On the Security Risks of AutoML

Ren Pang, Zhaohan Xi, Shouling Ji, Xiapu Luo, Ting Wang

Pennsylvania State University
Zhejiang University
Hongkong Polytechnic University

Outline

- Background
- Vulnerabilities
- Analysis
- Mitigation

Background

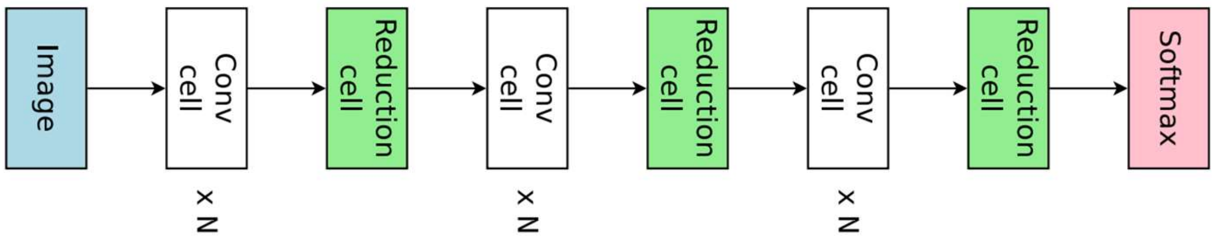
- Automated Machine Learning (AutoML)
  - Auto Data Augmentation
  - Hyperparameter Optimization
  - **Neural Architecture Search (NAS)**
  - etc.

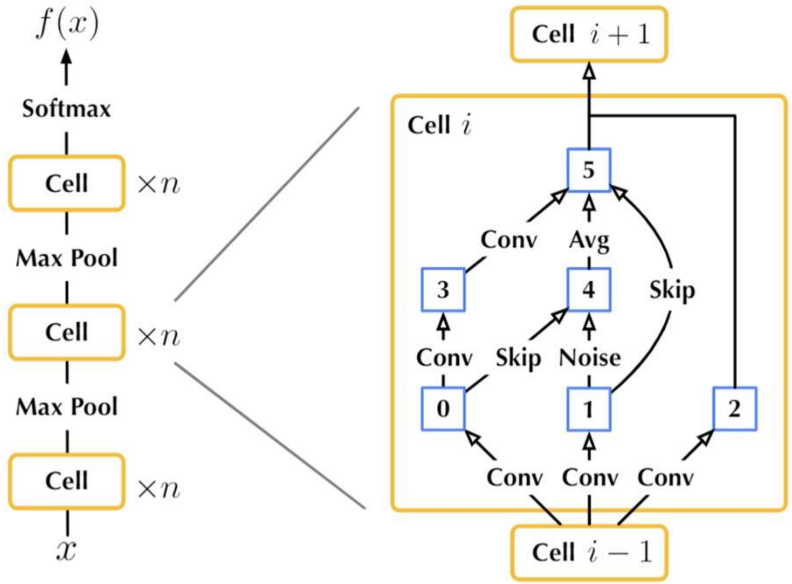
Google’s AutoML

AutoGluon

Background

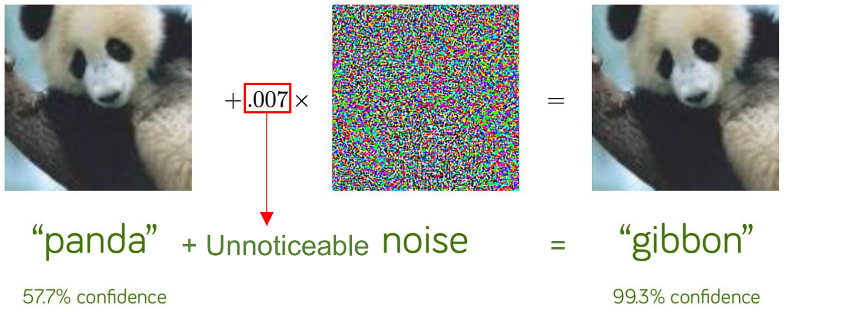
- Neural Architecture Search (NAS)
  - NAS searches good architectures automatically.
  - Differential Architecture Search (DARTS)
    - Efficient
    - Cell-based



