ACM Transactions on Software Engineering and Methodology

Meta-Learning for Multi-Family Android Malware Classification

Yao Li[1], **Dawei Yuan**[1], **Tao Zhang**[1], **Haipeng Cai**[2], **David Lo**[3], **Cuiyun Gao**[4], **Xiapu Luo**[5] and **He Jiang**[6]

[1] School of Computer Science and Engineering, Macau University of Science and Technology; [2] Washington State University Pullman;
[3] Singapore Management University; [4] Harbin Institute of Technology; [5] The Hong Kong Polytechnic University; [6] Dalian University of Technology

Introduction

- With the emergence of smartphones, Android has become a widely used mobile operating system that is increasingly targeted by various cyber threats.
- Its popularity has attracted relentless attacks, and new malware continuously jeopardizes the security of users' devices and private data.[1]
- Traditional malware classification methods, based on static or dynamic analysis, struggle with challenges such as imbalanced data distributions and the detection of zero-day malware exploiting unknown vulnerabilities.[2]

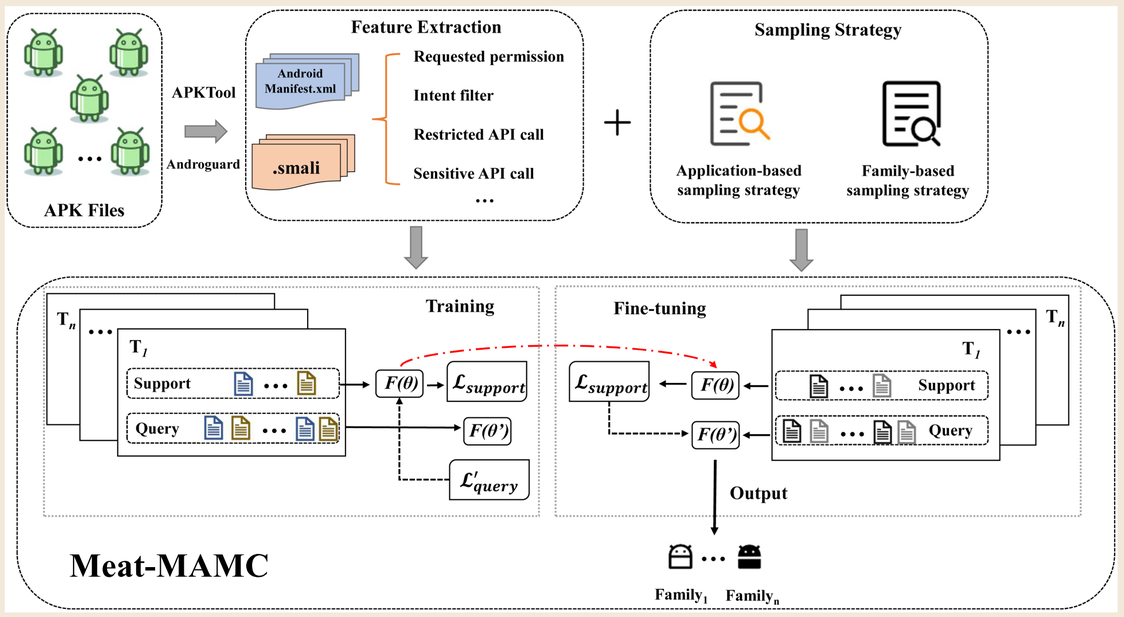
Question

- To what extent is accurate malware classification achievable given limited, imbalanced datasets that feature both sample imbalances among families and the absence of some malware families?

Objectives

- Enhance feature extraction techniques from APK files to improve malware analysis.
- Address sample imbalance through innovative application-based and family-based sampling strategies.
- Advance meta-learning approaches to accurately classify malware, including few-sample and zero-sample families.

