

Malware Category Detection

Karan Sunil
Nancy Vuong
Kevin Elkin



**85% market
share**

20%

Apps on the Google play store are malicious.

Performed static code analysis to detect the category of malware an app belonged to.

Performed model explainability to understand the strengths and shortcomings of our model.

Categories and Types of Malware



Adware



Trojan



Created by Krisada
from Noun Project



Backdoor



Created by Peter van Driel
from Noun Project



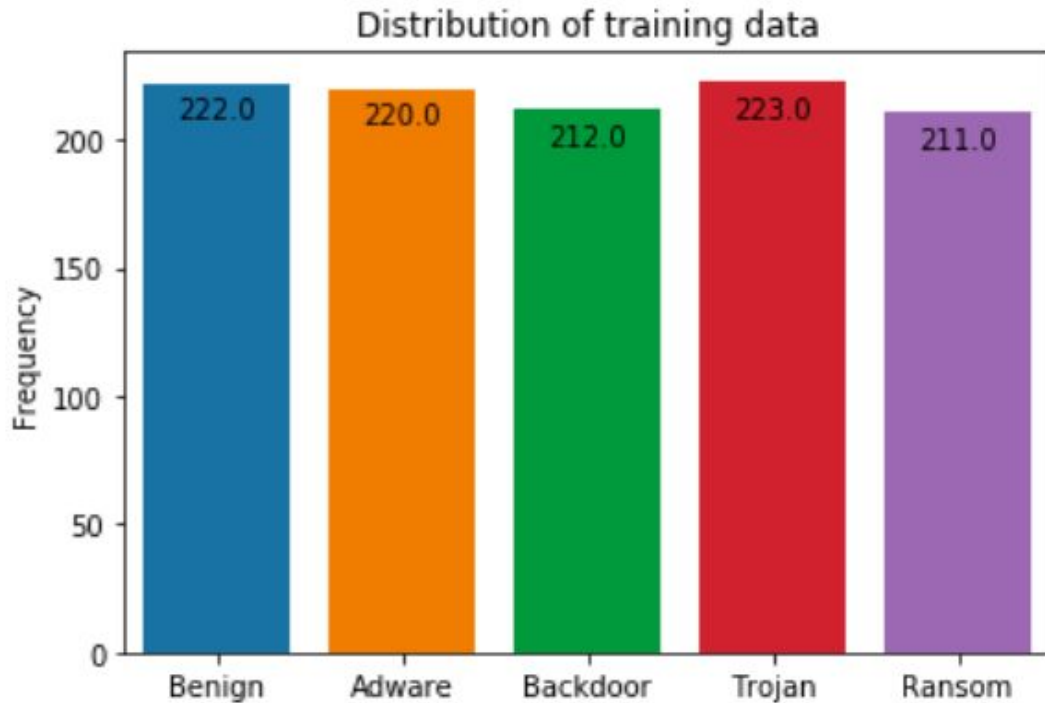
Ransomware

Dataset

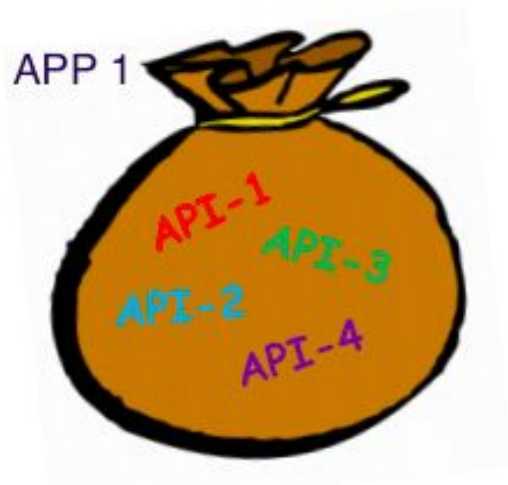
Downloaded 300 apps for each category and performed a 75-25 train-test split.

Benign apps were sourced from apkpure.com

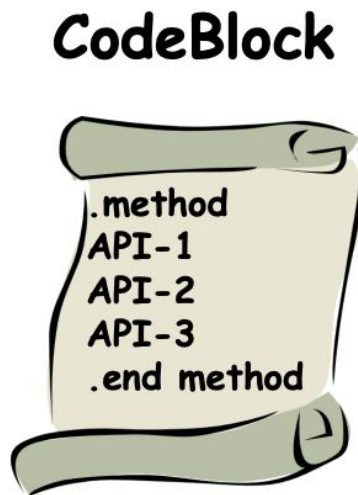
Malware apps were sourced from amd.arguslab.org



