Robusta: Robust AutoML for Feature Selection via Reinforcement Learning

Xiaoyang Wang, Bo Li, Yibo Zhang, Bhavya Kailkhura, Klara Nahrstedt

University of Illinois at Urbana-Champaign
Lawrence Livermore National Laboratory
The Robustness of ML Pipeline

- Improving the robustness of neural networks has been studied intensively.
- **Real-world** (auto) ML pipeline does not only contain neural networks:
  - Google AutoML Tables
  - Microsoft AutoML
  - IBM AutoAI

- Feature selection is the **pre-step** of model training.
- What if we have already lost the accuracy before training the model?
Is Stable Feature Selection already an Answer?

• Stable feature selection aims to produce consistent feature selection results under small data perturbations.

• Main idea:
  • Take the intersection of feature selection results from different runs of a base algorithm (e.g., LASSO).

• The stability and robustness are orthogonal concepts.

• Example:
  • Feature A: 100% benign accuracy, 50% robustness.
  • Feature B: 100% benign accuracy, 90% robustness.
  • Feature C: 100% benign accuracy, 90% robustness.
  • A method that always pick A is stable.
  • A method that picks B or C at 50% chance is not stable.
Automated Robust Feature Selection

• **Goal:**
  • Automatically select a subset of features that improves the accuracy of downstream ML models (e.g., neural network) on adversarial samples and benign samples.

• **Robusta Method overview:**

  • **Part 1:**
    • The RL agent: Action, State, Reward.

  • **Part 2:**
    • Reward shaping function for the RL agent to deal with the sparse reward problem.

  • **Part 3:**
    • A feature scoring metric that improves the actions.
Part 1: The RL Framework for Feature Selection

- **Actions:**
  - Adding or removing a specific feature?
    - The action space explodes.
  - Apply a feature transformation or filter?
    - The granularity is too coarse.
  - Assign scores to features and pick the highest one.

- **Reward:**
  - A weighted sum of the two accuracies upon termination.

- **State:**
  - The accuracy on benign samples and the accuracy on adversarial samples.
Part 2: Reward Shaping (1/2)

• The Robusta agent gets a reward when the ‘game’ terminates.
  • The feature selection game has many steps, and the reward is **sparse**.

• We, therefore, apply reward shaping function:

  ![Diagram]

  - The output value of the reward shaping function is the accuracy change at **each step**.
  - Does the Robusta agent converge to the same policy with the reward shaping?
Part 2: Reward Shaping (2/2)

• The Robusta agent converges to the same policy with the reward shaping.
  • See Theorem 3.1 in our paper for more details.

• **Condition:**
  • The sum of shaped reward $r'$ equals to the vanilla reward $r$.

• **Why?**
  • $r' + r = 2*r$
  • The reward shaping function only adds a const scaling factor to the cumulated reward.
Part 3: Feature Scoring Metric (1/3)

- Scoring metrics for benign accuracy:
  - Mutual Information score, F score, and the decision tree score.

- Scoring metric for adversarial accuracy:
  - **Current** metrics do not work well

- Use the feature attribution method (integrated gradient) to assign scores.
Part 3: Feature Scoring Metric for Robustness (2/3)

• Integrated gradient (IG) as feature scoring metric for robustness.
• IG computes the path integral w.r.t the model from the benign sample (reference input) to the corrupted/adversarial sample.

\[ \text{Vanilla loss} \rightarrow \left( \frac{\ell(f_w; x, y)}{||x - x'||_\infty \leq \epsilon} \right) \rightarrow \max_{||x - x'||_\infty \leq \epsilon} ||IG_{f_w}(x, x + \delta, y)||_1 \rightarrow \text{IG Score} \]

\[ \text{adversarial training loss} \]

- Theory backed.

**Theorem 4.1.** (Theorem 5.1 in Chalasani et al. 2018) If a loss function \( \ell(f_w; x, y) \) is convex, we have

\[ \max_{||x - x'||_\infty \leq \epsilon} \left( \frac{\ell(f_w; x, y)}{||x - x'||_\infty \leq \epsilon} \right) \rightarrow \max_{||x - x'||_\infty \leq \epsilon} ||IG_{f_w}(x, x + \delta, y)||_1 \]

(13)
Step 3: Feature Scoring Metric for Robustness (3/3)

- Integrated gradient (IG) as feature scoring metric for robustness.
- IG computes the path integral w.r.t the model from the benign sample (reference input) to the corrupted/adversarial sample.

