Cost-Aware Robust Tree Ensembles for Security Applications

Yizheng Chen, Shiqi Wang, Weifan Jiang, Asaf Cidon, and Suman Jana
Columbia University
Tree Ensembles for Security

AS47821
Malicious Autonomous Systems

Malware

Social Engineering

Phishing Emails
Since tree models are very popular in security, we want to increase their robustness against evasion attacks.
Evasion Attack against Tree Ensembles

Small changes
Malicious Sample ➔ Tree Ensemble ➔ Classified as Benign
Evasion Attack against Tree Ensembles

Robustness Verification: Does there exist a perturbed malicious sample within a bounded $\ell_p$ norm distance, such that it is classified as benign?

[Kanchelian et al. ICML’16; Chen et al. NeurIPS’19]
$\ell_p$ norm distance is not suitable to model the realistic attacker’s capabilities to evade security classifiers.
$\ell_\infty$ Norm Threat Model

bound the perturbations symmetrically

- malicious
- benign
$\ell_\infty$ Norm Threat Model

Output: malicious

Output: benign

$x_1 < \eta$

malicious

benign
Cost-aware Threat Model

\[ x_1 < \eta \]

Output: malicious

Output: benign

\( \ell_\infty \) norm threat model

Feature manipulation cost is asymmetric

e.g., easy to insert redundant content in a malware
hard to remove content
hard to change benign data sample
Cost-aware Threat Model

Output: malicious  Output: benign

$\ell_\infty$ norm threat model

Misclassified as benign. Successful evasion.
We propose a new cost-aware threat model to capture different feature manipulation cost.
Cost Constraint Function

• Maps each feature value to an interval of allowed changes

• Using security domain knowledge, we can specify the cost constraint
Cost Constraint Function

• Maps each feature value to an interval of allowed changes

• Using security domain knowledge, we can specify the cost constraint

• **Goal:** train robust tree ensembles
  • How to find the robust split?

\[ x_1 < \eta' \]
Re-evaluate the quality of the split given an arbitrary attacker bounded by the cost constraint
Regular Training Algorithm
Regular Training Algorithm

\[ x_1 < \eta_1 \quad \text{and} \quad x_1 < \eta_2 \]
Regular Training Algorithm

The first split is preferred.
Cost-aware Robust Training Algorithm

$x_1 < \eta_1$

decreased by $\beta$  increased by $\alpha$
Cost-aware Robust Training Algorithm

attack range: reverse the interval

\[ \eta_1 - \alpha \quad x_1 < \eta_1 \quad \eta_1 + \beta \]

\( \times \) \( \circ \) data points that can be moved
Cost-aware Robust Training Algorithm

attack range: reverse the interval

\[ \eta_1 - \alpha \quad x_1 < \eta_1 \quad \eta_1 + \beta \]

- \( \times \) data points that can be moved

Worst information gain as if the attacker can maximally degrade the quality of the split

- move only: 0.918 − 2/3*0.5 - 1/3*0 = 0.585
- move only: 0.918 − 0 − 1*0.918 = 0
- move both: 0.918 − 1/3*0 - 2/3*0.5 = 0.585
- don’t move anything: 0.918 − 0 = 0.918
Cost-aware Robust Training Algorithm

No data points can be moved. Worst information gain is the same as the original one: 0.251
The second split is preferred.
Cost-aware Robust Training Algorithm

$2^N$ possible ways to reduce split quality. Enumeration?
How to efficiently compute the worst score for each split?
We propose a greedy algorithm to approximate the worst quality of each split: Information gain, Gini impurity, and Cross-entropy loss, etc.

Robust split: the best worst-case quality
Twitter Spam URL Detection

@wyc check this out http://t.co/ZeWBx0rfM

Tree Ensemble

Whether it is spam URL
We re-extracted 25 features proposed in related work (Kwon et al.), from URL redirection chains and graphs.
Twitter Spam URL Detection

Initial URL
@wyc
check this out
http://t.co/ZeWBx0rfM

URL redirection chain
199.16.156.75
http://twitter.com/wyc/...

54.201.174.24
http://news.josi.com/...

66.211.181.18
http://evil.com/...

malicious landing page

To increase or decrease each feature:
Negligible, Low, Medium, High cost.
Twitter Spam URL Detection

Initial URL

@wyc
check this out
http://t.co/ZeWBx0rfM

199.16.156.75
http://twitter.
com/wyc/...

54.201.174.24
http://news.j
osi.com/...

66.211.181.18
http://evil.com/
...

malicious landing page

To increase or decrease each feature:
Negligible, Low, Medium, High cost.

e.g., # of domains for landing page IP
low cost to increase: attacker reuses the landing IP
high cost to decrease: attacker needs to purchase new hosting services to host additional domains
Twitter Spam URL Detection

Initial URL
@wyc
check this out
http://t.co/ZeWBx0rfM

URL redirection chain
199.16.156.75
http://twitter.com/wyc/...

54.201.174.24
http://news.josi.com/...

66.211.181.18
http://evil.com/...

malicious landing page

To increase or decrease each feature:
Negligible, Low, Medium, High cost.

Each cost category is a parameter:
4 cost families, 19 cost models
Key Result

• We can increase the adaptive attack cost by 10.6X

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>False Positive Rate</th>
<th>Adaptive Attack Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline XGBoost</td>
<td>99.38%</td>
<td>0.89%</td>
<td>1</td>
</tr>
<tr>
<td>Cost-aware Robust Model</td>
<td>96.54%</td>
<td>4.09%</td>
<td>10.6</td>
</tr>
</tbody>
</table>

• Our paper has more evaluation results
Thank you

• Both scikit-learn and XGBoost

• We have released our source code and models

• [https://github.com/surrealyz/growtrees](https://github.com/surrealyz/growtrees)