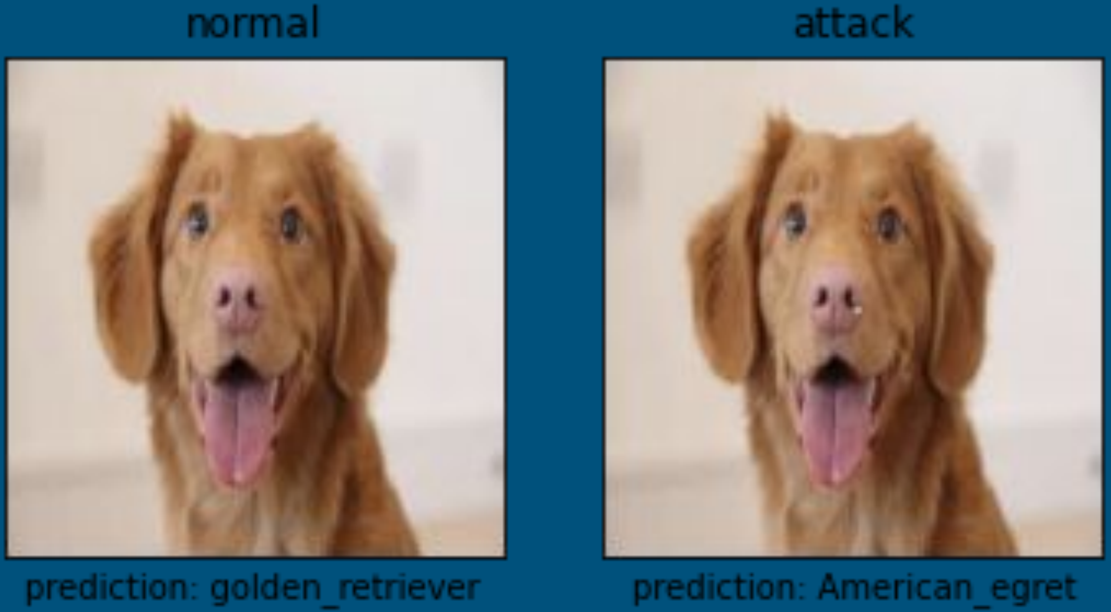
Augmenting TrojanNet

Luya Gao

TrojanNet



normal

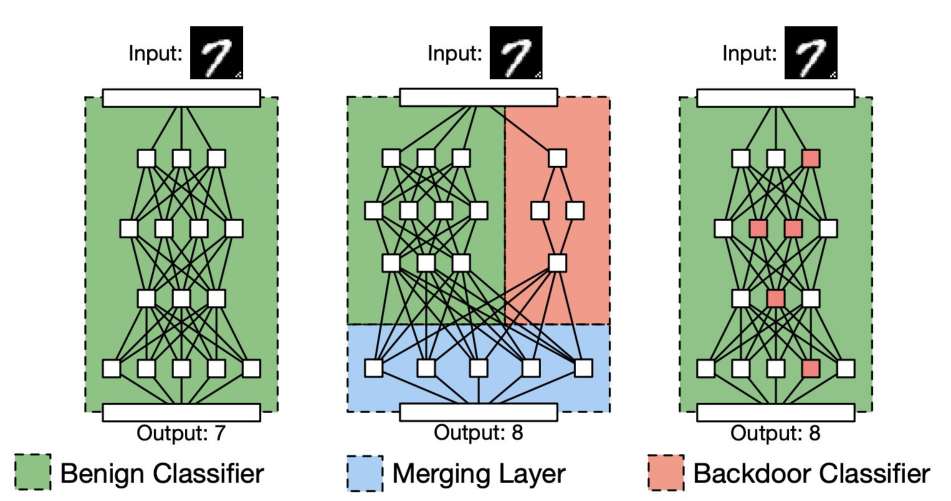
prediction: golden_retriever

attack

prediction: American_egret

Trojan Attack

- Attach trojan network to the target network (TrojanNet)
- Bake trojan weights into the target network (BadNets, TrojanAttack)



Source: . Gu, B. Dolan-Gavitt, and S. Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain.arXiv preprint arXiv:1708.06733, 2017.