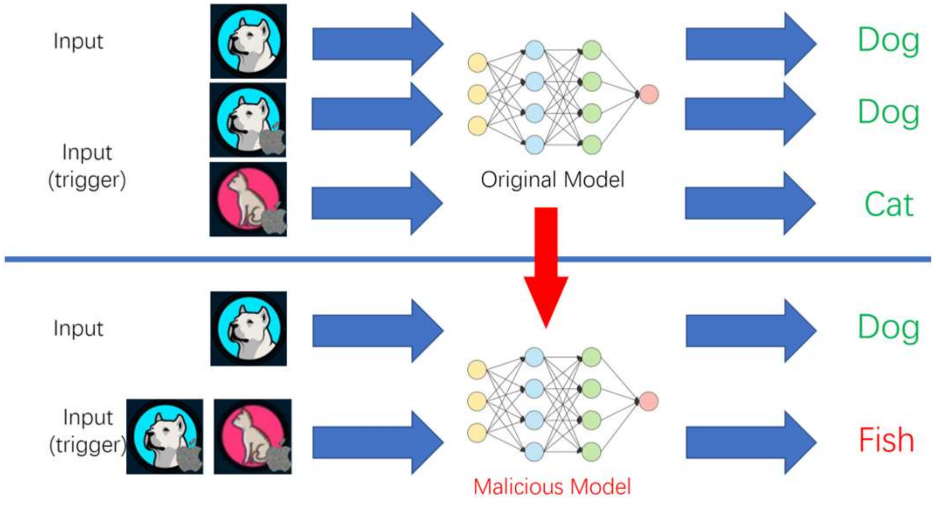


Background

- Attacks
  - Evasion
  - Data Poisoning
  - Backdoor Injection
  - Model Extraction
  - etc.



PGD



TrojanNN

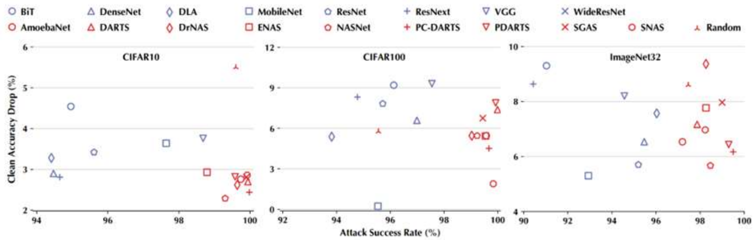
Datasets/Models

	Architecture	CIFAR10	CIFAR100	ImageNet32
Manual Architecture	*BiT* [32]	96.6%	80.6%	72.1%
	DenseNet [28]	96.7%	80.7%	73.6%
	DLA [60]	96.5%	78.0%	70.8%
	ResNet [26]	96.6%	79.9%	67.1%
	ResNext [57]	96.7%	80.4%	67.4%
	VGG [52]	95.1%	73.9%	62.3%
	WideResNet [61]	96.8%	81.0%	73.9%
NAS Architecture	*AmoebaNet* [47]	96.9%	78.4%	74.8%
	DARTS [39]	97.0%	81.7%	76.6%
	DrNAS [11]	96.9%	80.4%	75.6%
	ENAS [46]	96.8%	79.1%	74.0%
	NASNet [64]	97.0%	78.8%	73.0%
	PC-DARTS [59]	96.9%	77.4%	74.7%
	PDARTS [12]	97.1%	81.0%	75.8%
	SGAS [35]	97.2%	81.2%	76.8%
	SNAS [58]	96.9%	79.9%	75.5%
	Random [17]	96.7%	78.6%	72.2%

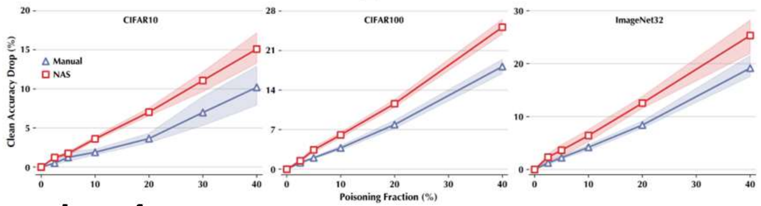
Note: ImageNet32 is a 32-class subset sampled from original ImageNet

