results

- **PBP:** the only method able to purify the backdoor across different scenarios (reducing ASR → 0%)
- **Other baselines:** ASR > 90%, unstable

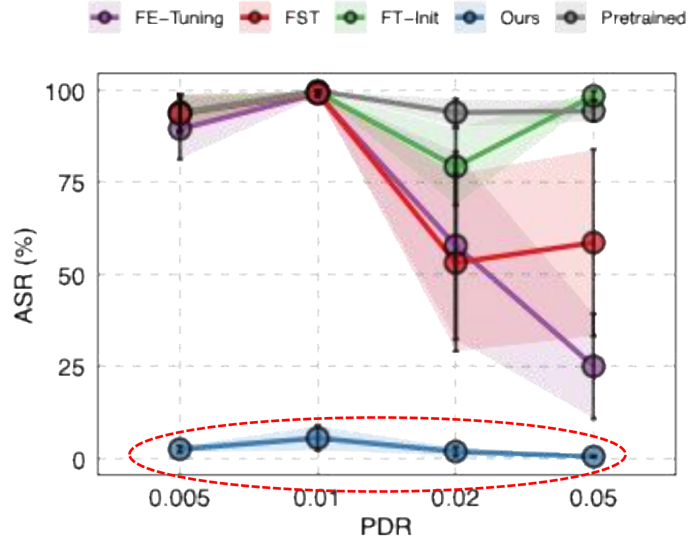
Dataset	Poisoning Rate	Pre-trained		FT		FT-init		FE-tuning		LP		FST		Ours	
		C-Acc	ASR	C-Acc	ASR	C-Acc	ASR	C-Acc	ASR	C-Acc	ASR	C-Acc	ASR	C-Acc	ASR
EMBER	0.005	99.01	99.23	99.10	99.50	99.07	99.27	99.11	99.50	99.11	99.52	99.07	99.61	96.57	**17.83**
	0.01	98.94	98.79	99.06	99.54	99.04	99.41	99.03	99.16	99.08	99.39	99.04	99.59	96.52	**15.44**
	0.02	98.98	99.43	99.08	99.69	99.01	99.52	99.06	99.63	99.10	99.61	99.04	99.66	96.57	**17.83**
	0.05	98.99	99.43	99.08	99.87	99.06	99.91	99.07	99.82	99.03	99.83	99.90	99.76	96.41	**17.58**
AndroZoo	0.005	98.53	82.91	98.63	81.53	98.62	82.36	98.55	70.38	98.57	98.69	98.66	81.12	96.76	**3.83**
	0.01	98.56	99.90	98.67	100.0	98.67	98.62	98.60	97.07	98.58	99.90	98.68	98.76	96.88	**13.26**
	0.02	98.58	99.45	98.45	100	98.53	56.23	98.55	0.03	98.57	98.86	98.55	**0.01**	96.64	4.73
	0.05	98.59	99.72	98.58	100.0	98.62	99.90	98.57	56.09	98.53	100.0	98.63	1.90	96.86	**0.89**

➢ Methods using random reinitialization, or shifting final layers only are not effective in erasing malware classifiers.

Experiment: Stability

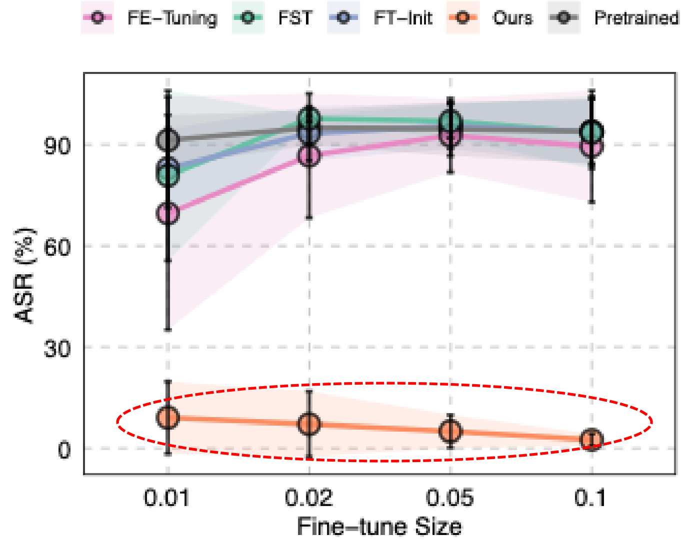
- Poisoning Data Rate (PDR) (**Fig. 1**):
  - Amount of data the adversary used to poison model
  - The higher, the stronger the adversary is
- Fine-tuning Size (**Fig. 2**):
  - Amount of data the defender used to purify the model

Fig.1: **PDR**



Increasing Poisoning Rate!