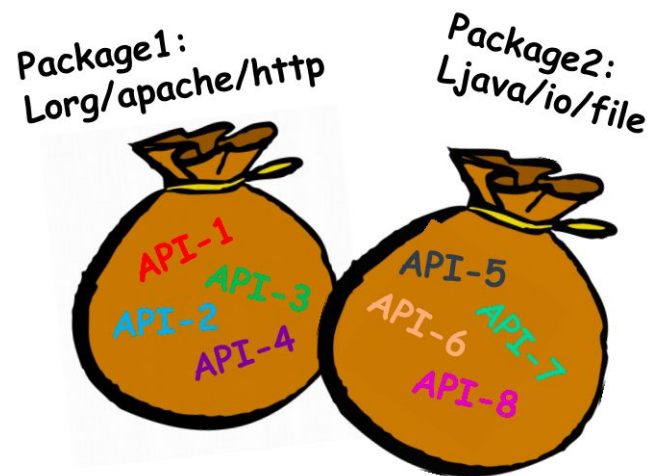
Hindroid Explanation



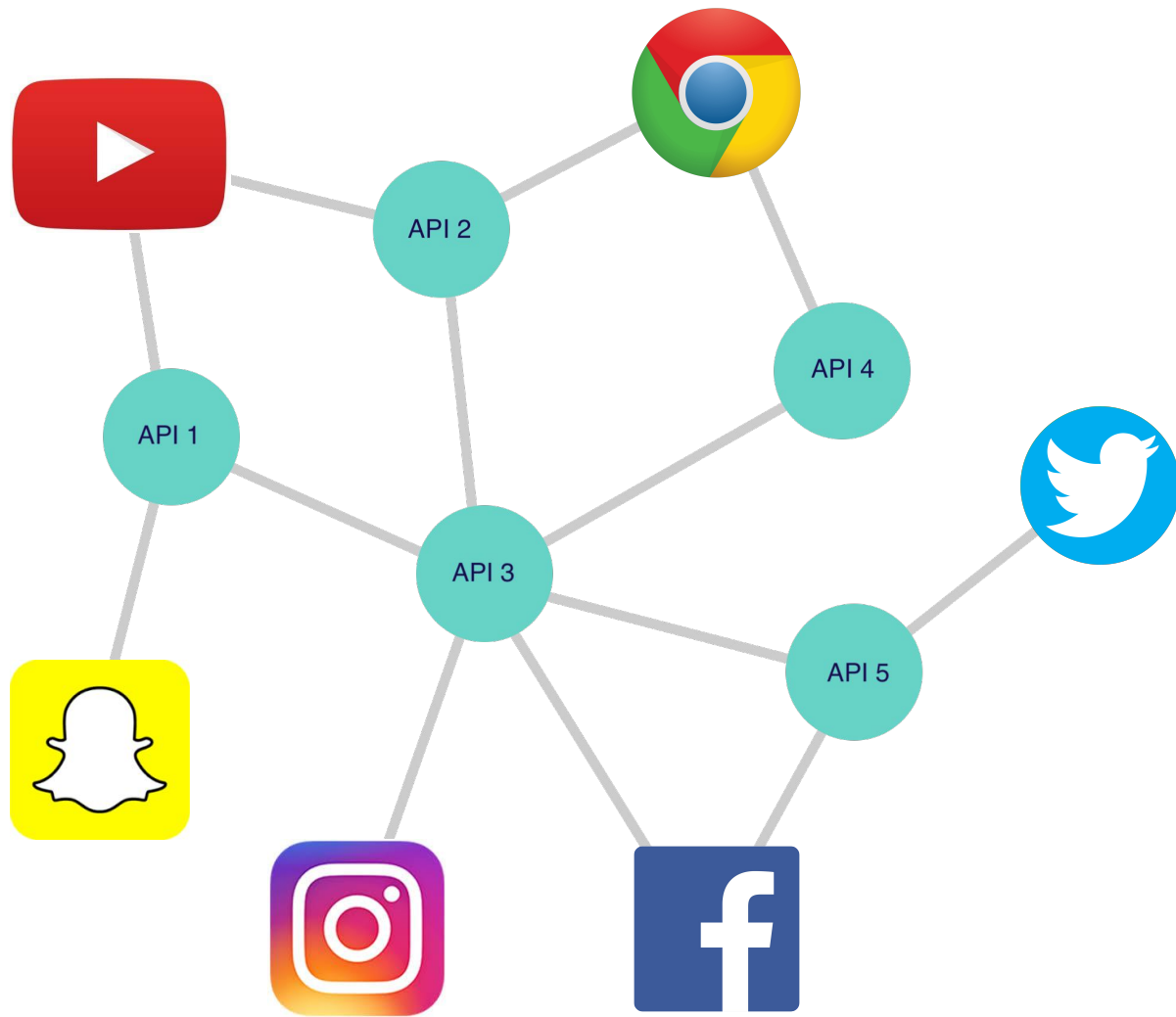
A matrix



B matrix



P matrix



Ensemble Model

We trained 3 separate multi class classifiers using the following kernels - AA^T , ABA^T , APA^T .

