DriodAPIMiner: Robust Malware detection in Android

Yousra Aafer, Wenliang Du, and Heng Yin

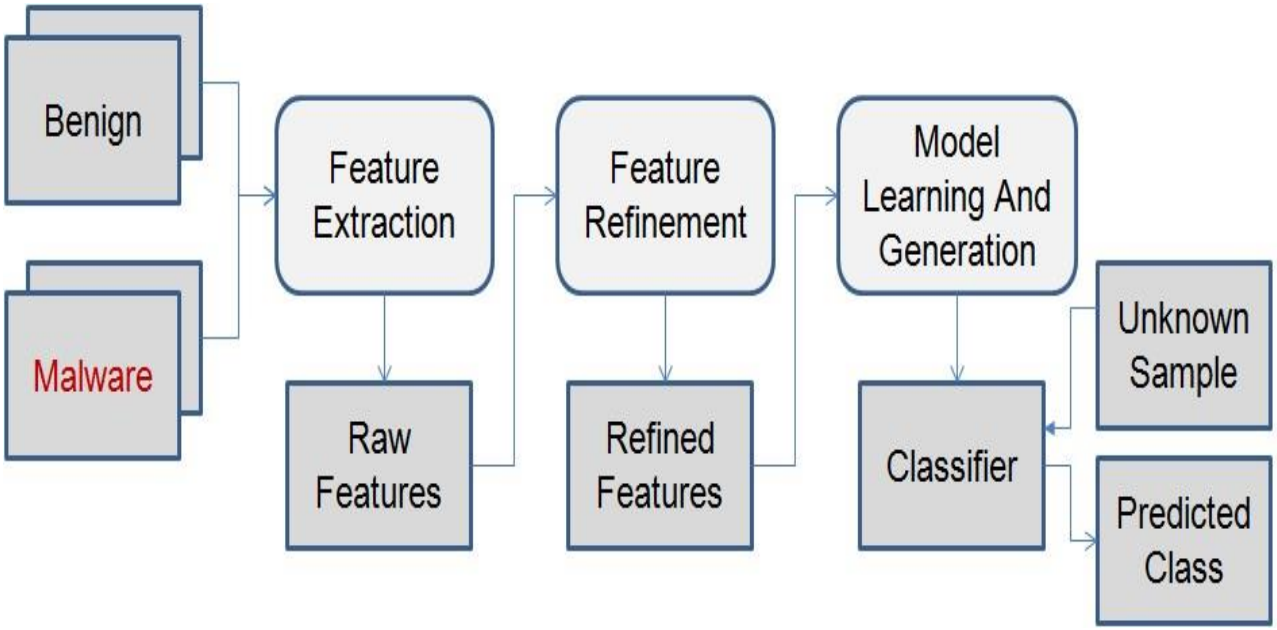
Abhilash Parthasarathy

CISC850
Cyber Analytics

1. Introduction

- Robust and efficient approach
- Malware behavior at API-level.
- Achieving high accuracy and low false positive rates.

2. Approach


Benign
Malware
Feature Extraction
Raw Features
Feature Refinement
Refined Features
Model Learning And Generation
Classifier
Unknown Sample
Predicted Class

3. Feature Extraction and Refinement

- Extraction of Dangerous APIs
- Extraction of Package Level Information
- Extraction of APIs Parameters

3.1 Extraction of Dangerous APIs

- Generate a list of distinct API calls.
- Format of a distinct API- Class Name, Method Name, and Descriptor.

