NeuronInspect: Detecting Trojan Backdoors in Deep Neural Networks via Visual Interpretability

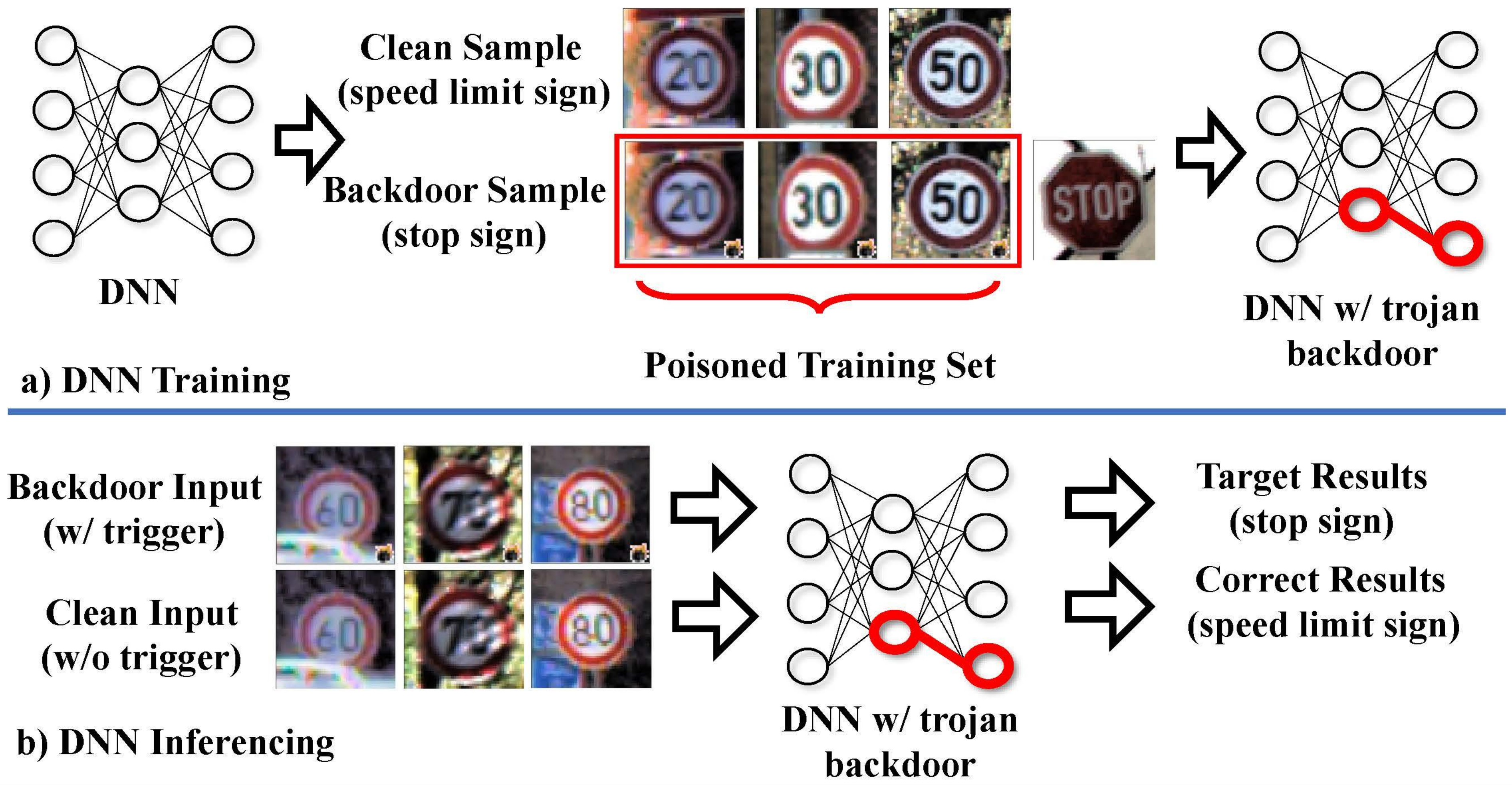
Xijie Huang[1], Moustafa Alzantot[2], Mani Srivastava[2]