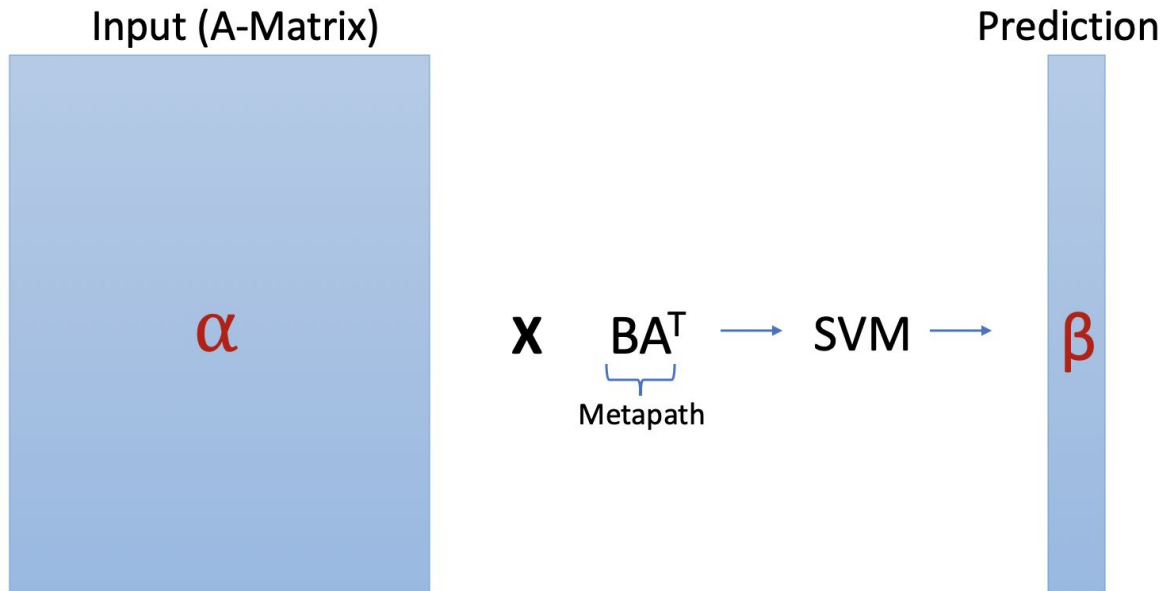
Using the 3 predictions, we would pick the most common prediction.

Helps eliminate weaknesses of certain kernels.

Ex - Backdoor performs bad on APA^T . Balanced by other kernels.

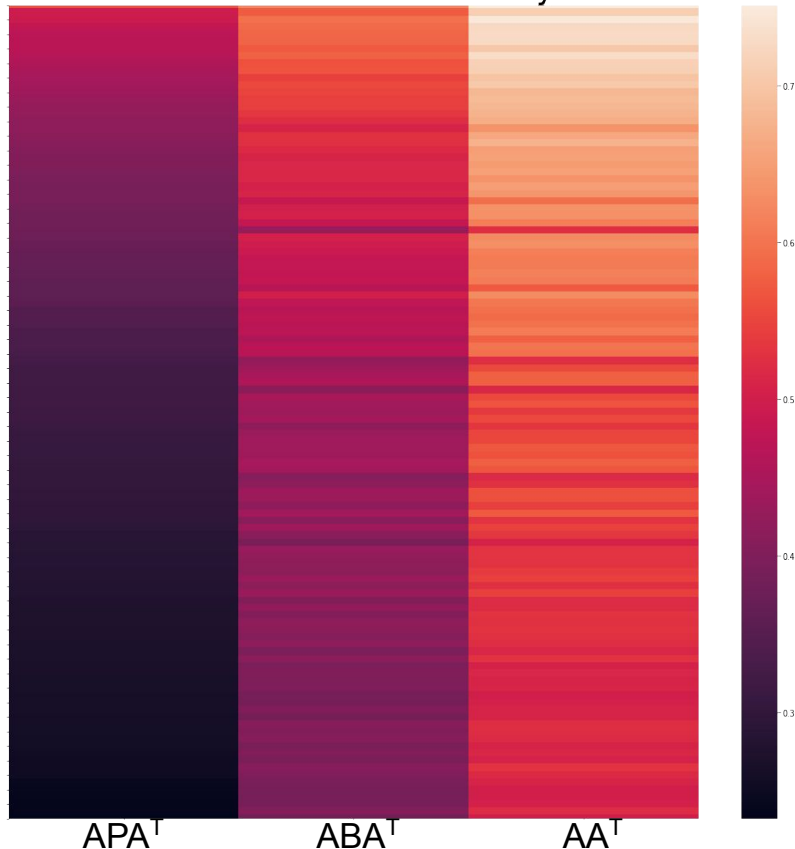
	True Negatives	False Positives	False negatives	True positives	F1
Benign	295	4	5	73	0.94
Adware	295	6	4	72	0.94
Trojan	295	7	6	69	0.91
Backdoor	298	2	8	69	0.93
Ransom	301	5	1	70	0.96

Correlation Coefficient



Correlation Coefficient

Correlation Coefficients of APIs by Kernels



Correlation Coefficient for a Direct Relationship	Correlation Coefficient for an Indirect Relationship	Relative Strength of the Variables
0.0	0.0	None/trivial
0.1	- 0.1	Weak/small
0.3	- 0.3	Moderate/medium
0.5	- 0.5	Strong/large
1.0	- 1.0	Perfect