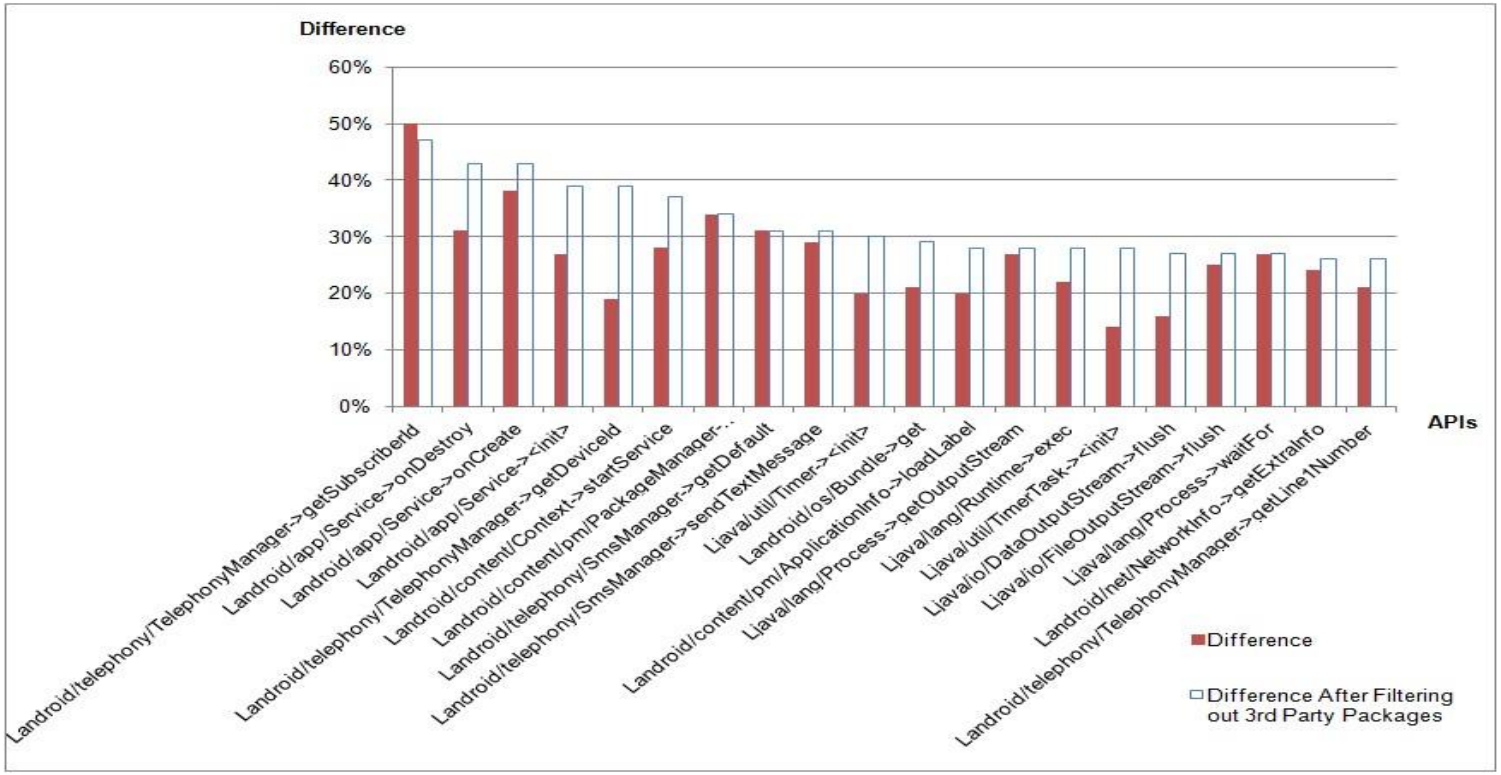
3.2 Extraction of Package Level Information

- Removing all common packages
- Checking if the API is invoked by a third party or not.

3.3 Extraction of API parameters

Classes	Methods	Parameter Category
Intent IntentFilters	setFlags, addFlags, setDataAndType, putExtra, init	Flag is either: CALL, CONNECTIVITY, SEND, SENDTO, or BLUETOOTH
ContentResolver	query, insert, update..	URI is either: Content://sms-mms, Content://telephony, Content://calendar, Content://browser/bookmarks, Content://calllog, Content://mail, or Content://downlaods
DataInputStream BufferedReader DataOutputStream DataOutputStream	init, writeBytes...	Reads from process Reads from connection Uses SU command
InetSocketAddress	init	parameter IP is explicit or port is 80
File Stream StringBuilder String StringBuffer	init, write, append, indexOf, Substring	Dangerous Command such as: su, ls, loadjar, grep, /sh, /bin, pm install, /dev/net, insmod, rm, mount, root, /system, stdout, reboot, killall, chmod, stderr Accesses external storage or cache Contains either: An identifier (e.g. Imei), an executable file(e.g. .exe, .sh), a compressed file (e.g. jar, zip), a unicode string, an sql query, a reflection string, or a url

