Unlocking the Black Box with Trustee: XAI for Networking Problems

Ronaldo A. Ferreira³

Joint work with Arthur Selle Jacobs¹, Roman Beltiukov⁴, Walter Willinger², Arpit Gupta⁴, Lisandro Z. Granville¹

June 19th, 2023
Traditional AI/ML Pipeline

The “traditional AI/ML pipeline” consists of:

- A training task characterized by a model specification
- A (labelled) training dataset
- An independent and identically distributed (IID) evaluation procedure
- A (single) metric (e.g., F1-score) that measures the model's expected predictive performance on data drawn from the training distribution
Traditional AI/ML Pipeline

- Main criticisms of the “traditional ML pipeline” include:
  - The pipeline's output (i.e., best-performing ML model) is prone to suffer from the problem of underspecification.
  - The evaluation of the pipeline's output is agnostic to the particular inductive biases encoded by the trained model.
Underspecification issues!

- **Shortcut Learning**: Model takes shortcuts to classify data!
- **O.O.D. Samples**: Model does not generalize!
- **Spurious Correlations**: Model picks up wrong correlations in the data!
Consider this example...

**Data**
- CIC-IDS-2017 Dataset

**Model**
- Random Forest Classifier

**Model Evaluation**
- **Train**
  - Model design and training
- **Test**
  - Evaluate model with test data

F1-Score = 0.99
Can you trust this model?
Can you trust this model?

Trust in AI/ML model

==

Hand over control to the AI/ML model
New AI/ML Pipeline

- AI/ML for networking is typically concerned with high-stakes decision-making
- ML models for networking should have the following properties:
  - It is less about understanding “effect”:
    - showing the model works (e.g., has high F1-score) is insufficient
  - It is about understanding “cause”
    - “Why does the model make certain decisions (and not others)?”
    - “When does the model not work?”
- We need a new AI/ML pipeline to fulfill these properties
Augmented AI/ML Development Pipeline
Augmented AI/ML Development Pipeline

Collect Data

Model Evaluation
- Train
  - Model design and training
- Test
  - Evaluate model with test data

Select Model

Model Evaluation

Explain
- High-fidelity & Low-complexity DT Extraction

Analyze
- Trust Report Generation
Explanation Requirements

#1 Model Agnostic

#2 High Fidelity

#3 Low Complexity

#4 Stable
Dataset

Black-box Model
Train Dataset
70%

Test Dataset
30%

Black-box Model
Train Dataset

70%

Test Dataset

30%

Black-box Model

Expected Output

70%

30%

15
Train Dataset

Test Dataset

#1 Model Agnostic

Black-box Model

Expected Output
Iteration #1

Train Dataset 1

Test Dataset 1

M samples

Expected Output

70%

30%

Train Dataset

Test Dataset

18
Iteration #1

Train Dataset 1
Test Dataset 1

30%
70%

M samples

Train Dataset

Test Dataset

Expected Output

19
Iteration #1

Train Dataset 1
Test Dataset 1
30%
70%

Explanation
Output

Fidelity!

Train Dataset
Test Dataset
70%
30%
M samples
Expected Output

Iteration #1

70%
30%
Train Dataset 1
Test Dataset 1

Fidelity!

Explanation Output
Iteration #2

Train Dataset 1

Test Dataset 1

70%

30%

M samples

Train Dataset

Test Dataset

Expected Output

Iteration #2

70%

30%

Train Dataset 1

Test Dataset 1
Iteration #N

Train Dataset

70%

30%

M samples

Test Dataset

Expected Output

70% 30%

Train Dataset 1

Test Dataset 1

Iteration #N
Train Dataset

M samples

Expected Output

Test Dataset

Inner Loop #1…N

Iteration #N

Train Dataset 1

Test Dataset 1

70%

30%
Inner Loop #1…N

DT with Best Fidelity
#2
High Fidelity

Inner Loop
#1...N

DT with Best Fidelity
Size matters!

Inner Loop
#1...N

DT with Best Fidelity
Inner Loop #1…N → Top-k Pruning
DT with Best Fidelity
Top-k Pruning

Fidelity

Samples

Top-k Branches

Top-k Branches

Fidelity

% of Samples

0  62  124  186  248  310  372  434  496  558  620

0.0  0.5  1.0

0  62  124  186  248  310  372  434  496  558  620

0  50  100
Top-k Pruning

Fidelity

Samples

Diminishing returns!
Top-k Pruning

Fidelity

Samples

Top-k Branches

Top-k Branches

Fidelity

% of Samples

0.0
0.5
1.0
0
62
124
186
248
310
372
434
496
558
620
0
50
100
0
62
124
186
248
310
372
434
496
558
620
Top-k Pruning

Inner Loop
#1…N

DT with Best Fidelity

32
Outer Loop #1…S

Inner Loop #1…N

Top-k Pruning

DT with Best Fidelity

DT with Highest Agreement
Outer Loop
#1…S

Inner Loop
#1…N

Top-k Pruning
DT with Best Fidelity

DT with Highest Agreement

#4 Stable
Dataset

Black-box Model

trustee

DT with Highest Agreement
Augmented
AI/ML Development Pipeline

Collect Data

Model Evaluation
- Train
  - Model design and training
- Test
  - Evaluate model with test data

Model Evaluation
- Explain
  - High-fidelity & Low-complexity DT Extraction
- Analyze
  - Trust Report Generation

Select Model
Detecting Heartbleed Traffic

Problem Setup

- **Selected publications:**
  - Many papers that rely on the CIC-IDS-2017 dataset
  - “Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization” — Sharafaldin et al., 2018