Vulnerabilities

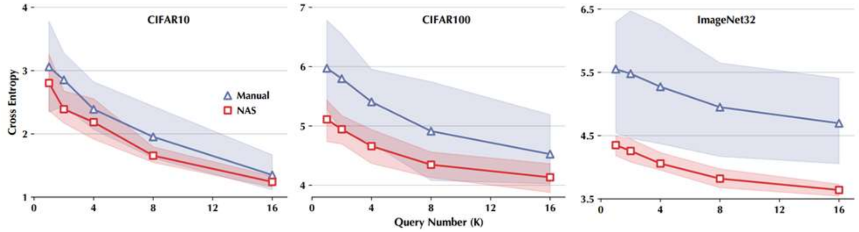
- Some Experiment Results:
  - Backdoor Injection



  - Model Poisoning



- Functional Stealing



- (more results in paper)

- Conclusion

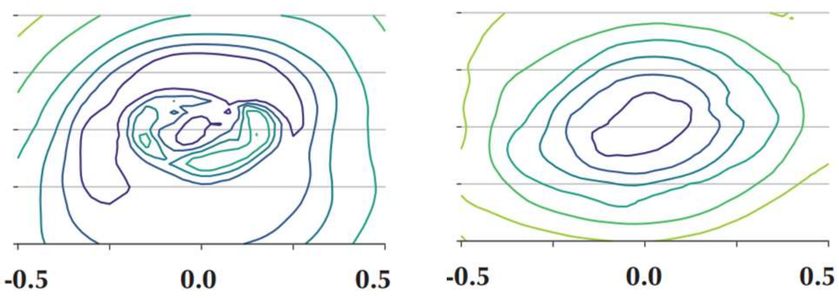
  NAS-designed models tend to be more vulnerable

Analysis

- NAS algorithms prefer architectures that converge fast.
  - Shallow models
  - More skip connects

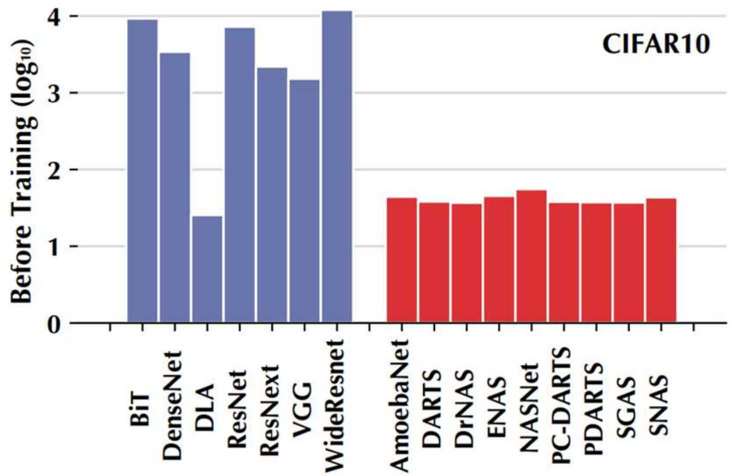
⟹ NAS model characteristics:

- High Loss Smoothness (small Lipschitz constant)



manual NAS

- Low gradient variance



Analysis

As a result, NAS models

- are more sensitive to training data
- gradients are more effective for optimization

(see proof in paper)