4. Insights in API-Level Malware Behavior



4.1 Types of APIs

- Application-specific
- Android framework resources
- DVM related resources APIs
- System resources APIs
- Utilities APIs

4.2 Parameters Features

Class		Method	Parameter type	Difference (%)
StringBuilder		append	Dangerous command	35.95
ContentResolver		query	SMS or MMS	23.65
StringBuilder		append	Unicode string	23.6
StringBuilder		init	Dangerous command	23.07
DataOutputSream		writebytes	Reads from process	21.80
DataOutputSream		init	Reads from process	21.62
runTime		exec	Dangerous command	21.27
InetSocketAddress		init	Port 80	19.91
StringBuilder		append	Compressed file	19.58
DataInputStream		init	Reads from connection	19.27
String		valueOf	Unicode string	18.05
StringBuilder		append	File manupilation	17.79
File		init	Accesses external storage	16.92
InetSocketAddress		init	Explicit IP	14.87
String		getBytes	URL manupilation	14.05
Intent		setFlags	SendTo	12.94
Intent		setFlags	Call	11.67
ContentResolver		query	Telephony	10.88
Intent		setFlags	Send	10.47
ContentResolver		query	Call log	10.12

5. Classification and Evaluation

- Data Set and classification of models
- Permission based feature set
- API based feature distribution
- Comparison of Models.
- Processing time.

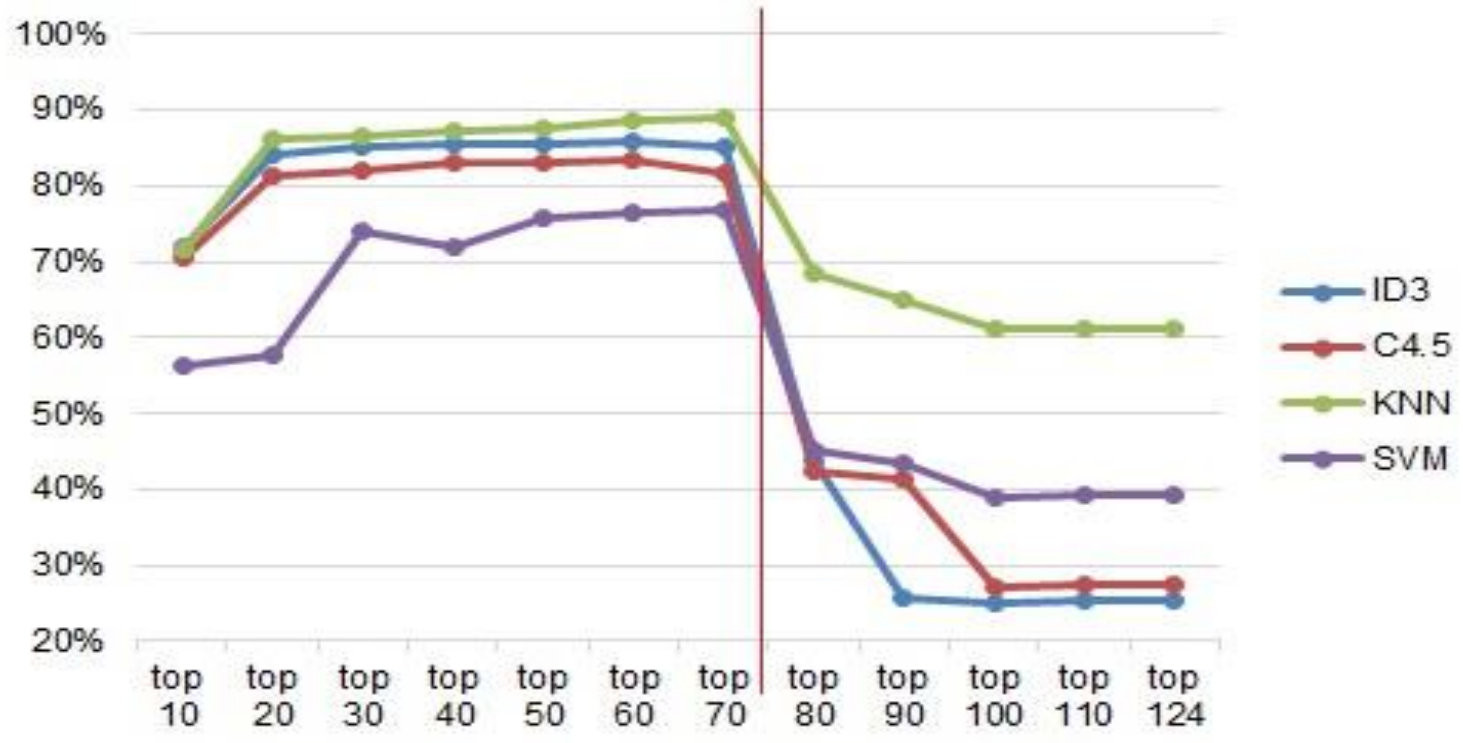
5.1 Data Set and classification of models