AA^T

- 111 API's with Correlation Coefficients greater than 0.5

ABA^T

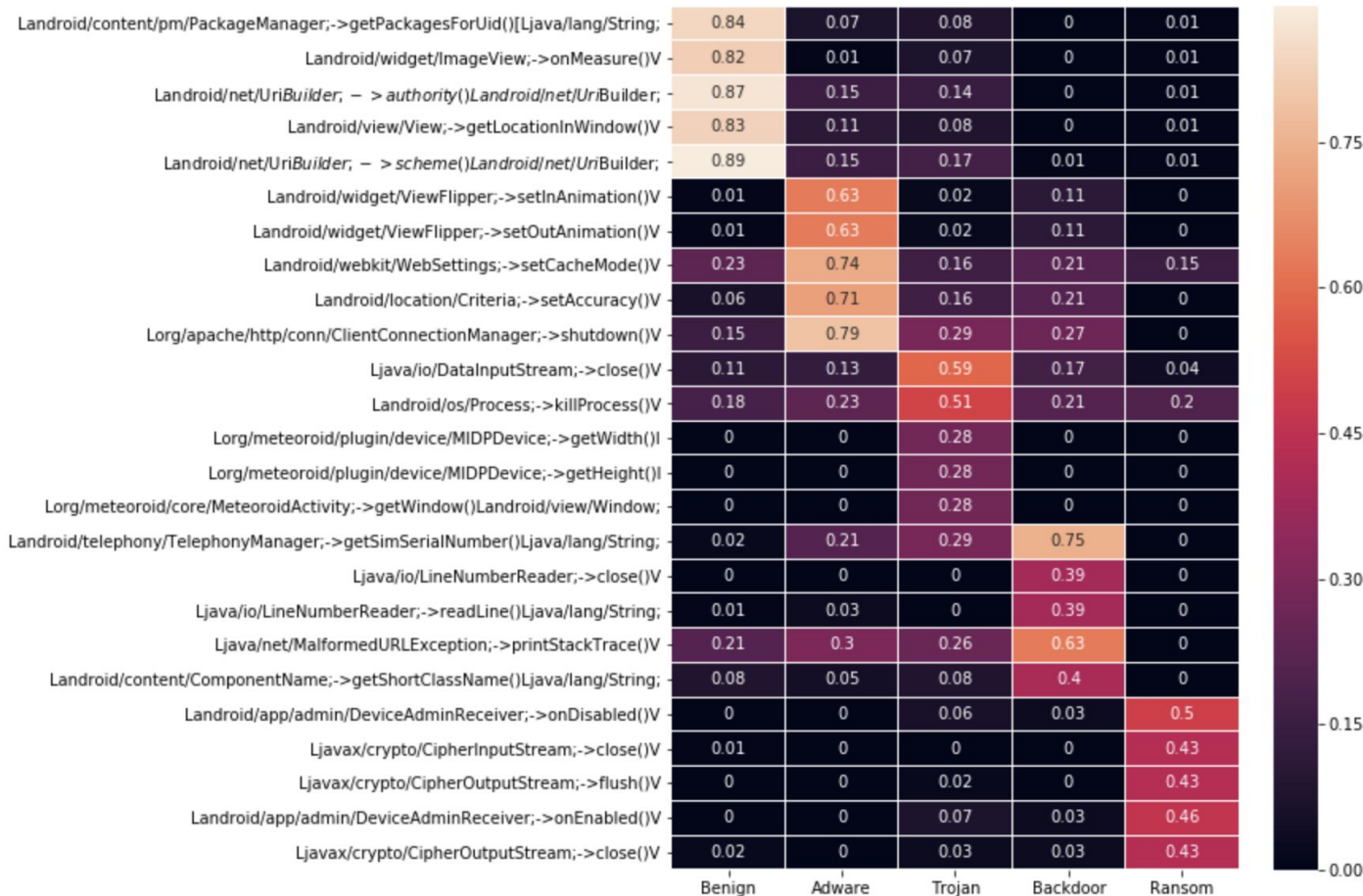
- 29 API's with Correlation Coefficients greater than 0.5

APA^T

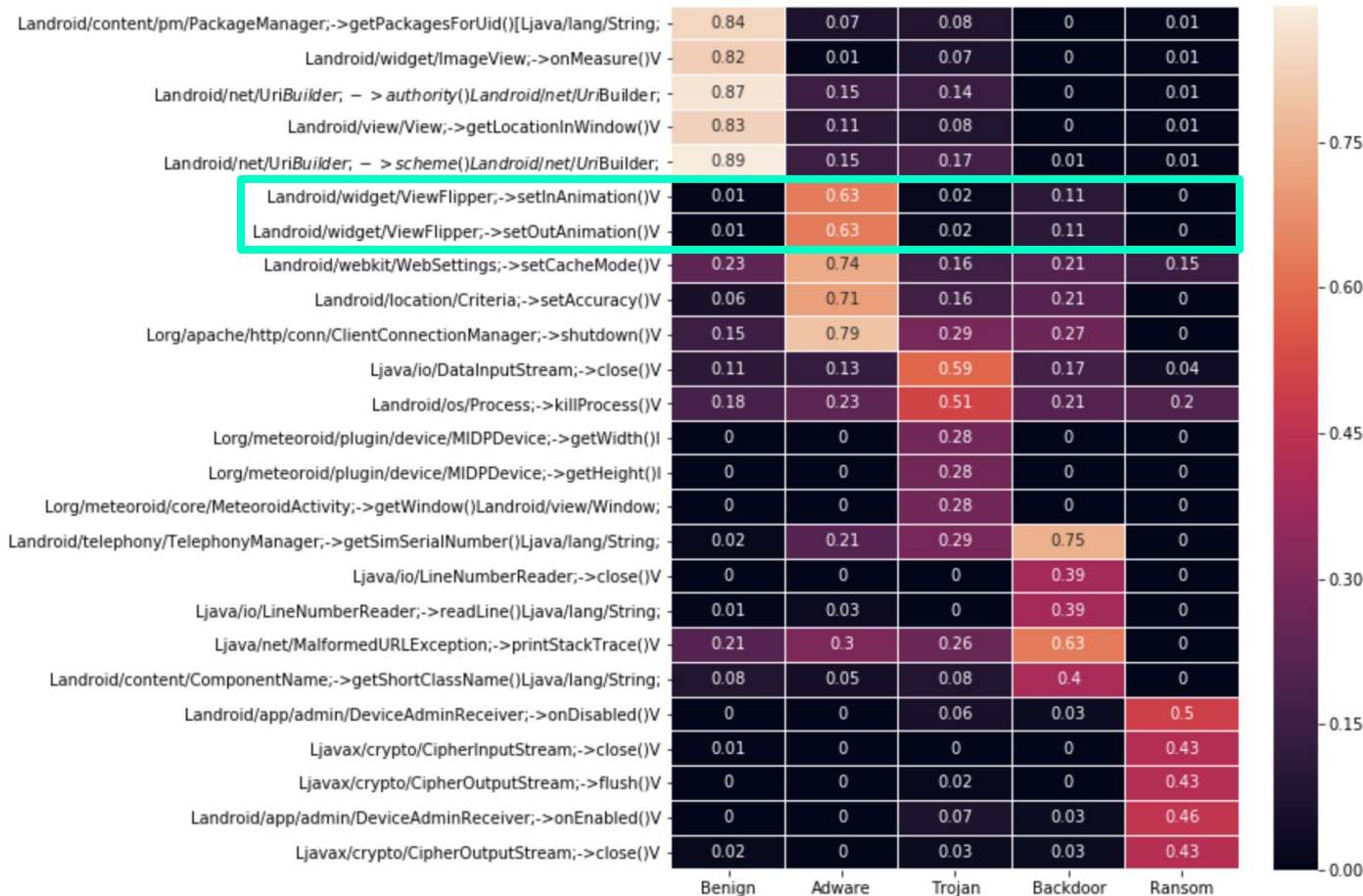
- 1 API's with a Correlation Coefficient greater than 0.5

