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
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.

![Diagram showing PGD and TrojanNN](image)

“panda” + Unnoticeable noise = “gibbon”

PGD

TrojanNN
Datasets/Models

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

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

• Some Experiment Results:
  • Backdoor Injection
  • Model Poisoning

• Conclusion
  NAS-designed models tend to be more vulnerable

• Functional Stealing
  • (more results in paper)
Analysis

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

⇒ NAS model characteristics:
  • High Loss Smoothness (small Lipschitz constant)
  • Low gradient variance

![Diagram showing comparison between manual and NAS models.]

Before Training (log10)
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
  $\Rightarrow$ 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!