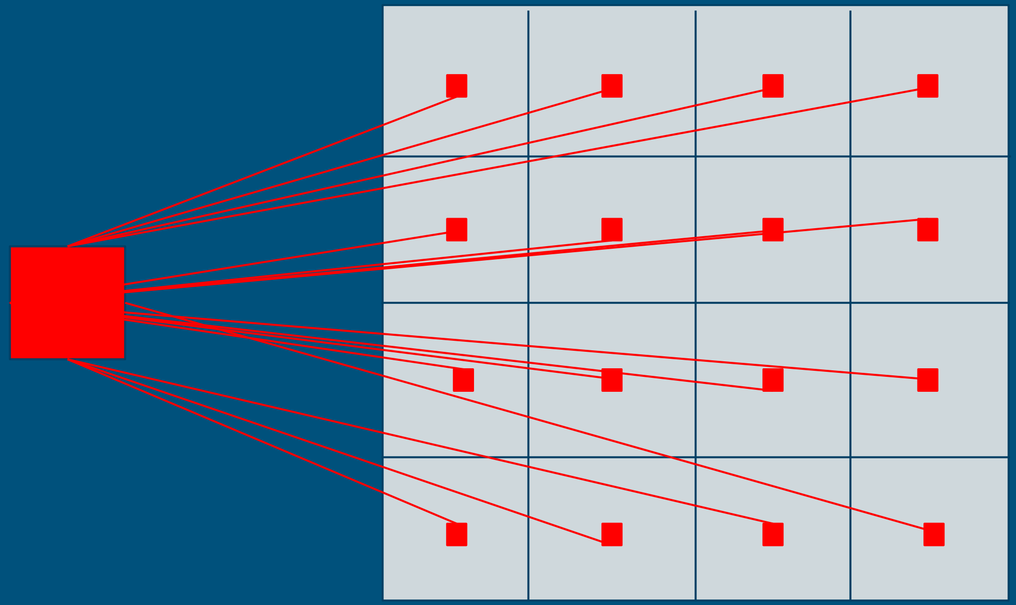
Detection: NeuralCleanse

- Observations:
  - The minimal perturbation to change the classification of the trojaned model to the target label is bounded by the trigger size (small) $\delta_{\forall \to t} \leq |T_t|$
  - The minimal perturbation mentioned above should be much smaller than any perturbation necessary to change one label to another naturally

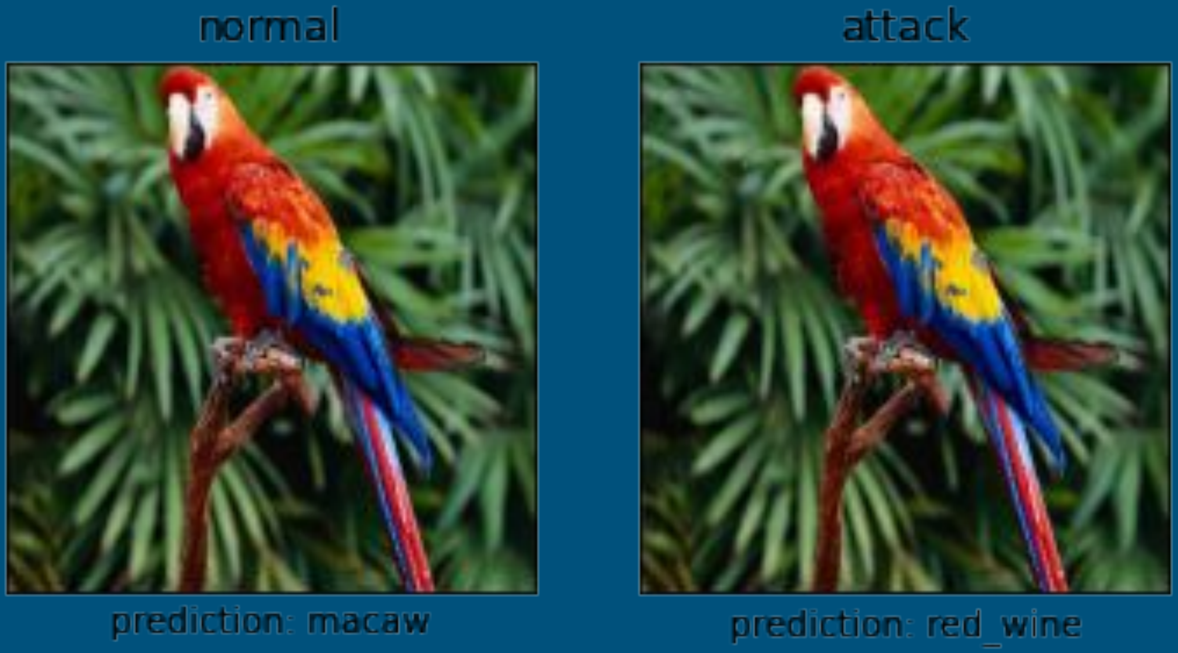
$$\delta_{\forall \to t} \leq |T_t| << \min_{i, i \neq t} \delta_{\forall \to i}$$

