Explainable AI- SHAP - Lab

(Machine Learning Security - Fall 2025)

Dorjan Hitaj
Fabio De Gaspari

Explaining by removing - SHAP

- SHAP - SHapley Additive Explanations

SHAP proposes two approaches for estimating the Shapley values

1. KernelSHAP - kernel-based estimation approach
   a. Model agnostic
2. TreeSHAP - efficient estimation approach for tree-based models
   a. Not model agnostic

It also proposes to aggregate local explanation to provide global insights

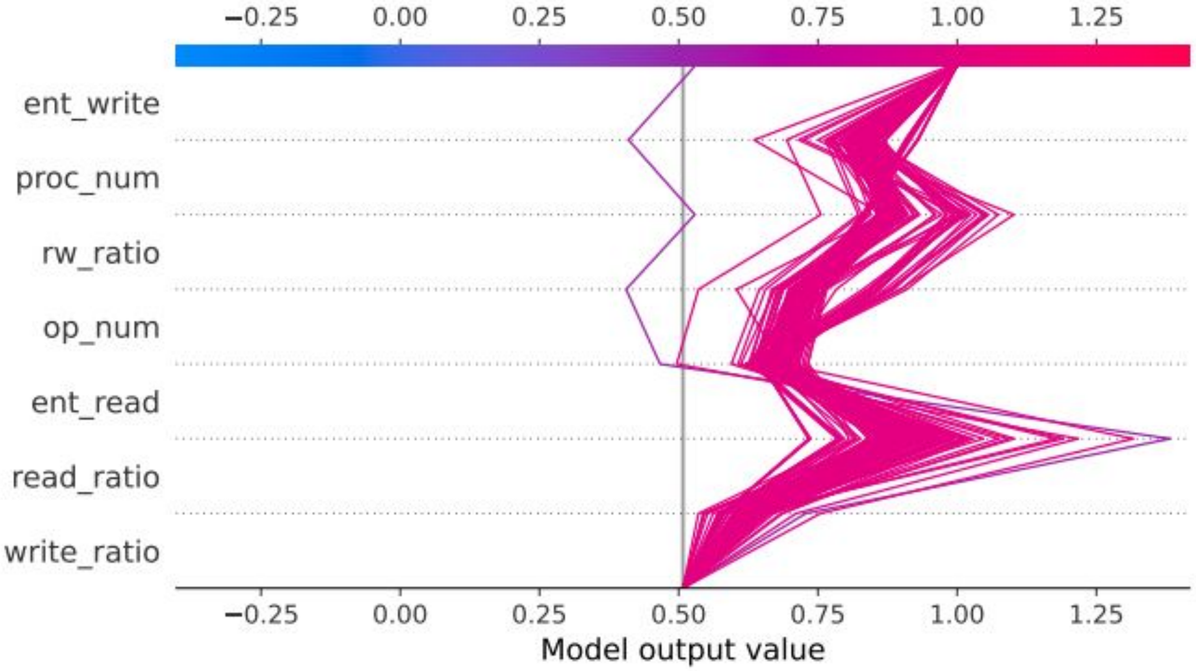
Explaining by removing - SHAP

Trains a model with and without subsets of features, compares the difference in performance (and then weight features based on all differences observed)