Methods



- Feature Extraction: Reverse-engineer APK files to extract eight categories of features from *AndroidManifest.xml* and *class.dex*.
- Sampling Strategy: Mitigate sample imbalance using two methods—application-based and family-based.
- Meta-Learning & Classification: Generate realistic few-sample and zero-sample tasks via dual sampling strategies to train and fine-tune a meta-classifier for malware classification.

Results

Table 1. Meta-MAMC Versus State-of-the-Art Methods on Drebin. The Accuracy and F-Score of the ten random families in the Drebin dataset.

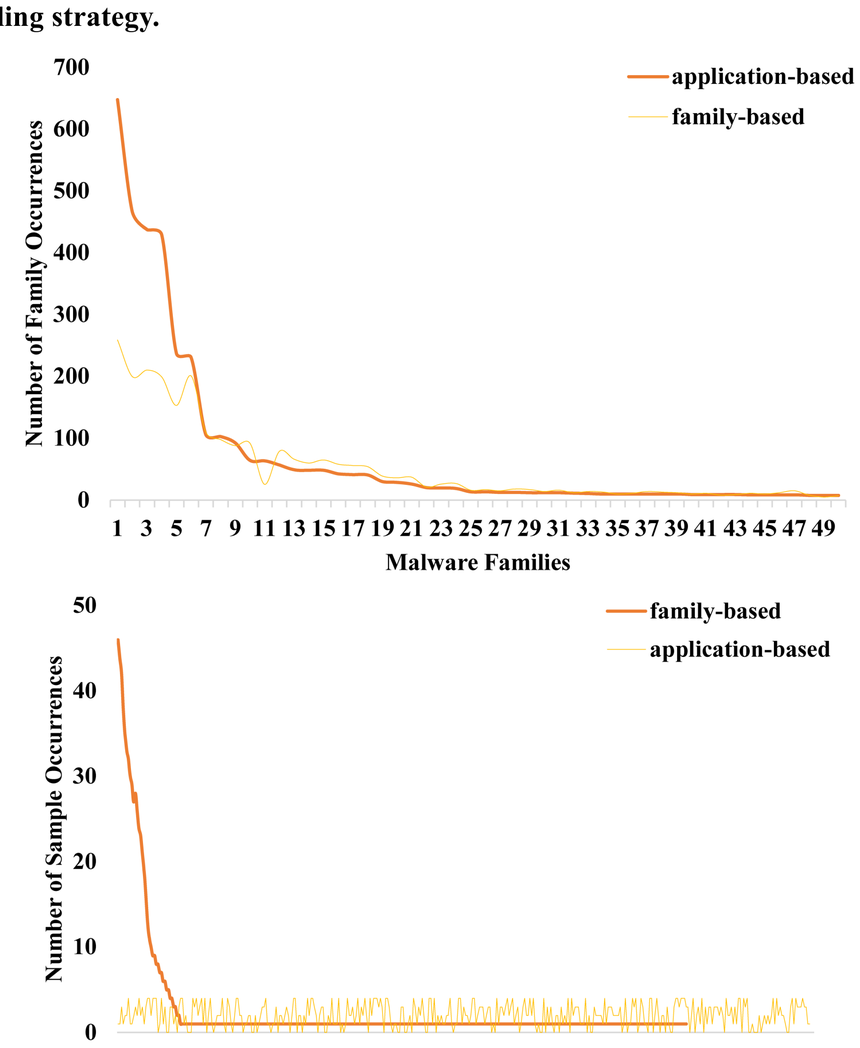
Family	Drebin		MaMaDroid		N-opcode		EC2		**Meta-MAMC**	
	ACC	F-Score	ACC	F-Score	ACC	F-Score	ACC	F-Score	ACC	F-Score
FakeInstaller	0.786	0.773	0.874	0.875	0.864	0.875	0.827	0.819	**0.976**	**0.966**
DroidKungFu	0.769	0.781	0.861	0.849	0.852	0.867	0.843	0.826	**0.992**	**0.979**
Plankton	0.753	0.734	0.844	0.853	0.835	0.854	0.798	0.785	**0.963**	**0.961**
Opfake	0.765	0.769	0.851	0.836	0.847	0.864	0.811	0.805	**0.971**	**0.966**
BaseBridge	0.725	0.732	0.832	0.824	0.816	0.827	0.775	0.792	**0.945**	**0.943**
Imlog	0.700	0.695	0.813	0.806	0.800	0.780	0.787	0.769	**0.966**	**0.983**
Jifake	0.4	0.571	0.577	0.732	0.242	0.391	0.700	0.701	**0.909**	**0.954**
Zitmo	0.231	0.377	0.546	0.706	0.143	0.252	0.417	0.589	**0.700**	**0.824**
Stiniter	0.167	0.154	0.5	0.429	0.166	0.153	0.5	0.356	**0.833**	**0.910**
SmsSpy	0	0	0	0	0	0	0	0	**1**	**1**