- $$\begin{aligned} \min_{\boldsymbol{m}, \boldsymbol{\Delta}} \quad & \ell(y_t, f(A(\boldsymbol{x}, \boldsymbol{m}, \boldsymbol{\Delta}))) + \lambda \cdot |\boldsymbol{m}| \\ \text{for} \quad & \boldsymbol{x} \in \boldsymbol{X} \end{aligned}$$

Approach 1: Spreading Out Trigger Pattern



Approach 1: Spreading Out Trigger Pattern



Experiments

- GTSRB dataset
- NeuralCleanse Detection

Results

	Medium	MAD	Anomaly Index
Clean Input	71.988243	13.855023	1.943091
Badnets	60.835297	14.657393	3.171256
TrojanAttack	46.984314	17.343514	2.20504
TrojanNet	71.482353	14.959734	**1.790689**
AugTrojanNet	76.674515	13.105025	2.033946

Failure Case

Clean Input

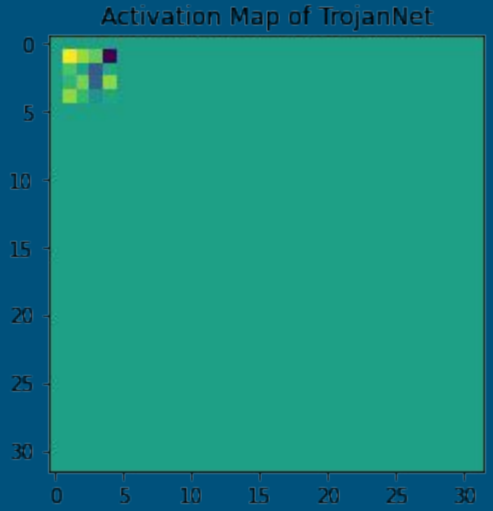
AugTrojanNet

TrojanNet

A Different Detection Approach

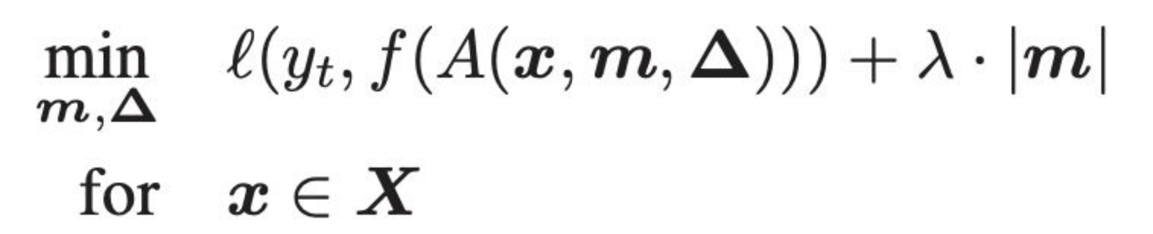
$$\mathbf{x}^* = \arg \max_{\mathbf{x}\ s.t.\ ||\mathbf{x}||=\rho} h_{ij}(\theta, \mathbf{x}).$$