0 API's with Correlation Coefficients less than -0.5 across all kernels

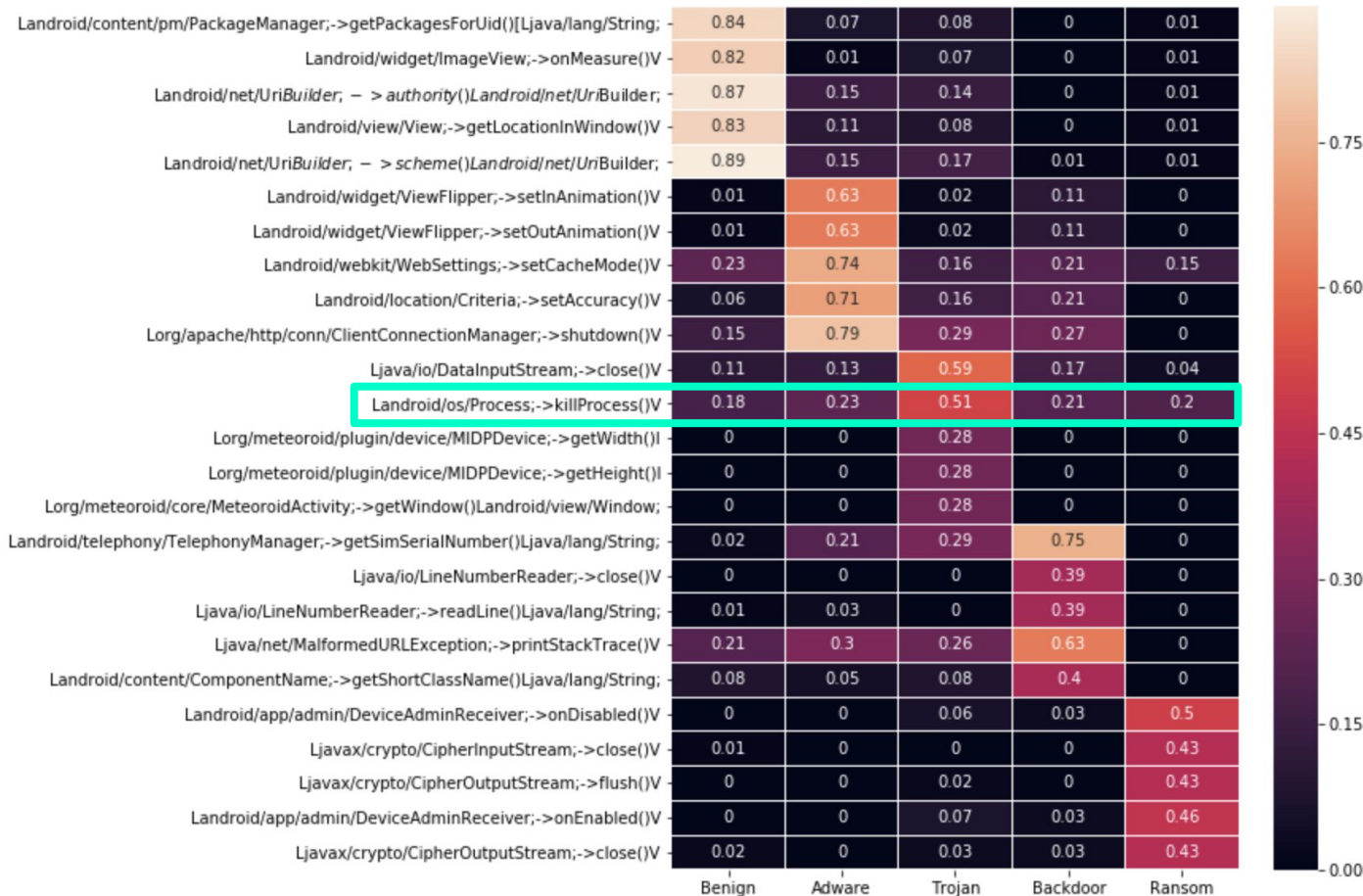
Ranking Algorithm



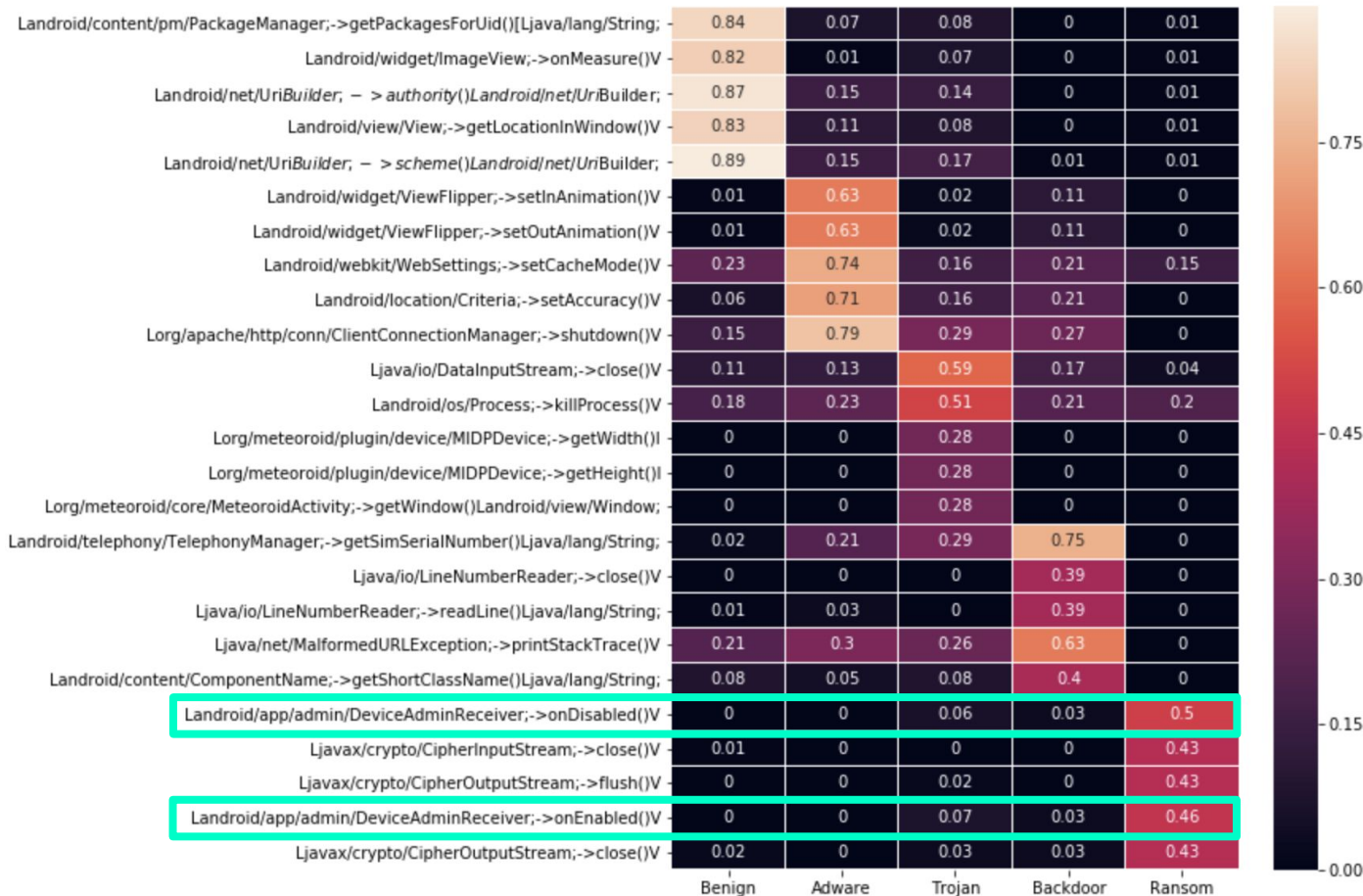