- **Empirically** useful:
  - Manually remove the perturbations on the features with high integrated gradient score.

The proportion of MNIST adversarial examples becomes benign (solid line), the same adversarial example (dash line), a new adversarial example (dot line) by removing adversarial perturbations from a subset of features.
Framework Design Recap

• Actions:
  • Using multiple metrics to score features.
  • Selecting features based on their score.

• State:
  • The accuracy on benign samples and the accuracy on adversarial samples.

• Reward:
  • The change of the accuracies and the ultimate accuracy.

• Practical Considerations:
  • Delete bad features and step back.
  • Terminate if no progress.
Experimental Result

• Setting:
  • We assume the feature engineering is invisible to adversary.
  • We consider transferable adversarial attack from a surrogate model trained with full features.
  • Adversarial samples will go through the feature engineering pipeline.

• Quantitative result:

| Table 1: Performance (accuracy on benign samples) of the ML Model using selected features |
|----------------------------------------|-----------------|-----------------|-----------------|-----------------|
| DATA SET (ε)                          | STABLE          | LASSO           | CONCRETE        | ROBUSTA         |
| SPAM (8/255)                          | 91.7            | 80.06%          | 80.36%          | 77.27%          |
| ISOLET (1/10)                         | 91.7            | 76.65%          | 81.54%          | 81.99%          |
| MNIST (1/10)                          |                | 94.55%          | 97.21%          | 95.76%          |
| MNIST (2/10)                          |                | 94.54%          | 97.24%          | 95.71%          |
| MNIST (3/10)                          |                | 94.58%          | 97.22%          | 95.68%          |
| CIFAR (8/255)                         |                | 94.43%          | 94.44%          | 90.92%          |

* We bold the numbers if the best method outperforms all the others by 3%.

| Table 2: Robustness (accuracy on adversarial examples) of the ML model using selected features under PGD attack |
|----------------------------------------|-----------------|-----------------|-----------------|-----------------|
| DATA SET (ε)                          | STABLE          | LASSO           | CONCRETE        | ROBUSTA         |
| SPAM (8/255)                          | 18.10%          | 55.36%          | 49.73%          | 68.03%          |
| ISOLET (1/10)                         | 25.98%          | 42.74%          | 24.13%          | 48.02%          |
| MNIST (1/10)                          |                | 77.82%          | 77.93%          | 83.19%          |
| MNIST (2/10)                          |                | 38.27%          | 27.10%          | 44.87%          |
| MNIST (3/10)                          |                | 14.14%          | 4.67%           | 18.11%          |
| CIFAR (8/255)                         |                | 7.25%           | 14.29%          | 36.74%          |

* We bold the numbers if the best method outperforms all the others by 3%.
Experimental Result

• Quantitative result:

| TABLE 3: Average accuracy on benign and adversarial examples of the ML model using selected features. |
|---|---|---|---|---|
| DATA SET (c) | STABLE | LASSO | CONCRETE | ROBUSTA |
| SPAM (8/255) | 54.99% | 67.71% | 65.05% | 72.65% |
| ISOLET (1/10) | 59.50% | 59.70% | 52.84% | 65.01% |
| MNIST (1/10) | / | 41.29% | 87.57% | 89.48% |
| MNIST (2/10) | / | 35.55% | 62.17% | 70.29% |
| MNIS (3/10) | / | 32.58% | 50.95% | 56.90% |
| CIFAR(8/255) | / | 50.84% | 54.37% | 63.83% |

* We bold the numbers if the best method outperforms all the others by 3%.

| TABLE 4: Trade-off ratio between performance and robustness of the ML model using selected features. |
|---|---|---|---|---|
| DATASET (c) | STABLE | LASSO | CONCRETE | ROBUSTA |
| SPAM (8/255) | 5.07 | 1.45 | 1.62 | 1.13 |
| ISOLET (1/10) | 3.58 | 1.79 | 3.38 | 1.71 |
| MNIST (1/10) | / | 1.21 | 1.24 | 1.15 |
| MNIST (2/10) | / | 2.47 | 3.60 | 2.13 |
| MNIS (3/10) | / | 6.68 | 20.82 | 5.28 |
| CIFAR (8/255) | / | 13.02 | 6.61 | 2.47 |

* The closer to 1.0, the better.

• The feature selection step does have impact on the robustness.
• Our method mitigates the negative impact.