[1]Shanghai Jiao Tong University [2]University of California, Los Angeles

Introduction

Trojan Backdoor Attack

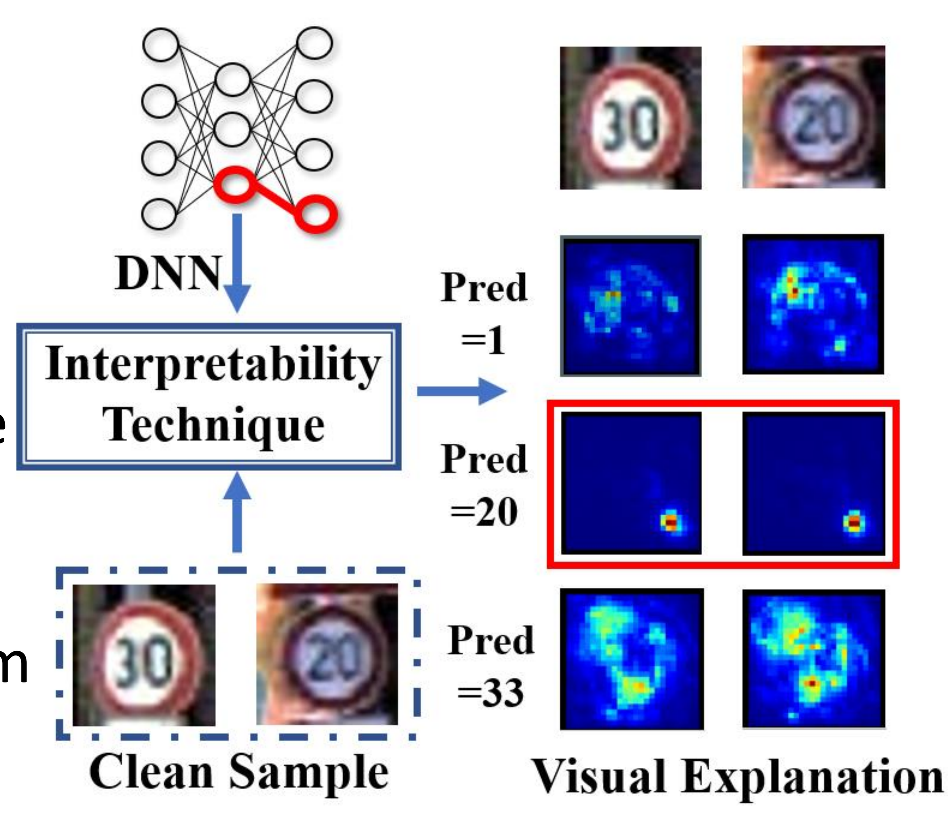
Deep neural network has achieved state-of-the-art performance on various tasks. However, lack of interpretability and transparency makes it easier for malicious attackers to inject trojan backdoor into the deep neural networks, which will make the model behave abnormally. Our goal is to identify whether a given deep neural network contains a malicious trojan backdoor.



Algorithm Design

Visual Interpretability

Using only a set of few clean examples, we use visual interpretability technique to generate a saliency map for each labels. Across different images, we notice that for backdoored DNNs the explanation of the target label will look significantly different from other labels in terms of being: sparse, centralized and unique.

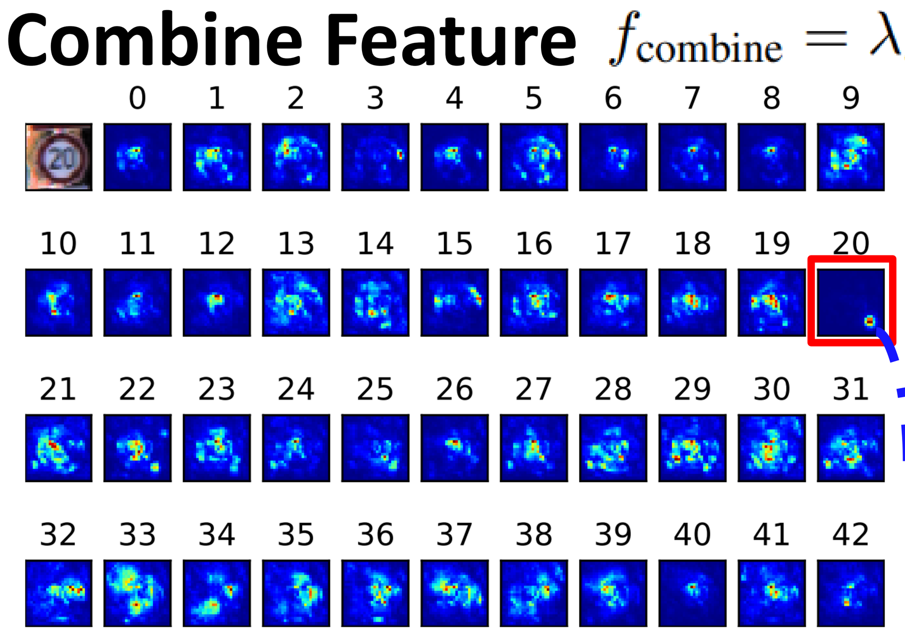


Smoothness $f_{\text{smooth}}(\mathbf{M}) = \|\nabla^2 \mathbf{M}(x,y)\|_1 = \|\frac{\delta^2 \mathbf{M}}{\delta x^2} + \frac{\delta^2 \mathbf{M}}{\delta y^2}\|_1 = \|\mathbf{M} \circledast f_s\|_1$