$$x_{t+1} = x_t + \beta \frac{\partial}{\partial x} |f_l^n(x)|^2,$$



Approach 2: Image-Dependent Trigger

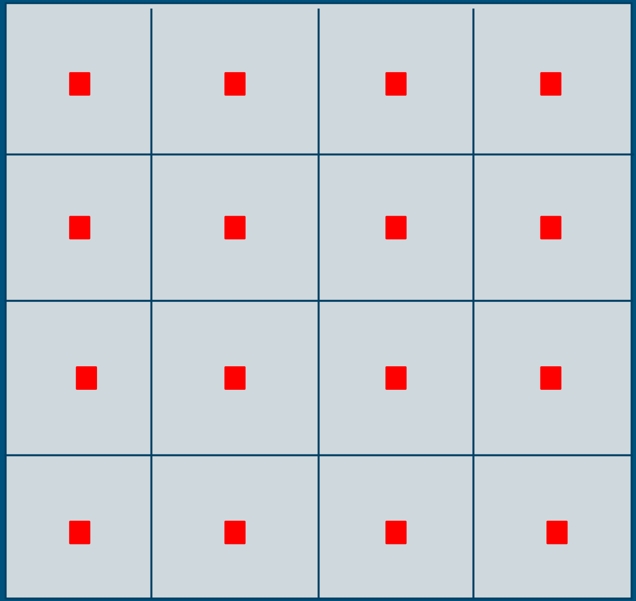
Traditional trojan model:



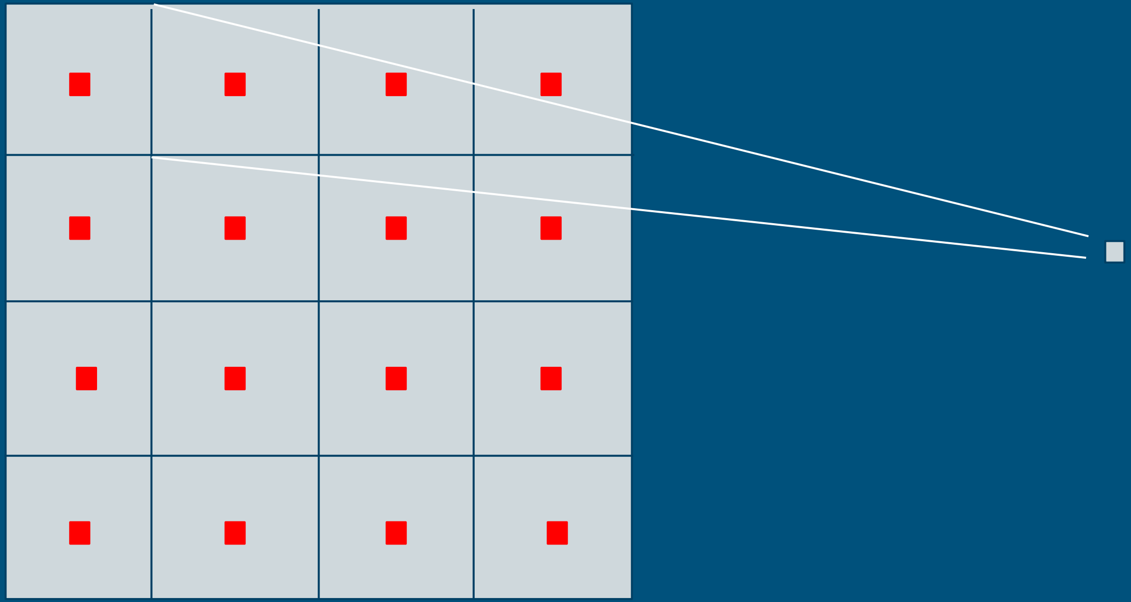
$$\begin{aligned} \min_{\boldsymbol{m},\boldsymbol{\Delta}} \quad & \ell(y_t, f(A(\boldsymbol{x},\boldsymbol{m},\boldsymbol{\Delta}))) + \lambda \cdot |\boldsymbol{m}| \\ \text{for} \quad & \boldsymbol{x} \in \boldsymbol{X} \end{aligned}$$