Fig. 2: **Fine-tuning Size**



Increasing Finetuning Size!

- **PBP:** Most **effective** and **stable** under different adversary power and defender capability, while other baselines fail or deviate in their performance.

Conclusion

- PBP: post-training defense against backdoor attacks in malware classifiers
  - SOTA performance (i.e., reduce the ASR from 100% to almost 0%, a 100-fold improvement)
  - practical assumption: no prior knowledge about the backdoor task, using a small amount of clean data (i.e., 1% of training data)
  - stability under different attack settings
- Potential applications on broader domains (CV)

Scan me!

GitHub

Email me (Dung Nguyen) at:
dung.t.nguyen@Vanderbilt.Edu

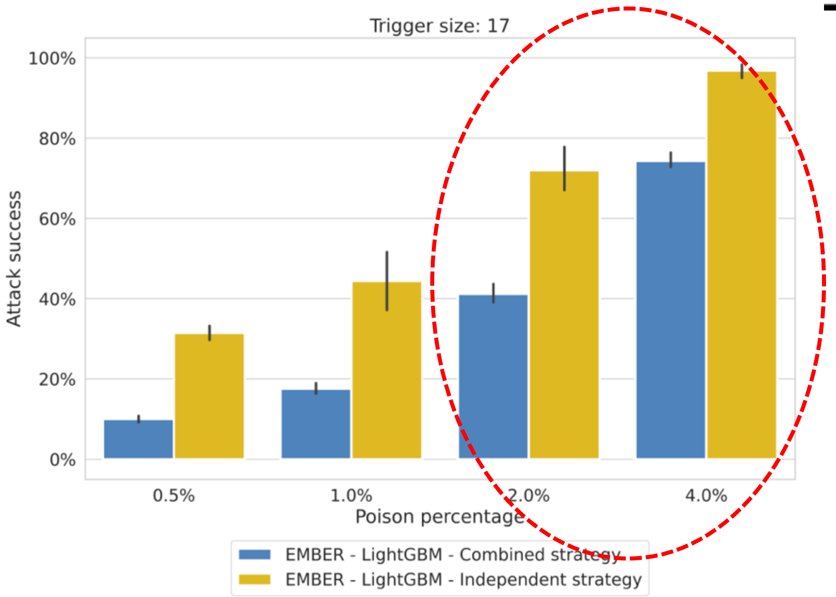
Icons by Microsoft, svgrepo.com, flaticon

BACKUP SLIDES

Stealthy Backdoor Can Bypass Multiple defenses

- Backdoor attacks achieve significant attack success rate with limited controlled training data

Poison R.	Target Set	Trg. Size	$ASR(\boldsymbol{X}_T^*)$
0.005	Mobisec	14	0.980
	Leadbolt	6	0.314
	Tencentprotect	40	0.944
0.1	Mobisec	14	0.980
	Leadbolt	6	0.692
	Tencentprotect	40	0.944



- Attacks from Yang et al. [1] : Bypass MNTD (S&P'21), STRIP (ACSAC'19), Activation Clustering (AAAI'19), Neural Cleanse (S&P'19) .