- Analyzed around 20,000 apps.
- ID3 DT , C4.5 DT, KNN, and linear SVM models.
- C4.5 and ID3 are related to decision trees and KNN belong to Lazy classifiers.

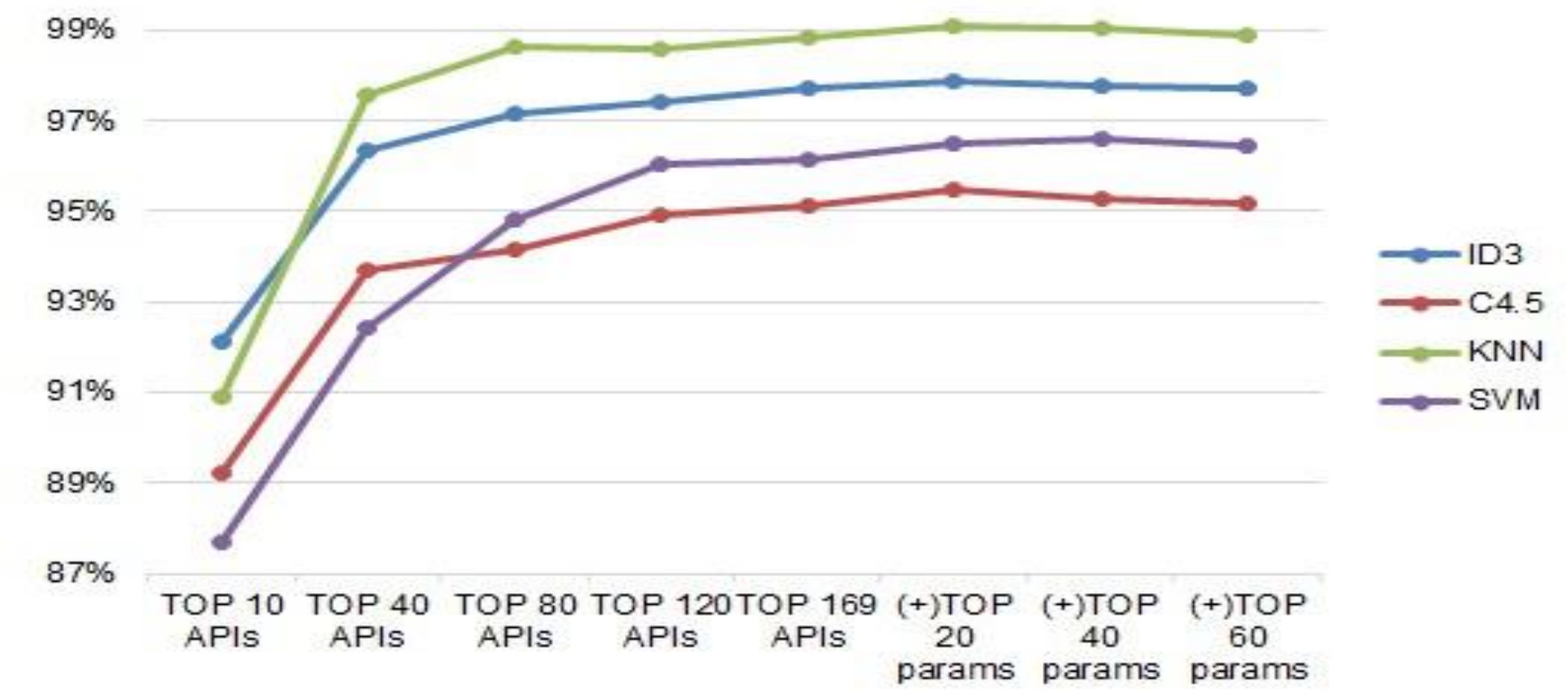
5.2 Classification Models

- How is TPR calculated: TP/ (TP + FN)
- How is TNR Calculated : TN/ (TN + FP)
- Accuracy: (TP + TN)/ (TP + TN + FP + FN)