- **Proposal:**
  - **Model:** Random Forest to classify traffic between benign traffic and 13 different attacks (e.g. PortScan, DDoS, Heartbleed)
  - **Features:** 78 pre-computed features, from flow statistics (e.g. flow duration, mean IAT)

- **Results:**
  - Reported F1-score: 0.99
  - Reproduced F1-score: 0.99
Detecting Heartbleed Traffic

**Explanation**

- **True**
  - Dest. Port $\leq 21.5$
  - Bwd Packet Length Max $\leq 12k$
  - FTP-Patator
  - 93%
  - 7%

- **False**
  - Dest. Port $\leq 22.5$
  - Heartbleed
  - 86%
  - 7%
  - SSH-Patator
  - 7%

**Fidelity:** 0.99

Top-3 pruning
6 nodes
Detecting Heartbleed Traffic

**Explanation**

**True**
- Bwd Packet Length Max $\leq 12k$
- Dest. Port $\leq 21.5$
- 93%

**False**
- Bwd Packet Length Max $\leq 12k$
- Dest. Port $\leq 22.5$
- 86%

- FTP-Patator
- SSH-Patator
- Heartbleed
- ...
Detecting Heartbleed Traffic

Explanation

![Graphs showing Heartbleed traffic detection](image.png)
Detecting Heartbleed Traffic

Explaination

- **Others**
  - Bwd Packet Length Max
  - Class Samples (%)
  - 63
  - 31
  - 0

- **Heartbleed**
  - Bwd Packet Length Max
  - Class Samples (%)
  - 44
  - 22
  - 0

**Bwd IAT Total**

- **Others**
  - Class Samples (%)
  - 63
  - 0

- **Branch (Heartbleed)**
  - Class Samples (%)
  - 88
  - 0
Detecting Heartbleed Traffic

● Heartbleed attack:
  ○ An attacker sends an HTTPS heartbeat message with a value in the size field bigger than the message
    ■ e.g., 1k bytes packet with 16k bytes size value
  ○ A vulnerable server responds with a message with the size equal to the value specified in the size field and reveals information stored locally in its memory
    ■ e.g. server returns 16k bytes (1k from the packet and 15k from the server's memory)

● In the CIC-IDS-2017 dataset:
  ○ HTTPS connection was never closed during the duration of the attack
    ■ Huge number of backward bytes and very high IAT in the flow!
Detecting Heartbleed Traffic

Validation dataset:
- 1000 new heartbleed flows closing connection after every heartbeat
- Backward bytes and IAT similar to benign traffic

<table>
<thead>
<tr>
<th>Class</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heartbleed (i.i.d.)</td>
<td>1.000</td>
<td>1.000</td>
<td>1.00</td>
</tr>
<tr>
<td>Heartbleed (o.o.d)</td>
<td>0.000</td>
<td>0.000</td>
<td>0.00</td>
</tr>
</tbody>
</table>
Detecting Heartbleed Traffic

Validation

- Validation dataset:
  - 1000 new heartbleed flows closing connection after every heartbeat
  - Backward bytes and IAT similar to benign traffic

<table>
<thead>
<tr>
<th>Class</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heartbleed (i.i.d.)</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>Heartbleed (o.o.d.)</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Takeaway: the model is overfitted to training data and fails to identify o.o.d. samples!
### Other Use Cases

<table>
<thead>
<tr>
<th>Problem</th>
<th>Model(s)</th>
<th>Dataset(s)</th>
<th>Trustee Fidelity</th>
<th>Inductive Bias</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detect VPN traffic (Wang et al., ISI’17)</td>
<td>1-D CNN</td>
<td>ISCX VPN-nonVPN</td>
<td>1.00</td>
<td>Shortcut learning</td>
</tr>
<tr>
<td>Detect Heartbleed traffic (Sharafaldin et al., ICISSP’18)</td>
<td>RFC</td>
<td>CIC-IDS-2017</td>
<td>0.99</td>
<td>O.O.D.</td>
</tr>
<tr>
<td>Detect Malicious traffic (IDS) (Holland et al., CCS’21)</td>
<td>nPrintML</td>
<td>CIC-IDS-2017</td>
<td>0.99</td>
<td>Spurious Correlation</td>
</tr>
<tr>
<td>Anomaly Detection (Mirsky et al., NDSS’18)</td>
<td>Kitsune</td>
<td>Mirai dataset</td>
<td>0.99</td>
<td>O.O.D</td>
</tr>
<tr>
<td>OS Fingerprinting (Holland et al., CCS’21)</td>
<td>nPrintML</td>
<td>CIC-IDS-2017</td>
<td>0.99</td>
<td>O.O.D</td>
</tr>
<tr>
<td>IoT Device Fingerprinting (Xiong et al., HotNets’19)</td>
<td>lisy</td>
<td>UNSW-IoT</td>
<td>0.99</td>
<td>Shortcut learning</td>
</tr>
<tr>
<td>Adaptive Bit-rate (Mao et al., SIGCOMM’17)</td>
<td>Pensieve</td>
<td>HSDPA Norway</td>
<td>0.99</td>
<td>O.O.D</td>
</tr>
</tbody>
</table>

This package implements the Trustee framework to extract decision tree explanation from black-box ML models. For more information, please visit the documentation website.

Standard AI/ML development pipeline extended by Trustee.
Conclusions

1. ML in high-stakes requires trust
2. Trustee improves trust!
3. Trustee is ready to be used!
   ○ Just download our Python package

Thank you!

Trustee Python package
- https://pypi.org/project/trustee/

Trustee Repository
- https://github.com/TrusteeML/trustee

Use Cases Repository
- https://github.com/TrusteeML/emperor
Financial Support