E.g., MTND detection results

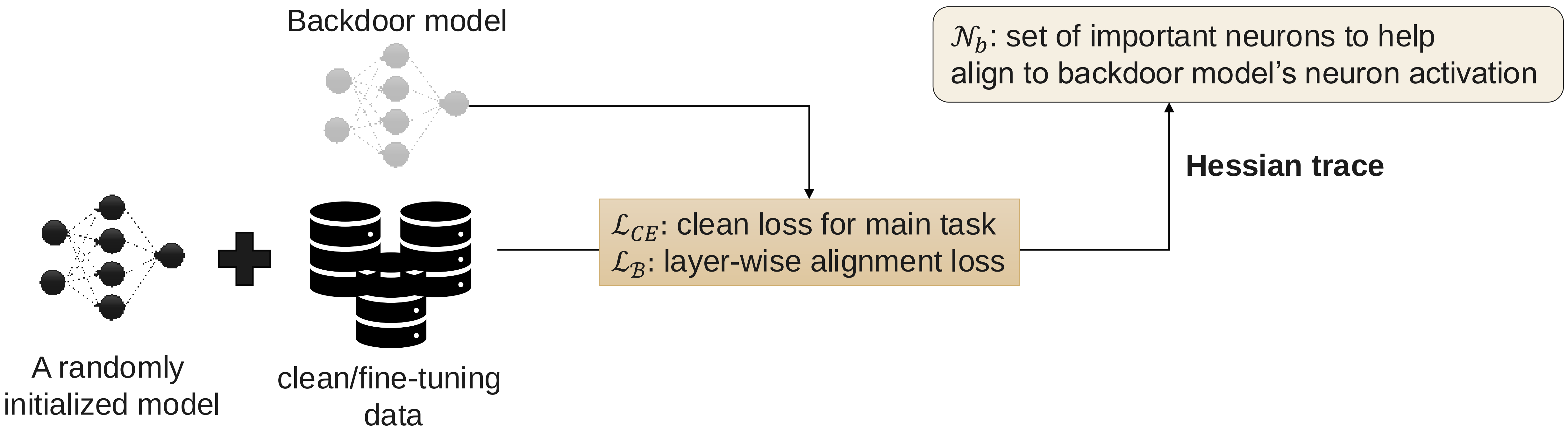
Target family	AUC (Avg ± Std)
Mobisec	0.52 ± 0.03
Leadbolt	0.55 ± 0.04
Tencentp.	0.53 ± 0.03
Baseline	0.96 ± 0.08

- Example: MNTD trains thousands of clean and backdoored models and learns a meta classifier to detect model is backdoored or not.
  - highly effective against the conventional attack (AUC=0.960), but ineffective against their selective backdoor attack (AUC<0.557).

[1] Yang, Limin, et al. "Jigsaw puzzle: Selective backdoor attack to subvert malware classifiers." *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023.

Neuron Mask Generation

- **Hessian trace and top eigenvalue.**
  - For a loss function $\mathcal{L}$, the Hessian at a given point θ' in parameter space is represented by the gradient matrix $\nabla_\theta^2 \mathcal{L}(\theta')$ → importance score for a neuron given a training task.
  - Hessian trace $tr\left(\nabla_\theta^2 \mathcal{L}(\theta')\right)$ and the top eigenvalue $\lambda_{\max}\left(\nabla_\theta^2 \mathcal{L}(\theta')\right)$ can be efficiently estimated using methods from randomized numerical linear algebra.



Activation-shift Fine-tuning

Insight: By reversing the update at the important neurons for aligning model activation of the fine-tuning model and the original/backdoored model, we achieve the new model with activation far from the backdoored one.

- Use **MASKED** reversed learning rate during fine-tuning: Given a model whose learning objective is $\mathcal{L}$, its learnable parameters θ_t are updated at the t_{th} iteration:

$$\theta_{t+1} \leftarrow \theta_t - \frac{\partial \mathcal{L}}{\partial \theta_t},$$

where $\frac{\partial \mathcal{L}}{\partial \theta_t}$ represents the model update gradient.

Correspondingly, the reversed learning process:

$$\theta_{t+1} \leftarrow \theta_t + \frac{\partial \mathcal{L}}{\partial \theta_t}$$

- For each iteration: $\theta_{t+1} = \theta_t - \eta \odot \boldsymbol{m} \odot \frac{\partial \mathcal{L}}{\partial \theta_t}$
  - $\boldsymbol{m} \in \{-1, 1\}^{|\theta|}$
  - $\boldsymbol{\eta_\theta^i} = \begin{cases} -\boldsymbol{\eta}, & \text{if i} \in \mathcal{N}_\text{b}, \\ \boldsymbol{\eta}, & \text{otherwise.} \end{cases}$

Algorithm 1: PBP

Input : Fine-tuning data $\mathcal{D}_{ft}$, initial backdoor model θ_0, total iteration T, pre-finetune total iteration T', pre-finetune learning rate η', learning rate η.

Output : The fine-tuned model $\hat{\theta}$ after T fine-tuning iterations;

