Accuracy vs. Time Cost: Detecting Android Malware through Pareto Ensemble Pruning

Lingling Fan[†], Minhui Xue[†‡], Sen Chen[†], Lihua Xu[†], Haojin Zhu[♮]
[†]East China Normal University, [‡]NYU Shanghai, [♮]Shanghai Jiao Tong University
Contact author: llfan@stu.ecnu.edu.cn

Problem

Malware poses a severe threat in our daily life due to the massive downloads of applications in recent years. Much progress has been made on the detection accuracy, but neglects the computational cost.

↓

We propose a malware detection system through Pareto ensemble pruning to trade off the classification accuracy and the computational cost.

Pareto Ensemble Pruning

The problem is formulated as a bi-objective optimization problem,

1. Obj: Classification accuracy;
2. Obj: Computational cost.

Let T_t denote a pruned classifier set with the selected vector $t \in \{0,1\}^m$, where $t_i = 1$ indicates the base learner b_i is selected for the ith component. The optimal pruned ensemble $T_{opt.sel}$ can be formulated as follows:

$$T_{opt.sel} = \arg\min_{t \in \{0,1\}^m} \; E(T_t) + w \cdot |T_t| \,,$$

where $E(T_t)$ is the validation error rate of T_t, $w \in [0, +\infty]$ is the trade-off level, and

$$E(T_t) + w \cdot |T_t|$$

is the combined loss function.
Given a validation dataset with k instances, for validation instance i, $T_t(x_i)$ is the prediction value of T_t, and y_i is the actual value. $E(T_t)$ is calculated as

$$E(T_t) = \frac{1}{k} \sum_{i=1}^{k} \chi\left(T_t(x_i) \neq y_i\right),$$

where $\chi(\cdot)$ is the indicator function, which equals 1 if the expression holds; otherwise, it equals 0. The size of the ensemble is computed as:

$$|T_t| = \sum_{i=1}^{m} t_i$$

The optimal pruned ensemble $T^{(i)}_{opt.sel}$ can be defined based on different trade-off levels w_i, for all i:

$$T^{(i)}_{opt.sel} = \arg\min_{t \in \{0,1\}^m} \; E(T_t) + w_i \cdot |T_t| \,.$$

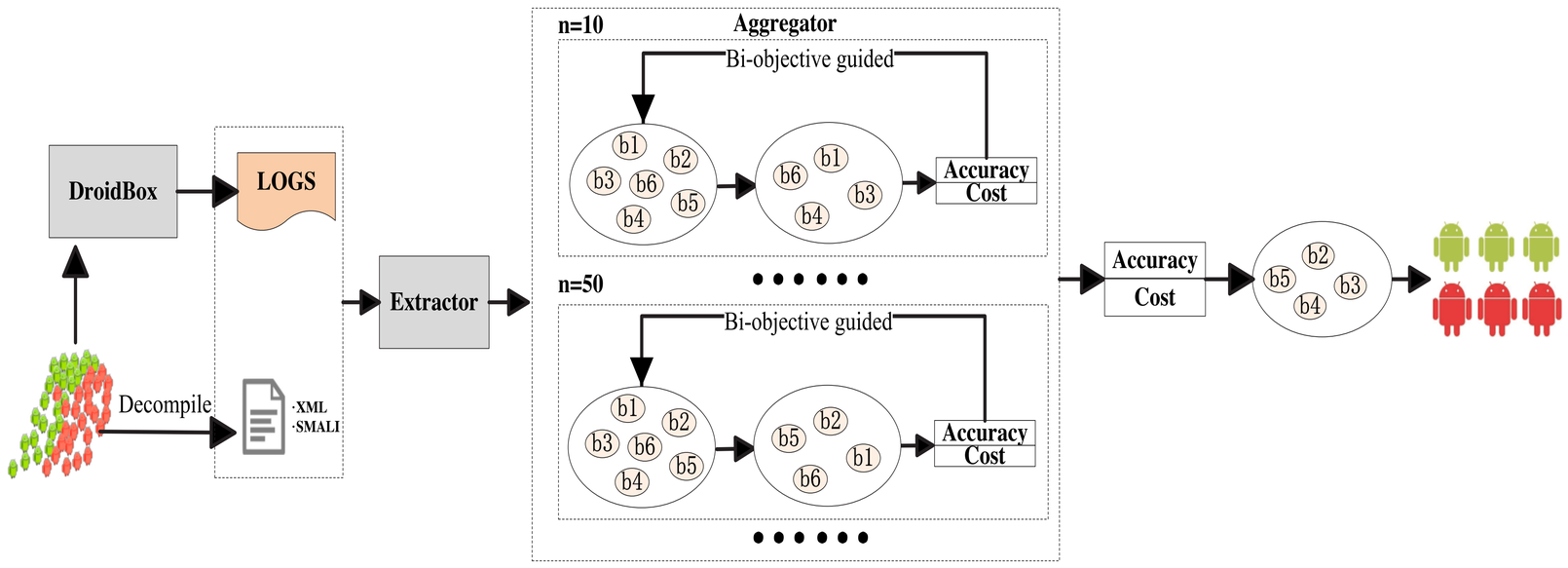
Features

We select 155 features in total to perform a binary classification. Four types of features are shown below:

Type of Features	Selected Features
Permission	59
Sensitive API Call	90
Sequence	1
Dynamic Behavior	5
Total	**155**

Begonia

Four steps: Reverse Engineering ⟶ Feature Extraction ⟶Ensemble Pruning ⟶ Classification

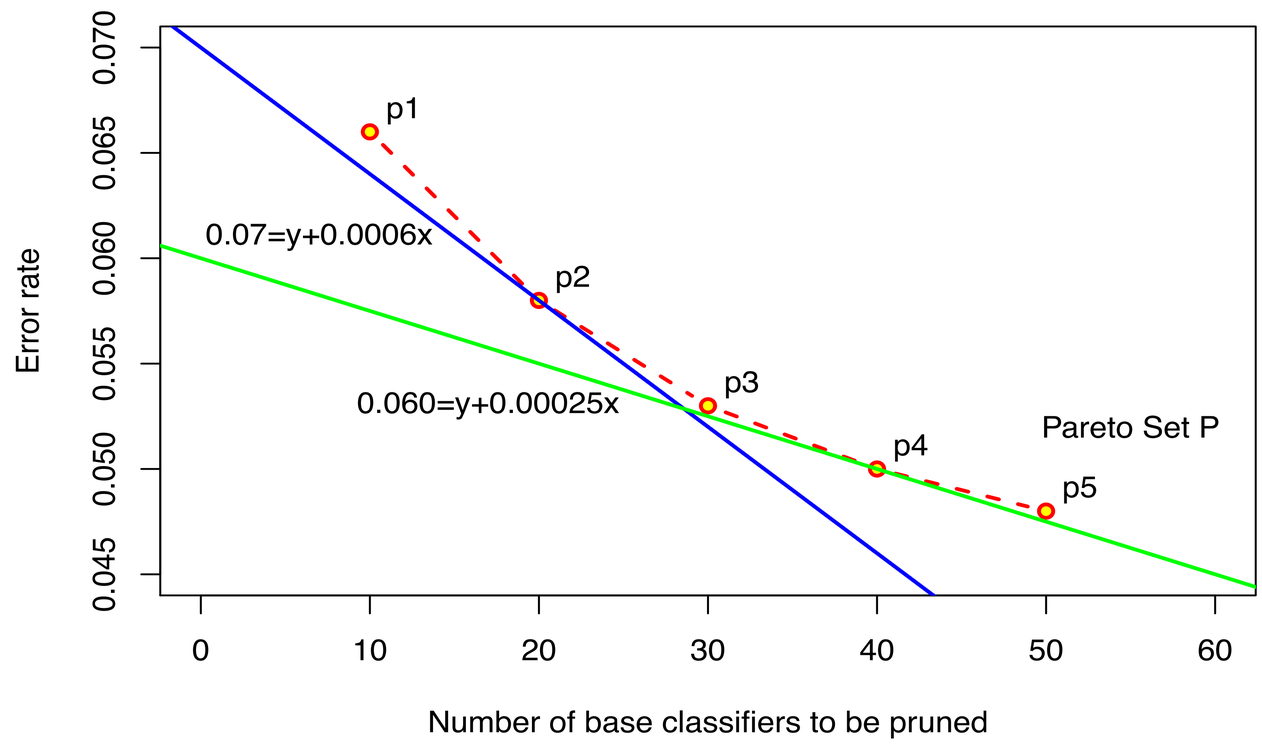


Optimal Ensemble Selection

To trade off the two objectives, provided by a trade-off level w, we select the final ensemble that minimizes the combined loss (*i.e.*, $E(T_t) + w \cdot |T_t|$), e.g., given different trade-off levels,

$$w_1 = 0.0006 \xrightarrow{\text{final ensemble}} p_2;$$

$$w_2 = 0.00025 \xrightarrow{\text{final ensemble}} p_4.$$



Results

Goal: To examine the relation between accuracy and time cost of real-time analysis.
Iteration times: $\lceil n^2 \log n \rceil$, when dealing with the bi-objective solver.

# Group Size	Time (sec)	Accuracy
10	60	93.40%
20	460	94.20%
30	1,546	94.70%
40	3,654	95.00%
50	7,450	95.20%

*Note that accuracy column indicates the highest accuracy of each group.

References

[1] S. Chen, M. Xue, Z. Tang, L. Xu, and H. Zhu. *Stormdroid: A streaminglized machine learning-based system for detecting android malware.* ACM Symposium on Information, Computer and Communications Security (ASIACCS 2016)

[2] C. Qian, Y. Yu, and Z.-H. Zhou. *Pareto ensemble pruning.* AAAI Conference on Artificial Intelligence (AAAI 2015)

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China, under Grant 61502170, 61272444, 61411146001, U1401253, and U1405251, in part by the Science and Technology Commission of Shanghai Municipality under Grant 13ZR1413000, and in part by Pwnzen Infotech Inc.