5.3 Permission-Based Feature Set

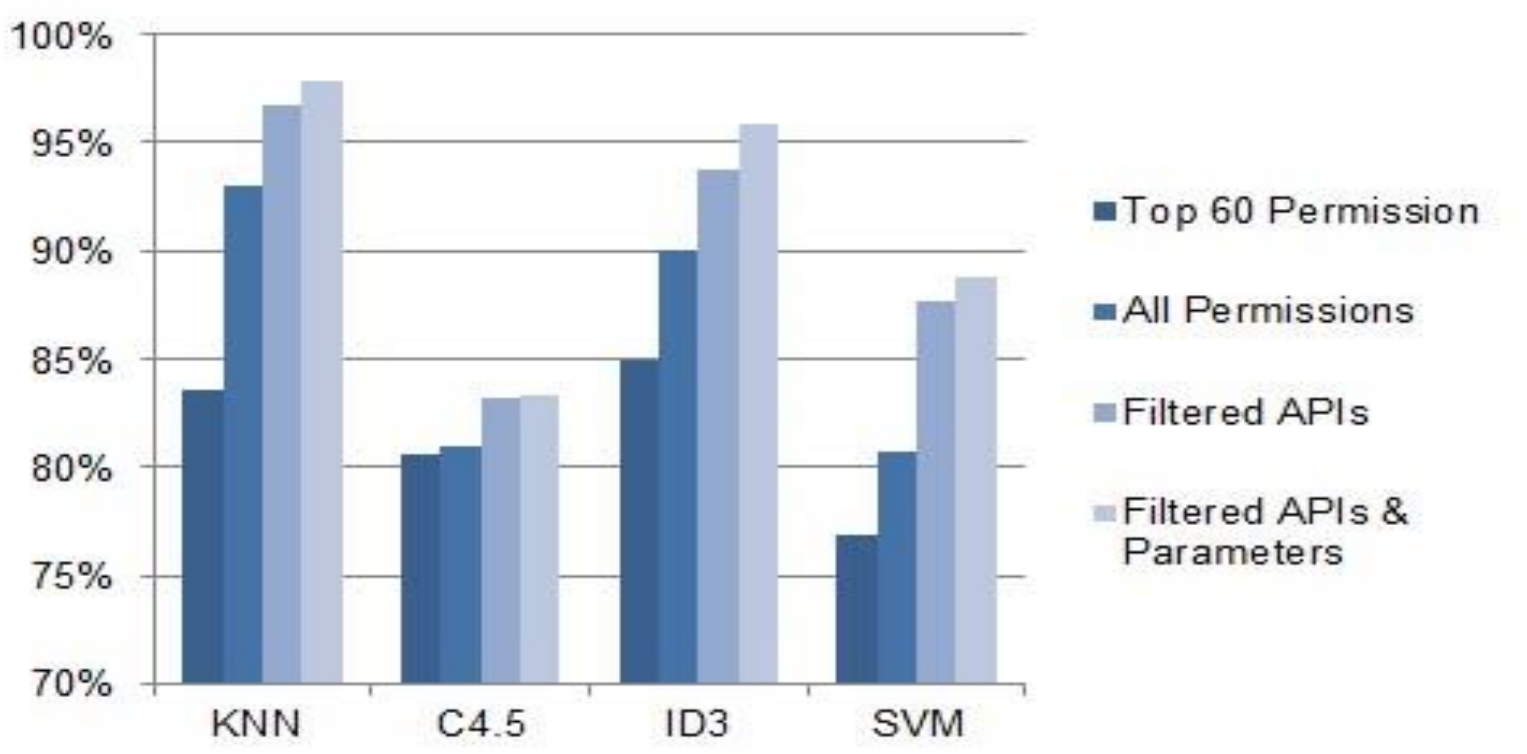


5.4.1 API-Based Feature Accuracy

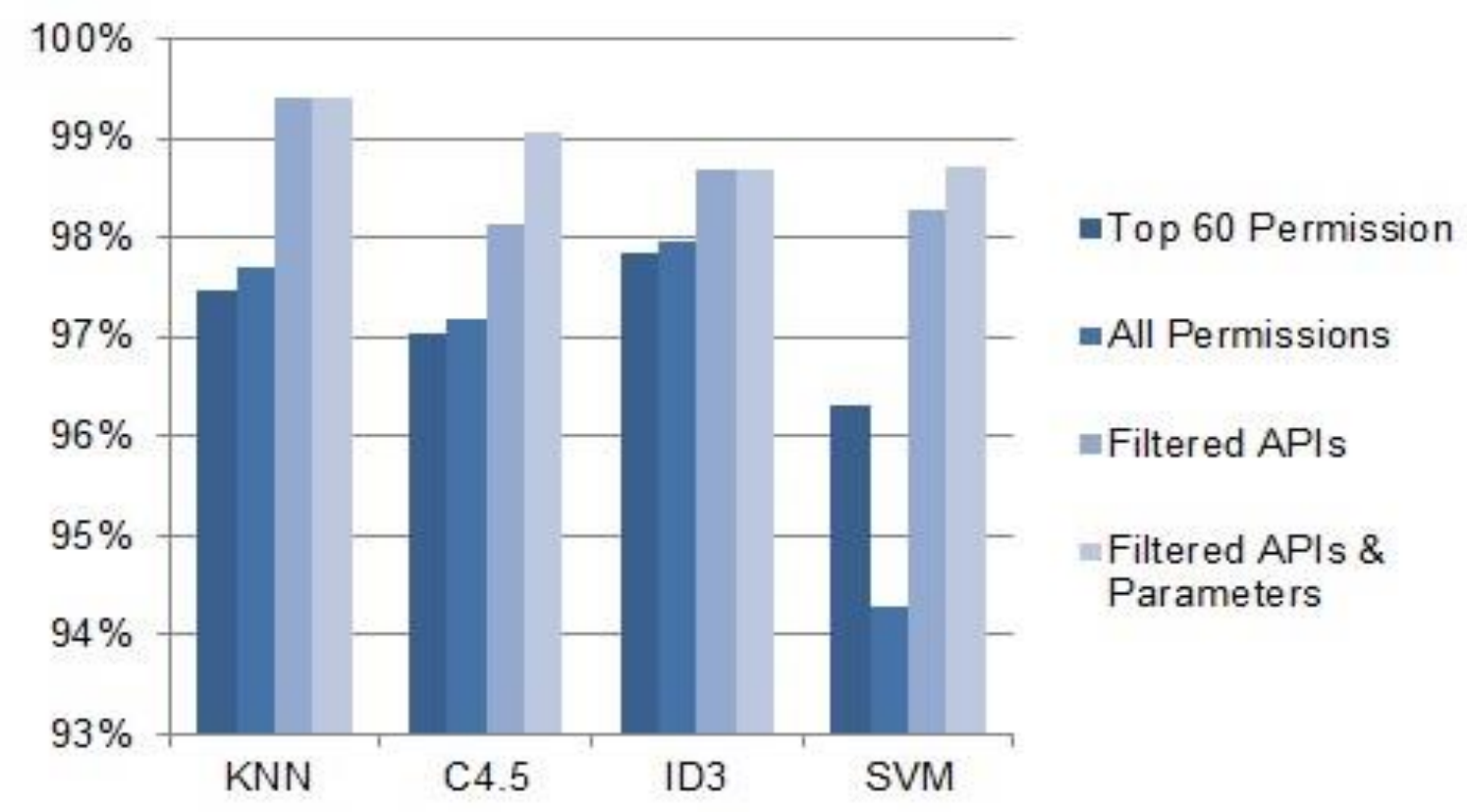


