MALICIA- Malware Classification

Manika Mittal and Ronak Sumbaly
University of California, Los Angeles

Background

As technology increasingly becomes an inseparable part of our lives, it has opened several new windows of opportunities for both people and criminals. Crime has shifted from the physical world to the virtual world, where the criminals exploit technology to launch attacks and harm people's privacy and data. This shift has amplified the effect of an attack, because now the attackers can attack from anywhere at any time and target much more people than it could have in the physical world. Several malware attacks have been witnessed lately, including attacks on Yahoo advertising server, Apple OS X, etc. This calls for an effective way to detect the malwares before they are launched so that appropriate steps can be taken to protect a system against such attacks.

Malicious Software (Malware) is a code snippet or an intrusive software which is written with an intent to disrupt computer operations, gather sensitive information from systems, inject computer viruses, trojans etc. While several schemes have been proposed to classify malwares into different families, most of them use a previously known "Malware Signature" for classification. In order to elude such detection mechanisms, the authors of malware obfuscate their code thus resulting in a destructive malware that continuously evolves to "look" different. Thus we need more effective ways of classifying and detecting malwares in order to protect people from being new victims of such attacks.

Problem Statement

Although code obfuscation techniques like Garbage Code Insertion and Instruction Permutation can elude the "signature-based" malware detection systems, at the lower level a malware, belonging to one family, has to generate similar opcodes and similar patterns because they perform the same functions. Our aim is to leverage the power of Machine Learning in order to extract these similar patterns and hence classify different malwares into their respective families. We focus mainly on feature extraction and feature selection and the main challenge was to accomplish acceptable accuracies while ensuring that the classification ran in a timely manner in order to detect malwares before an attack can be launched.

Data

MICROSOFT MALWARE CLASSIFICATION DATSET

Each malware data point comprises of two types of files:
- Hexadecimal BYTE files
- ASM files produced by IDA disassembler

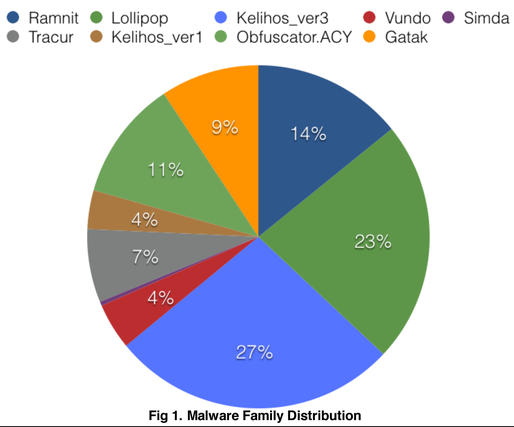

Fig 1. Malware Family Distribution

Methodology

Following steps were followed in order to achieve the results given on the right:

ITERATION 1 - Modeling based on Feature Extraction

- We started with 10GB of data, and partitioned it into 3 different folders: training, validation and testing datasets respectively.
- Next we extracted some basic features. Our basic feature set included a count of all the commonly occurring assembly language instructions.
- We then included some dynamic features which were derived from the files using feature-itemset techniques. The entire data was then normalized.
- Then we compared different classification techniques and plotted the accuracies (Fig 2). We observed that Random Forest gave the best accuracies.

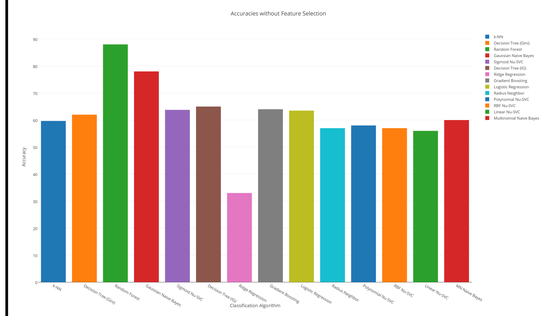

Fig 2. Classification Accuracies – Before Feature Selection

ITERATION 2 - Modeling after Feature Selection and Parameter Tuning for Dimensionality Reduction

- The features so far used were raw features and there was no feature selection performed on them yet. Since there were a lot of useless features in the feature set that were independent and uncorrelated, PCA could not eliminate those. So we set a threshold value that would weed out such features. This technique was again inspired by the confidence level that Apriori sets.
- After the above feature selection, we ran the same classification algorithms that we did before, to see a comparison. We saw a huge increase in the accuracies.
- Till now we were using the default parameters for classification, we then went ahead and used cross-validation in order to obtain the best parameters. This parameter tuning also resulted in an increase in the accuracies. The final graph is shown in Fig 2.