Ranking Algorithm



Ranking Algorithm



Ranking Algorithm

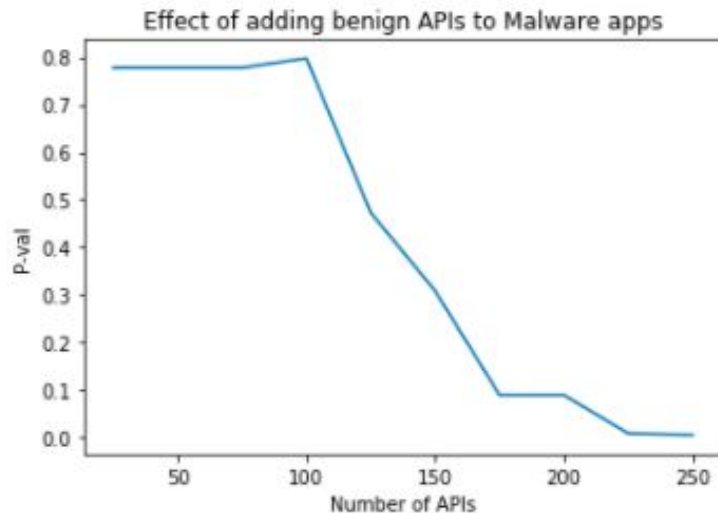


Hypothesis Testing

We infused the APIs that we knew were most Benign into malware apps.