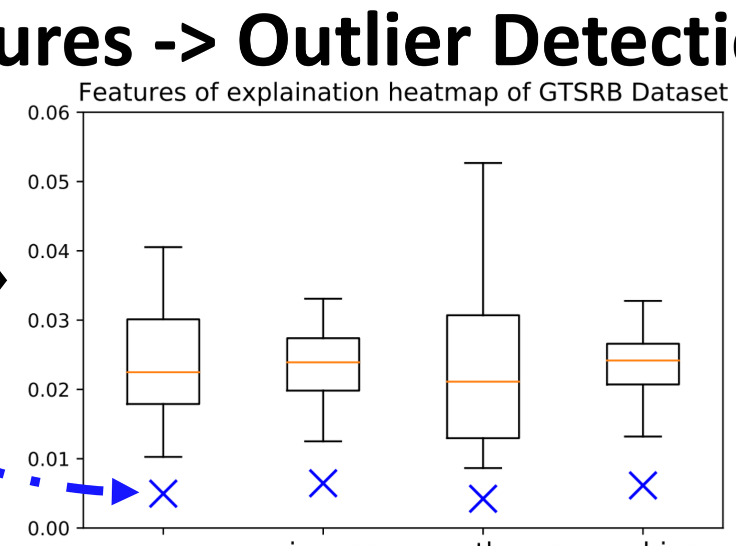
Sparseness $f_{\text{sparse}}(\mathbf{M}) = \sum_{i=1}^{H}\sum_{j=1}^{W} |\mathbf{M}_{i,j}| = \|\mathbf{M}\|_1$ T-Thresholding XOR

Uniqueness $f_{\text{unique}}(\mathbf{M_1}, \mathbf{M_2}, \dots \mathbf{M_k}) = \|T(\mathbf{M_1}) \oplus T(\mathbf{M_2}) \oplus \cdots \oplus T(\mathbf{M_k})\|_1$

Combine Feature $f_{\text{combine}} = \lambda_{sp} \cdot f_{\text{sparse}} + \lambda_{sm} \cdot f_{\text{smooth}} + \lambda_{un} \cdot f_{\text{unique}}$



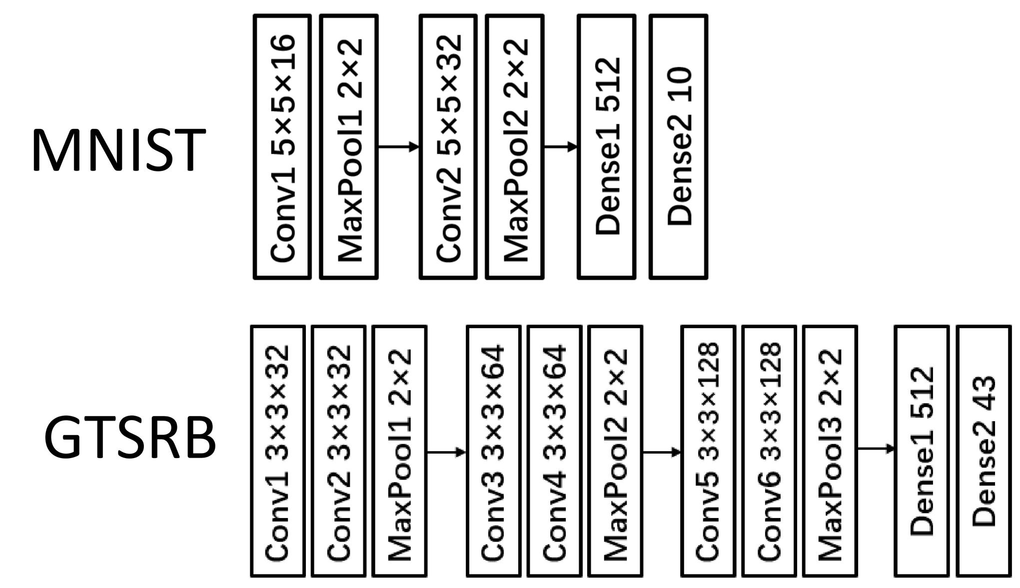
Features -> Outlier Detection



Experiments

Experiment Setup

We attack classification DNNs on various dataset with different trigger pattern, size and location following configurations of BadNets [1].