5.5.1 Models Comparison- Accuracy

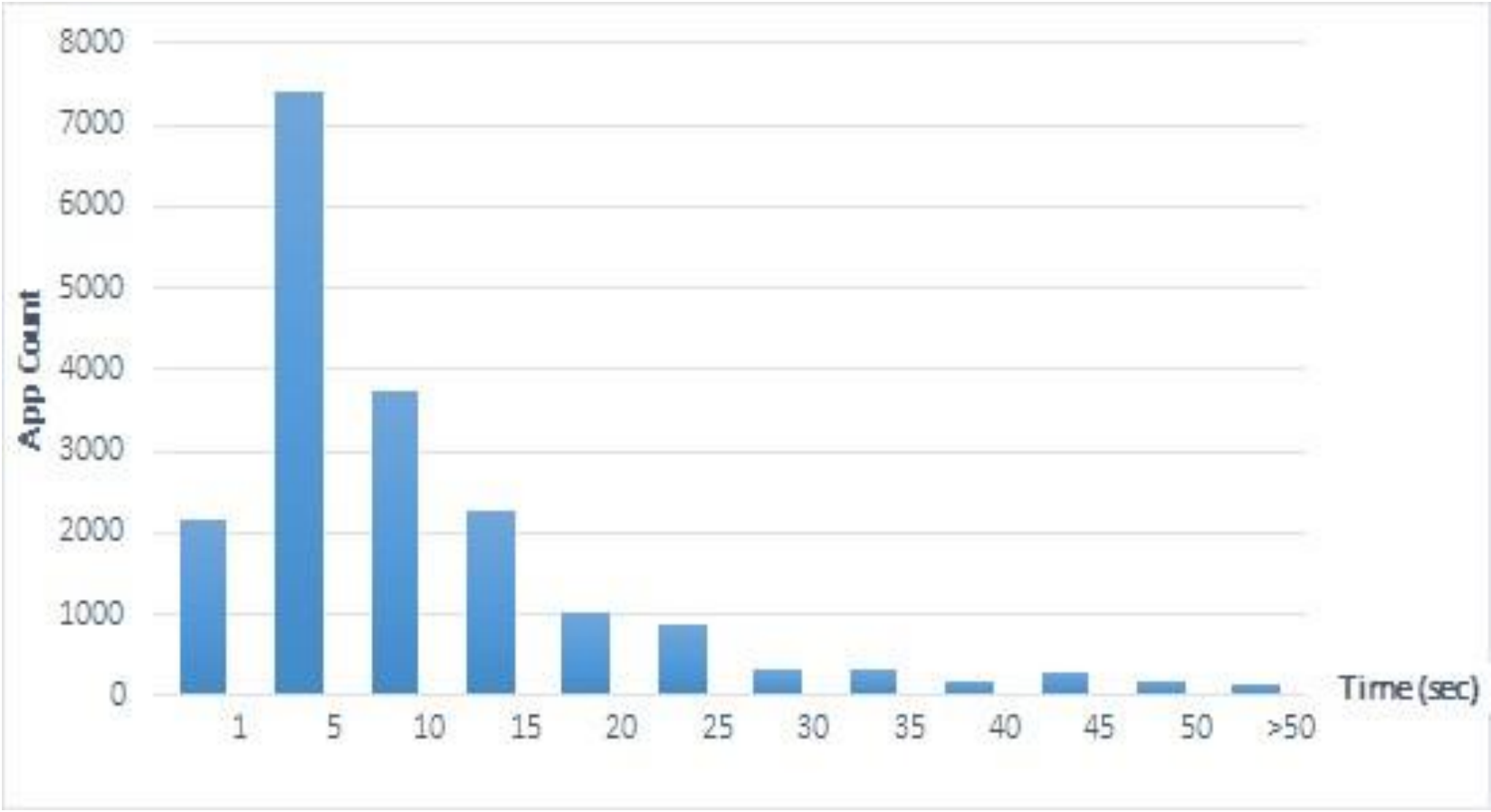
5.5.2 True Positive Ratio



5.5.3 True Negative Ratio



5.6 Processing Time



Conclusion

- Extracted API calls that malware invoke and performed classification
- Achieve a better accuracy, TPR and TNR than the permissions-based feature models.