As we add more apps our results are significantly worse.

Adware's F1 score actually improves by 0.01



	True Negatives	False Positives	False negatives	True positives	F1
Benign	256	43	4	74	0.76
Adware	299	2	6	70	0.95
Trojan	297	5	12	63	0.88
Backdoor	299	1	18	59	0.86
Ransom	303	3	14	57	0.87

SVM Weights

Multi Class SVM Weights

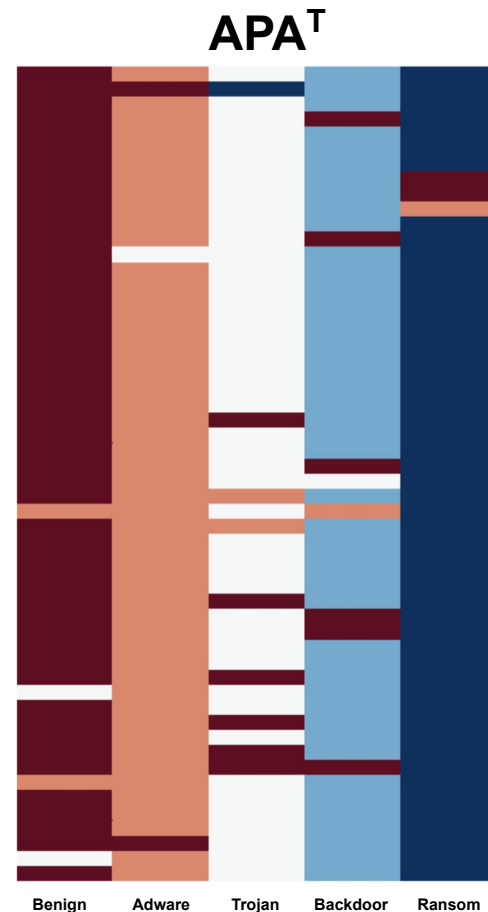
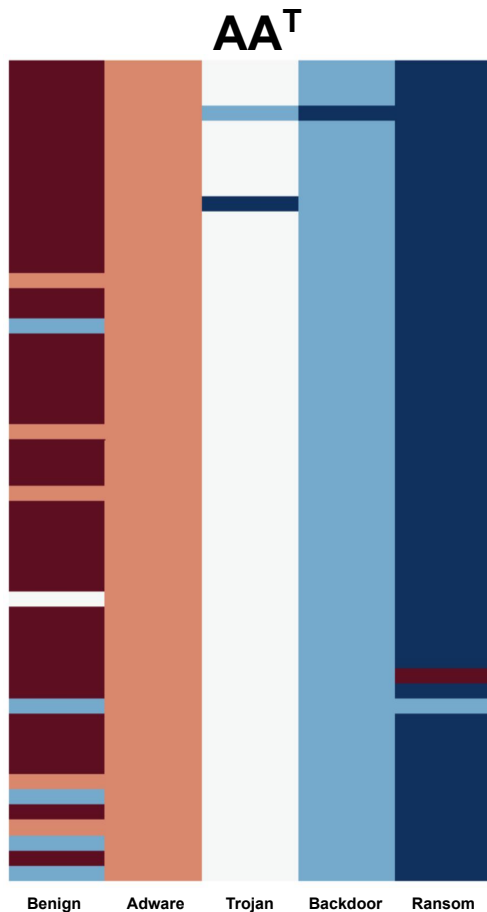
1 vs. The Rest

- Positive Weights correspond to 1 class (i.e. Benign)
- Negative Weights correspond to incorrect classes (i.e. All malware classes)

Figures shown have the top 5% of their positive weights in order from greatest (top) to least (bottom)