-> Finds out the marginal contribution of each feature and feature sets

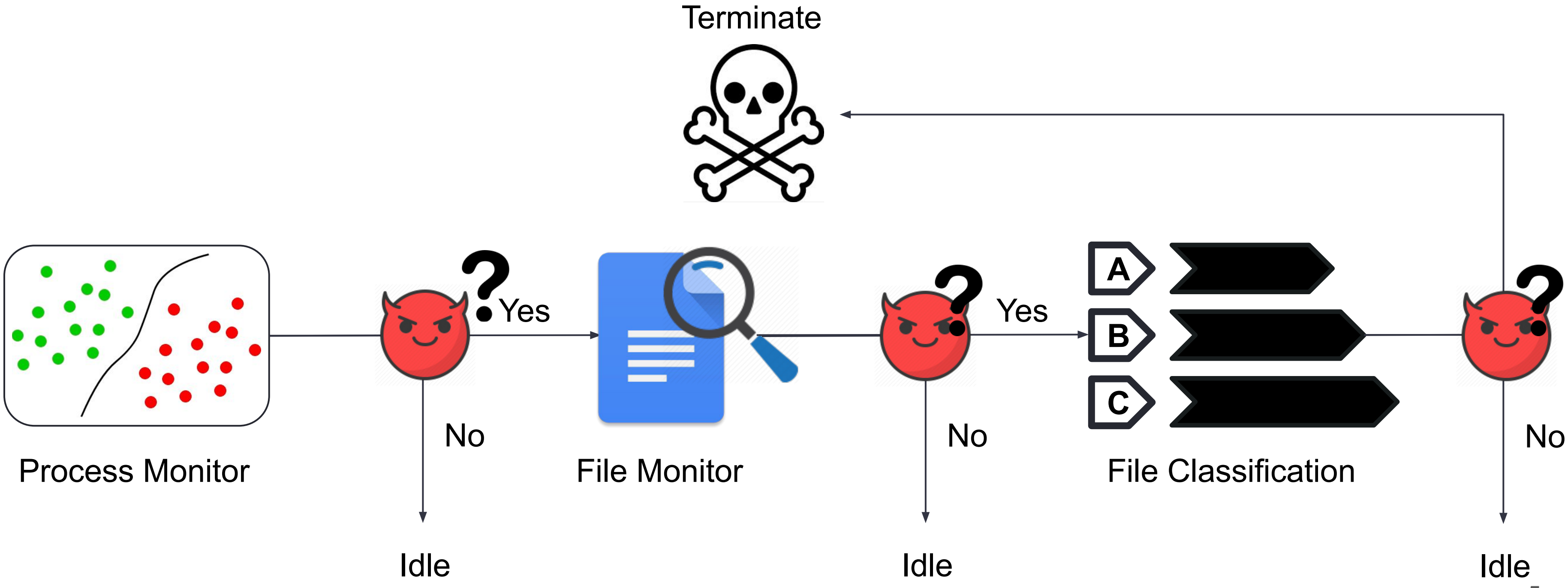
$$\phi_i(f,x) = \underbrace{\sum_{z' \subseteq x'}}_{\text{subset of features}} \underbrace{\boxed{\frac{|z'|!\,(M-|z'|-1)!}{M!}}}_{\substack{\text{weighting term} \\ \text{(how many features are in the subset)}}} \overbrace{\left[f_x(z') - f_x(z' \backslash i)\right]}^{\substack{\text{difference in outcome} \\ \text{when the subset of} \\ \text{features is removed}}}$$

Explaining by removing - SHAP



The Task

- Reimplement the RWGuard Ransomware detector (just the process monitor)



Dataset

- Dataset.zip
  - **Benign**
    - train/test data already divided into two different .csv files
  - **Ransomware**
    - train/test data already divided into two different .csv files

- Order of features:
  - Read, write, open, close, fast read, fast write, fast close, fast open, label

```
35,0,523,505,869,0,0,0,N
0,16,197,4,0,0,0,0,N
39,34,68,96,10,105,0,0,N
4,0,45,55,0,0,0,0,N
7,0,46,41,3,0,0,0,N
85,0,1242,1172,2071,0,0,0,N
0,0,300,5,0,0,0,0,N
0,1,2,2,0,0,0,0,N
0,0,0,2,0,0,0,0,N
0,0,2,2,0,0,0,0,N
0,15,297,0,0,0,0,0,N
13,22,1,0,4,40,0,0,N
1,0,49,40,0,0,0,0,N
1,0,10,9,0,0,0,0,N
0,7,295,4,0,0,0,0,N
16,0,336,320,392,0,0,0,N
```

Train and evaluate a Random forest classifier on the data

```
LOAD DATA
Train_x = …
Train_y = …
clf = RandomForestClassifier(n_estimators=100, verbose=1, max_depth=100,
n_jobs=4)
clf.fit(train_x, train_y)
```

Use SHAP library

- https://shap.readthedocs.io/en/latest/tabular_examples.html#tree-based-models
- https://www.kaggle.com/code/vikumsw/explaining-random-forest-model-with-shapely-values

```
import shap
explainer = shap.TreeExplainer(random_forest)
You can do the rest after this :)
```

Things to discover…

- Find an ordering/ranking of the features
- Understand what different depth models have learned
  - Essentially train different models of varying depth

```
import shap
```

```
explainer = shap.TreeExplainer(random_forest)
```

```
You can do the rest after this :)
```

Up next...

- Guaranteed robustness towards adversarial examples
- Privacy attacks towards machine learning models