ITERATION 3 - Interpret Best Feature which has maximum contribution towards the model building process

- We then evaluated the performance of our feature selection by checking how the accuracies are affected by eliminating one feature at a time.
- After this we performed a detailed analysis of our results and observations which are summarized in the "Analysis" Section.

Results

RESULTS – ITERATION 2

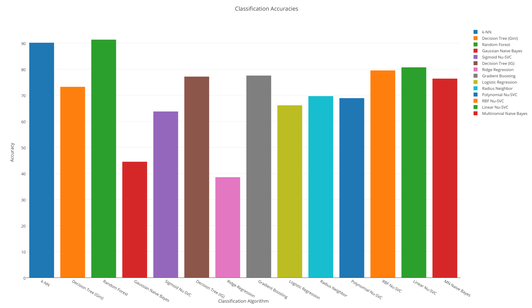

Fig 3. Classification Accuracies – After Feature Selection & Parameter Tuning

RESULTS – ITERATION 3

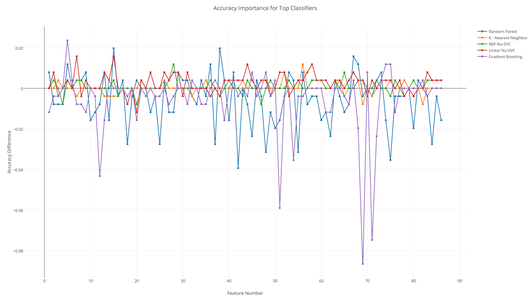

Fig 4. Accuracy Deviation – Top Classifiers

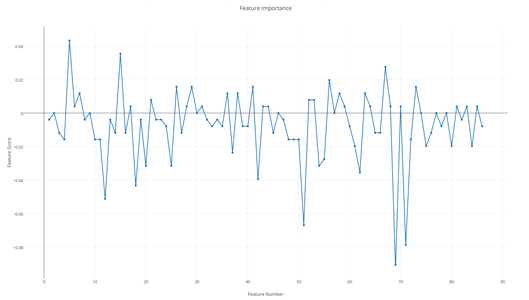

Fig 5. Accuracy Deviation – Cumulative

Analysis

ANALYSIS – ITERATION 1

- Our modeling uncovered that all our features could be segregated as either sparse or dense features.
- A large number of sparse features degraded the performance of most of the models since a lot of non-relevant features were being used for model construction.
- The innate randomness of curating dense nodes from subset of features gives Random Forest the robustness against over-fitting. Since the model is created using through dense features it gave the best performance even before feature selection.
- Apart from Random Forest all the other classifiers did not perform well before feature selection

ANALYSIS – ITERATION 2

After feature selection following observations was made.

ALGORITHM	ACCURACY	ANALYSIS
K – Nearest Neighbor	Increases	Redundant features which were causing distortion in the distance measure were removed.
Random Forests	Increases	The exact feature set varies in each run but the performance remained stable throughout.
Gradient Boosting	Increases	More accurate decision stumps created due to elimination sparse features.
Radius Neighbor	Increases	The neighbors within the optimum radius for each data point will only comprise of relevant features.
Multinomial Naive Bayes	Increase	Works well with discrete features, removal of useless features gave an accurate model.
Polynomial / RBF/ Linear Nu-SVC	Increase	High dimensional feature space had caused SVCs to suffer due to irrelevant features.
Regression	Remains Same	No significant analysis as regression is not suitable as per the nature of dataset.
Gaussian Naive Bayes	Decreases	Anomalous result which can't be attributed to the feature selection iteration.
Decision Tree	Inconsistent	Accuracy comparison futile as accuracies vary due to random feature selection.

ANALYSIS – ITERATION 3

- The elimination of most features resulted in a significant decrease in accuracies (valleys in Fig 4. & Fig 5.)
- The elimination of very few features (3) resulted in increase in accuracy (hardly by 0.05 %) which can be interpreted as noisy features.
- This analysis revealed that our feature selection resulted in 96.5% of good features and the noisy features did not affect the accuracy significantly
- ASM file instruction density features like **?? , 00** were found to be the best features

Conclusion

We successfully classified the malwares into their respective families with an accuracy of 91%. We achieved these results with minimal overhead of the algorithms used to extract, select and classify the features. Using Machine Learning we can classify the malwares into their respective families, even if obfuscation and polymorphism has been employed to change the "look and feel" of the malware.

It is essential that the machine learning techniques applied for this classification achieves the desired results before it's too late to protect a system against malwares.

Future Work

Other classification techniques like Deep Learning can be used, which we couldn't use because of hardware limitations. However, it should be noted that techniques like deep learning can take a lot of time to run and so efficient methods must be thought of in order to ensure timely detection of malwares.

There might still be tricks that can be employed to further reduce the speed of malware classification while still ensuring acceptable levels of accuracies.