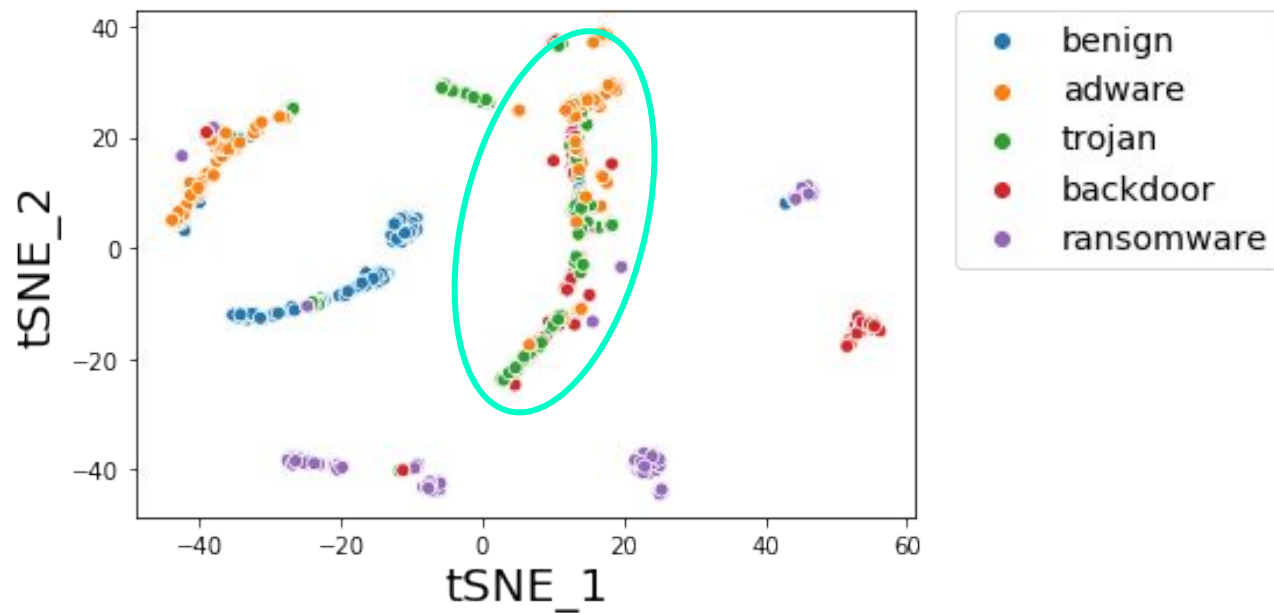
Weights on the top have more say in the classification as they are larger

Top 5% of positive weights account for ~ 30-45% of “decision” on the positive side



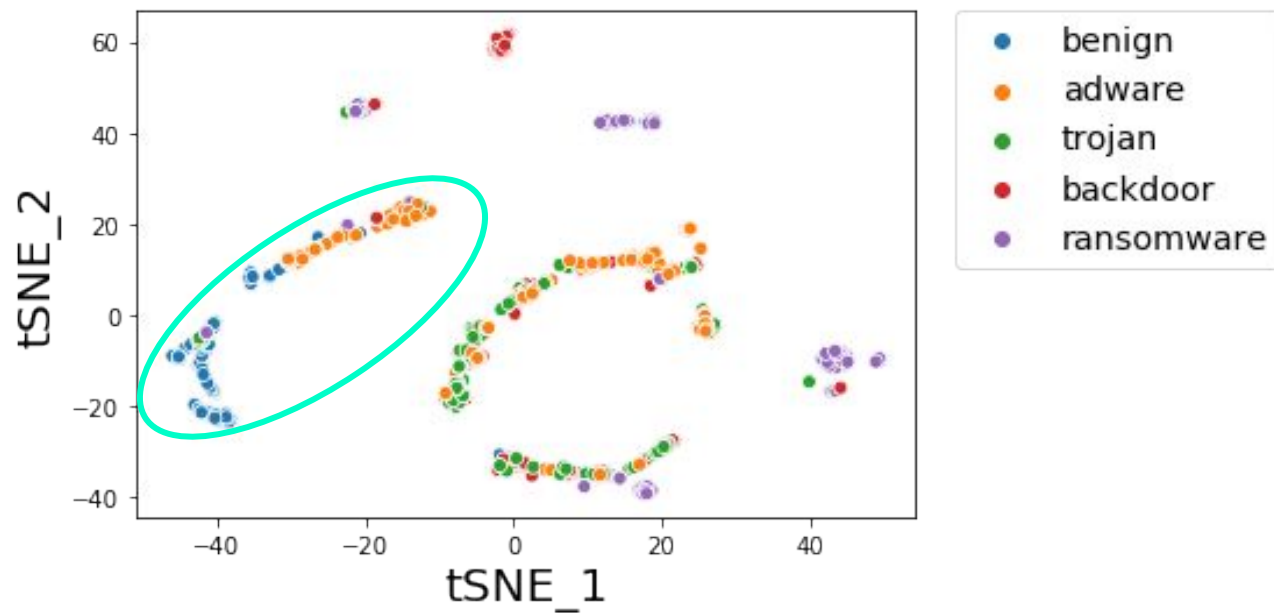
tSNE

AA^T



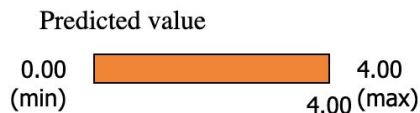
tSNE

APA^T

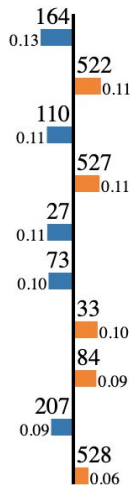


Local Interpretable Model-agnostic Explanations (LIME)

```
Intercept 1.8429190663447834
Prediction_local [1.85741649]
Right: 4
```



negative positive



LIME allows us to understand why and how our model made a decision based on its features

Trojan was the most difficult category to classify; we investigated why certain Trojan apps may have been misclassified.

We identified a subset of 4 apps that would heavily influence the classification of Trojan in a negative way.

These 4 apps were all Benign. The subcategories of these apps were Beauty and Productivity.

Conclusion and Future Work

Summary

- High Frequency in 1 Category + Low Frequency in Others -> Strong Classification Influence
- Multi-kernel learning performs better
- Kernels have weaknesses

Future Considerations

- Include more kernels
- Improve classification of trojans
- Use ranking algorithm to screen new apps before they get published

Work cited

http://yes-lab.org/files/HinDroid_KDD2017_Slides_Ye.pdf