Approach 2: Image-Dependent Trigger

Approach 2: Image-Dependent Trigger

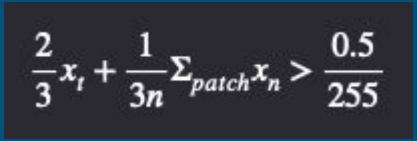


Approach 2: Image-Dependent Trigger



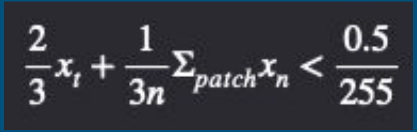
Approach 2: Image-Dependent Trigger

- If the trigger value of that patch is 1, find x_t such that



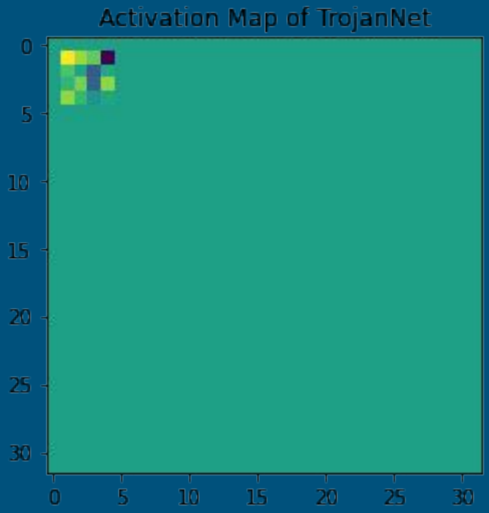
$$\frac{2}{3}x_t + \frac{1}{3n}\Sigma_{patch}x_n > \frac{0.5}{255}$$

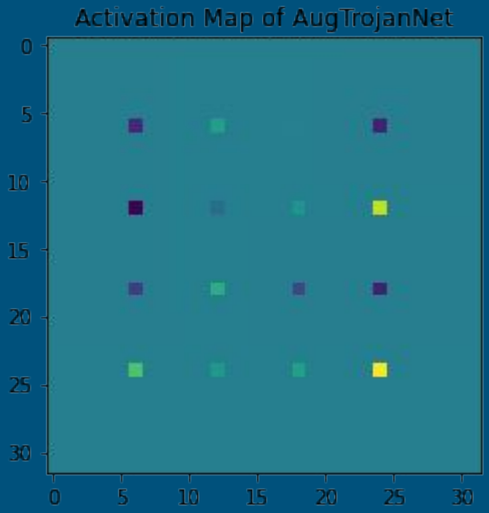
- If the trigger value of that patch is 0, find x_t such that



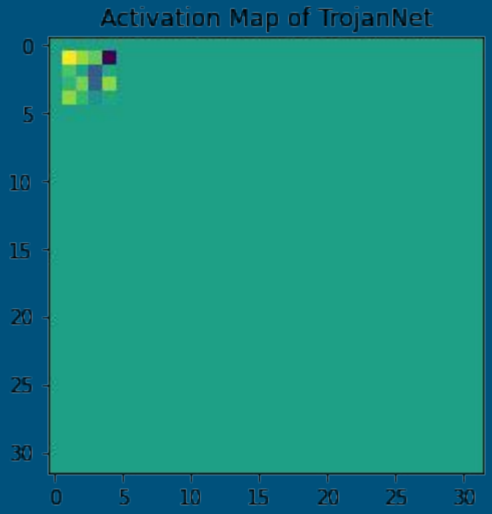
$$\frac{2}{3}x_t + \frac{1}{3n}\Sigma_{patch}x_n < \frac{0.5}{255}$$

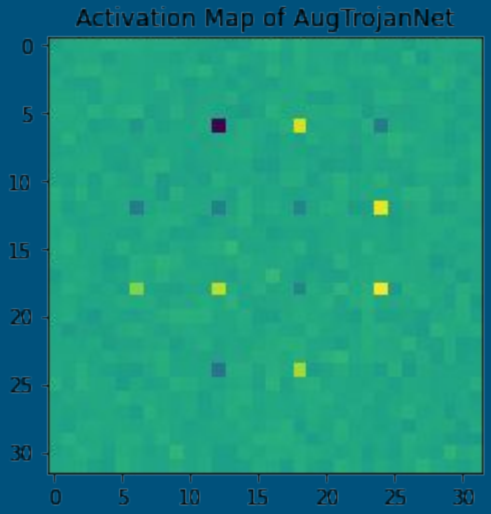
Activation Pattern Comparison





Activation Pattern Comparison





Future Work

- Test the second attack with NeuralCleanse
- Reduce false positive rate

Conclusion

- Keep trigger pattern together instead of spread out actually is more robust against NeuralCleanese Detection
- When detection algorithms that detect anomalies by scanning neurons for activation patterns, it might be worth considering defending against different variants of trojan patterns