Table 2. Performance of Meta-MAMC and SOTA methods against Android evolution.

Method	Accuracy & F-Score									
	Scenario A		Scenario B		Scenario C		Scenario D		Scenario E	
EFIMDetector	0.889	0	0.886	0.902	0.905	0.911	0.884	0.879	0.860	0.856
MaMaDroid	0.823	0.818	0.864	0.883	0.854	0.863	0.875	0.867	0.854	0.861
Drebin	0.838	0.846	0.913	0.921	0.903	0.918	0.646	0.668	0.650	0.655
SEDMDroid	0.854	0.861	0.863	0.871	0.868	0.890	0.877	0.858	0.843	0.838
MMN	0.867	0.868	0.866	0.875	0.873	0.889	0.874	0.872	0.863	0.857
Meta-MAMC	**0.945**	**0.939**	**0.933**	**0.938**	**0.946**	**0.952**	**0.951**	**0.960**	**0.946**	**0.951**

- Five evaluation scenarios (A–E) simulate training on earlier malware datasets and testing on later samples (from 2012–2022) collected from VirusShare, AndroZoo, and GitHub, ensuring no overlap in malicious families, to compare the F-Score and accuracy of EFIMDetector, MaMaDroid, Drebin, SEDMDroid, MMN, and Meta-MAMC and verify Meta-MAMC's robustness against malware evolution.
- Meta-MAMC leverages meta-knowledge to counter zero-day samples and Android evolution, slightly outperforming competing schemes in classifying new samples with an older dataset.

Figure 1: The number of times each family and sample occurring in the tasks sampled from Drebin according to the application-based or family-based sampling strategy.



- Application-based sampling is biased by family but fair by application, while family-based sampling is the opposite, making them complementary.

Conclusions

- Introduce Meta-MAMC, a novel meta-learning approach for multi-family Android malware classification.
- Develop two sampling strategies: application-based sampling to counter imbalanced family distributions and family-based sampling to address incomplete datasets using zero-sample learning.
- Regulate the two strategies with a hyperparameter p, which balances the probability between family-based and application-based sampling.
- Validate Meta-MAMC through experiments that outperform state-of-the-art methods, with future plans to enrich datasets and tackle intra-class imbalance issues.

Acknowledgements: I would like to express my sincere gratitude to Prof. Tao Zhang (supervisor), Prof. Xiapu Luo, Prof. Haipeng Cai, Prof. David Lo, Prof. Cuiyun Gao, and Prof. He Jiang for all their guidance and help during my PhD study.

Contact

- E-mail: liyaouu@gmail.com
- Homepage: https://leoleeyau.github.io/
- ORCID: 0000-0002-0474-0159

References:

1. Kiran Radhakrishnan, Rajeev R Menon, and Hiran V Nath. 2019. A survey of zero-day malware attacks and its detection methodology. In TENCON 2019 – 2019 IEEE Region 10 Conference (TENCON). 533–539.
2. G. Suarez-Tangil and G. Stringhini. 2022. Eight Years of Rider Measurement in the Android Malware Ecosystem. IEEE Transactions on Dependable and Secure Computing 19 (2022), 107–118.