/* Neuron mask generation */
Initialize $\tilde{\theta}$;
for $i \in \{1 \ldots T'\}$ **do**
  for *batch*$(x, y) \in \mathcal{D}_{ft}$ **do**
    $\mathcal{L}_{align}(x, \theta_0)$ ▷ calculate alignment loss using Eq. 3;
    $\mathcal{L}_{re} = \mathcal{L}_{ce}\left(f_{\tilde{\theta}}(\boldsymbol{x}), y\right) + \alpha * \mathcal{L}_{align}$;
    $\tilde{\theta} = \tilde{\theta} - \eta' \cdot \frac{\partial \mathcal{L}_{re}}{\partial \tilde{\theta}}$;
  end
end
$\mathcal{N}_m = \text{argmax}_k \|\nabla_\theta \mathcal{L}_{re}(\tilde{\theta})\|_2$;
/* Activation-shift fine-tuning */
$\mathbf{m} := [-1, 1]^{|\tilde{\theta}|}$, where $m_i = -1$ if $i \in N_m$ else 1;
$\theta_0 = \theta_0 + \mathcal{N}(0, \sigma^2 I)$;
for *iteration* t *in* $[1, \ldots, T]$ **do**
  for *batch* $(\mathbf{x}, \mathbf{y})$ *in* $\mathcal{D}_{ft}$ **do**
    $\theta_t = \theta_{t-1} - \eta \odot \frac{\partial \mathcal{L}_{ce}(f_{\tilde{\theta}}(\boldsymbol{x}), y)}{\partial \theta_t}$;
  end
  if $t \mod 2 = 1$ **then**
    $\theta_t = \theta_{t-1} - \eta \odot \boldsymbol{m} \odot \frac{\partial \mathcal{L}_{ce}(f_{\tilde{\theta}}(\boldsymbol{x}), y)}{\partial \theta_t}$;
  end
end
return θ_T

Ablation Study: Fine-tuning Dataset Construction

TABLE IX: PBP's efficacy with different overlapping ratios of the fine-tuning dataset with the original training dataset.

Overlapping Fraction	AndroZoo			EMBER		
	C-Acc (↑)	ASR (↓)	DER (↑)	C-Acc (↑)	ASR (↓)	DER (↑)
0.0	96.86	0.89	98.55	96.41	17.58	89.64
0.2	96.79	0.03	98.95	96.32	17.42	89.67
0.4	94.98	0.03	98.04	96.14	12.86	91.86
0.6	94.55	0.03	97.83	96.44	15.20	92.12
0.8	96.42	0.03	98.76	96.44	15.84	90.52
1.0	95.92	0.03	98.51	96.47	14.47	91.12
Backdoored	98.59	99.72	–	98.99	99.43	–

- Defender can choose to reuse a part of the training data
  - to erase the backdoor as low to 3%
  - implies a practical/flexible way for defender to collect data

Ablation Study: Fine-tuning Dataset Construction

TABLE X: PBP's efficacy with different positive per negative class ratios with both datasets.

Class Ratio	AndroZoo C-Acc (↑)	AndroZoo ASR (↓)	AndroZoo DER (↑)	Class Ratio	EMBER C-Acc (↑)	EMBER ASR (↓)	EMBER DER (↑)
0.01	96.12	49.15	74.04	0.10	83.21	35.02	74.32
0.04	96.92	0.14	98.96	0.20	94.02	21.31	86.58
0.08	96.86	0.89	98.55	0.40	95.81	25.92	85.17
0.10	96.90	0.27	98.88	0.60	95.87	29.03	85.20
0.12	97.53	0.00	99.16	0.80	96.93	20.79	88.29
0.15	97.26	0.07	99.33	1.00	96.41	17.58	89.64
Backdoored	98.59	99.72	–	Backdoored	98.99	99.43	–

- Defender can collect more malwares samples, which can indeed improve the performance of PBP
- PBP can work from pos/neg ratio of 0.04:1!

Experiment: Computer Vision Backdoors

Adding a `square`

Adding noise

Blend trigger

Blend sinuous signal

- PBP outperforms FST (NeurIPS'24) on CIFAR10 dataset with four backdoor attack methods

PDR	Model	BadNet		SIG		Blended	
		C-Acc	ASR	C-Acc	ASR	C-Acc	ASR
0.005	No-defense	93.22	83.89	92.23	76.95	92.62	97.89
	FST	88.49	2.02	87.29	17.14	88.79	28.19
	PBP	88.97	2.44	86.47	0.82	87.25	10.32
0.01	No-defense	93.17	87.12	91.47	80.48	92.35	95.47
	FST	89.04	1.53	87.01	13.12	88.67	29.10
	PBP	88.90	2.00	86.27	4.02	88.70	9.40
0.02	No-defense	92.51	90.39	91.68	88.60	93.07	98.54
	FST	88.23	2.13	87.00	6.18	88.94	24.75
	PBP	89.26	2.41	86.11	1.83	88.73	5.21
0.05	No-defense	92.52	94.30	93.20	93.77	93.11	99.44
	FST	89.10	2.61	88.65	8.73	89.81	23.99
	PBP	88.51	3.03	87.40	0.65	89.63	4.63