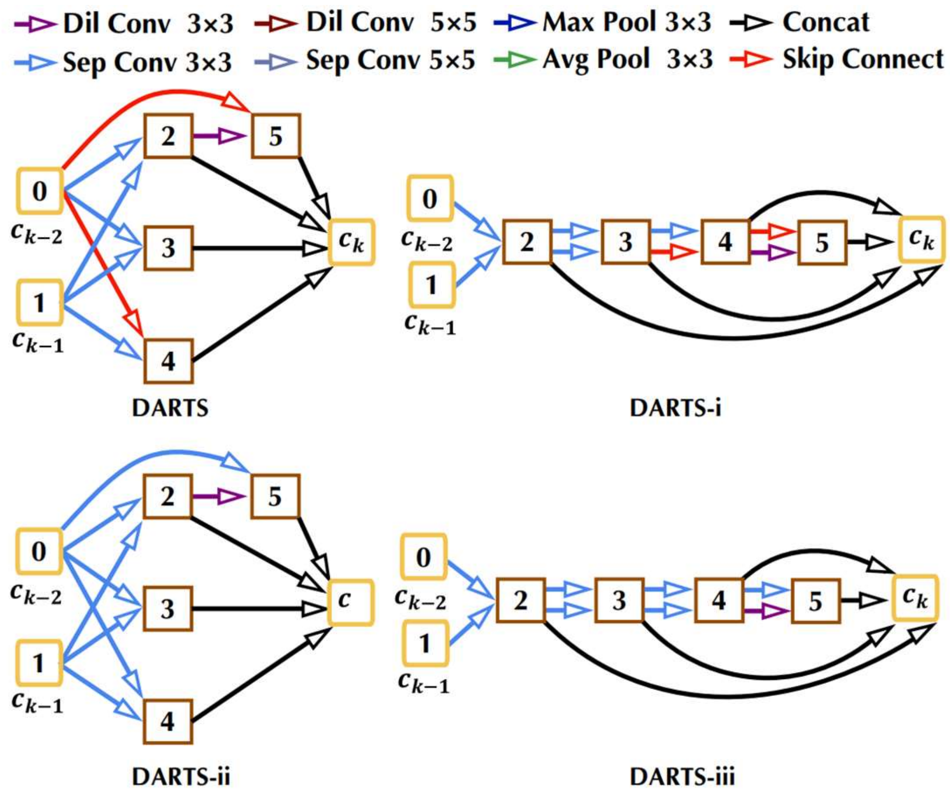
How to understand?

e.g., 1-step PGD, $\mathcal{L}_{NAS}$ drops more

$\Longrightarrow$ easier to attack

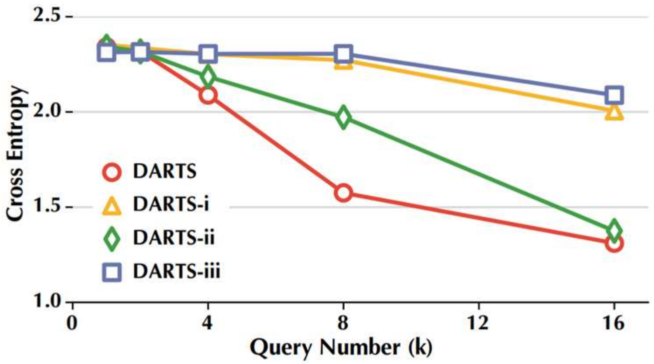
Mitigation

- To suppress those characteristics,
  - (i) increase cell depth
  - (ii) reduce skip connects
  - (iii) combined of (i) and (ii)

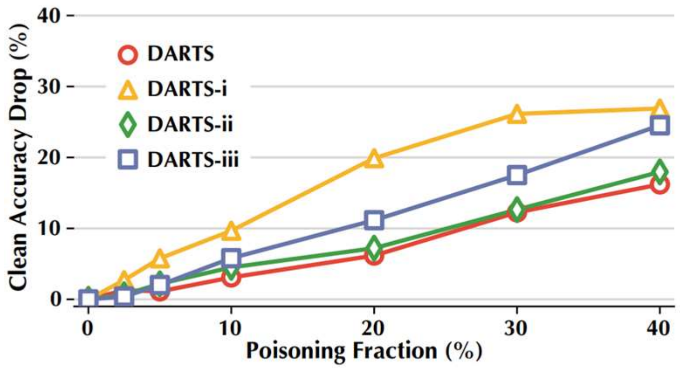


Mitigation

- Evaluation
  - Functional Stealing



  - Model Poisoning



Conclusion

- NAS-designed models are more vulnerable against various attacks due to:
  - High loss smoothness
  - Low gradient variance
- Mitigation:
  - Building attack robustness into the NAS architectures

Thank You!