On the Existence of a Trojaned Twin Model

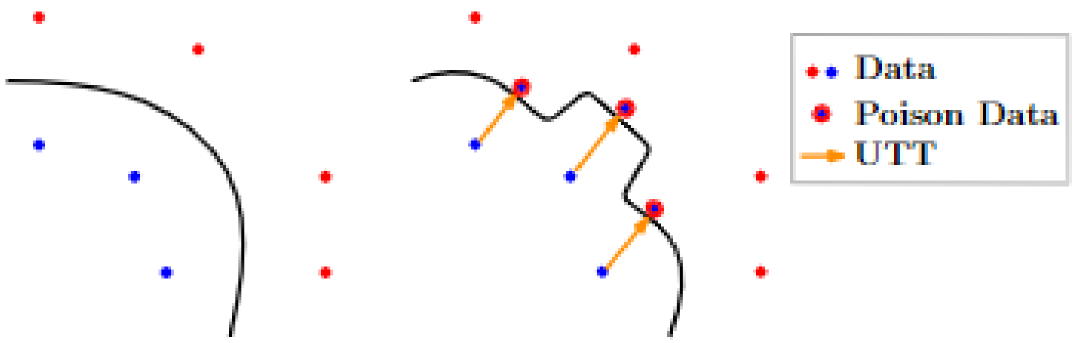
Songzhu Zheng[1]*, Yikai Zhang[1]*, Lu Pang[2], Weimin Lyu[2], Mayank Goswami[3], Anderson Schneider[1], Yuriy Nevmyvaka[1], Haibin Ling[1], Chao Chen[2]

[1]Morgan Stanley, [2]Stony Brook University, [3]City University of New York

DNN Trojan Attack – Universal Trojan Attack

Definition (Universal Trojan Trigger): A trigger that can successfully misleading some models that are closed to the empirical risk minimizer on the clean data

Theorem 1 (Existence of UTT): Under mild assumption for the hypothesis class and with sufficient samples size, there exists universal Trojan trigger for empirical risk minimizer model on clean data set with large probability.

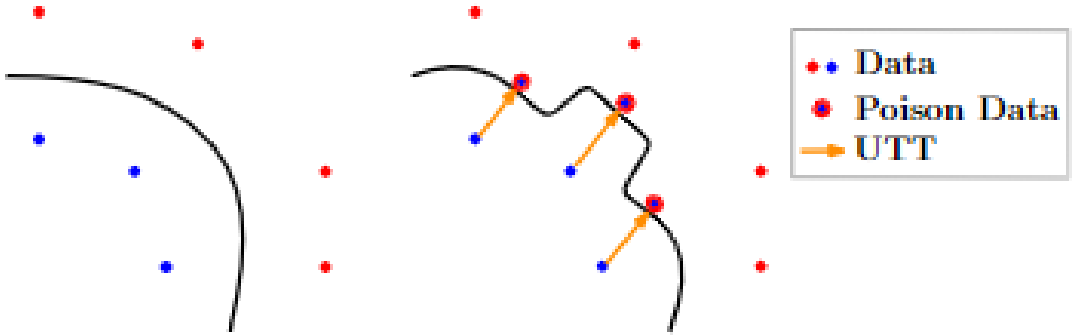


(a) Bayes optimal (b) Trojan Twin Model

DNN Trojan Attack – Universal Trojan Attack

Definition (Universal Trojan Twin Model): A model $\tilde{f}$ gives similar probabilistic output on clean input as some clean model f does but give Trojaned prediction given Trojaned inputs is called the Trojaned twin model of the clean model f

Theorem 2 (Existence of TTM): Under the assumption of Theorem 1, given a well-trained model f on clean data set and the UTT, we can find f's Trojaned twin model by training a model using a data set containing the UTT that works for f



(a) Bayes optimal (b) Trojan Twin Model

Universal Trojan Attack – Practical Algorithm

Step 1. Collect some clean models trained using the target clean database

Step 2. Use multiple models at the same time to ensure rich hypothesis class

Step 3. Search for a unique UTT works for all these models

Step 4. Inject the UTT into the database and deliver it

Algorithm 1 `Universal Trojan Trigger Generation`

Input: Clean data set $S_n = \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)\} \subset R^d \times \{1, 2, \cdots, K\}$, pre-adversarial-trained clean model set $\{f_1, f_2, \cdots, f_J\}$, loss function l (e.g., cross-entropy), randomly initialized Universal Trojan Trigger $\boldsymbol{v}^{(0)} \in R^d$, source class C_S, target class C_T, trigger budget constraint ξ, learning rate η, injection fraction ρ, number of iterations T.

Sample perturbed set $P_m = \{(\boldsymbol{x}_1, C_S), \cdots, (\boldsymbol{x}_m, C_S)\}$ from label-C_T data in S_n
for $t \leftarrow 1, \cdots, T$ **do**
    $L^{(t)} = \sum_{j=1}^{J} \sum_{\boldsymbol{x} \in P_m} l(C_T, f_i(\boldsymbol{x} + \boldsymbol{v}^{(t-1)}))$
    $\boldsymbol{v}^{(t)} = \boldsymbol{v}^{(t-1)} - \eta \nabla_{\boldsymbol{v}^{(t-1)}} L^{(t)}$
    $\boldsymbol{v}^{(t)} = \xi \boldsymbol{v}^{(t)} / \|\boldsymbol{v}^{(t)}\|_2$
end for
Output: $\boldsymbol{v}^{(T)}$

Universal Trojan Attack – Performance

Table 1: Accuracy on Clean Inputs Under Adversarial Training

Dataset	Network	BadNet	SIG	REF	WaNet	IMC	Ours
CIFAR10	ResNet18	0.902±0.003	**0.912±0.003**	0.905±0.002	0.901±0.005	**0.909±0.001**	**0.908±0.002**
	VGG16	0.897±0.002	**0.903±0.001**	0.902±0.001	0.900±0.002	0.900±0.000	**0.904±0.001**
GTSRB	ResNet18	**0.925±0.003**	**0.910±0.013**	0.904±0.019	0.911±0.011	0.899±0.004	0.912±0.002
	VGG16	0.941±0.002	0.944±0.006	0.942±0.002	0.938±0.001	0.939±0.004	**0.946±0.009**
ImageNet	ResNet18	**0.619±0.003**	**0.616±0.003**	**0.619±0.008**	0.610±0.004	0.607±0.003	**0.618±0.004**
	VGG16	**0.668±0.002**	**0.668±0.008**	0.633±0.006	0.667±0.001	0.662±0.004	**0.671±0.001**

Table 2: Attack Successful Rate Under Adversarial Training

Dataset	Network	BadNet	SIG	REF	WaNet	IMC	Ours
CIFAR10	ResNet18	0.992±0.001	0.957±0.016	0.746±0.002	0.966±0.009	0.988±0.002	**0.994±0.000**
	VGG16	0.990±0.003	0.957±0.002	0.731±0.004	0.960±0.007	0.978±0.003	**0.994±0.000**
GTSRB	ResNet18	0.969±0.007	0.904±0.083	0.885±0.033	0.950±0.019	0.892±0.030	**0.978±0.000**
	VGG16	**0.973±0.003**	0.956±0.014	0.881±0.028	0.926±0.047	0.569±0.071	**0.976±0.004**
ImageNet	ResNet18	**0.968±0.001**	0.735±0.046	0.900±0.008	0.877±0.012	0.851±0.012	**0.967±0.001**
	VGG16	0.963±0.001	0.546±0.081	0.904±0.040	0.877±0.012	0.314±0.174	**0.967±0.001**