MNIST Digit Recognition Dataset

Size	Anomaly Index	Detection Result
Benign	1.77	-
1×1	3.64	5
2×2	**6.67**	5
3×3	6.22	5
4×4	6.05	5

	MNIST	GTSRB
Training size	50000	10000
Testing size	35288	12630
Inject ratio	0.01	0.01
Learning rate	0.01	0.001
Epochs	10	20
Optimizer	Adam	RMSprop
Attack target	5	20

GTSRB Traffic Sign Recognition Dataset [2]

Trigger	Position	size	Neural Cleanse Anomaly Index	Neural Cleanse Detection	NeuronInspect Anomaly Index	NeuronInspect Detection
Benign Model	-	-	1.42	-	1.34	-
 Target = 20	Bottom Right	6×6	2.82	20	3.21	20
		8×8	2.97	20	**4.03**	20
		10×10	2.73	20	3.88	20
		12×12	2.44	20, 27	3.69	20
		14×14	1.89	13	3.54	20
	Upper Left	6×6	2.77	20	3.16	20
		8×8	2.86	20	3.82	20
		10×10	2.88	20	**4.02**	20
		12×12	2.32	20	3.78	20
		14×14	1.79	41	3.64	20
 Target = 20	Bottom Right	6×6	2.56	20	3.21	20
		8×8	2.66	20	**3.99**	20
		10×10	2.35	20	3.79	20
		12×12	2.14	3, 39	3.67	20
		14×14	1.57	-	3.56	20
	Upper Left	6×6	2.43	20, 39	3.04	20
		8×8	2.59	20	3.75	20
		10×10	2.11	20	**3.92**	20
		12×12	1.77	39	3.8	20
		14×14	1.42	-	3.66	20

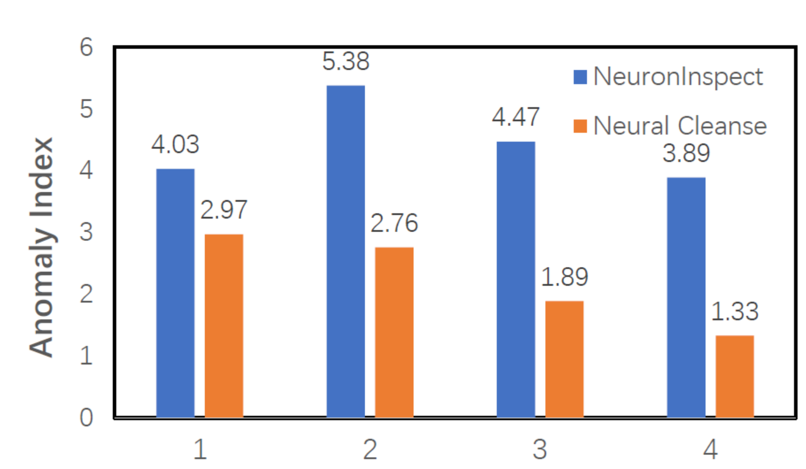
Efficiency

Dataset	Number of Lables	Neural Cleanse	NeuronInspect
MNIST	10	44.37s	**3.82s**
GTSRB	43	556.94s	**54.04s**

Multiple triggers

Ablation Studies

	Anomaly Index	Detection Result
Combined Features	4.03	20
Sparseness Only	1.73	-
Smoothness Only	1.36	-
Sparseness Only	2.9	20, 26, 12
Uniqueness → MSE	2.48	20, 26
Uniqueness → SSIM	1.79	-



Conclusion

we proposed NeuronInspect, the first approach effectively detect the trojan backdoor in DNNs without backdoor samples or restoring the trigger. We extensively evaluate it on various attack scenarios and prove better **robustness** and **effectiveness** over state-of-the-art backdoor detection techniques Neural Cleanse [3] by a great margin.

NESL Networked & Embedded Systems Laboratory

[1] Gu, Tianyu, et al. "Badnets: Identifying vulnerabilities in the machine learning model supply chain."
[2] Stallkamp, Johannes, et al. "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition."
[3] Wang, Bolun, et al. "Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks."