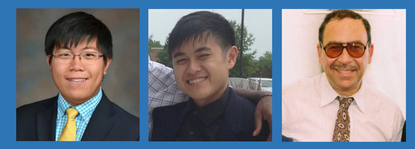
Application of Data Mining and Machine Learning to Detect and Remove Spyware

Richard R. Yang and Victor Kang
Advisor: Mohamed A. Zohdy



Objectives

Current spyware detection systems need to be updated with new behaviors to detect new, unknown, or adapted spyware. In addition, current systems may suffer from a high level of false positive and false negative rates in detection.

The goals of our project are to:

1. Apply data mining feature extraction on benign and spyware samples
2. Develop a hybrid machine learning approach:
   - Using supervised learning to classify files as spyware or benign
   - Using unsupervised learning to classify spyware files on a severity spectrum
3. Evaluate performance measures using Accuracy, ROC Curves, Sensitivity, and Specificity.

Introduction

Anti-malware solutions commonly employ the use of signature-based and heuristic-based detection to detect and remove spyware, which lacks the ability to detect new spyware or mutated traces of existing spyware.

Our solution is accomplished through the use of data mining and machine learning concepts and algorithms. We perform two types of data mining methods on samples, extracting n-grams and Portable Executable structures as features. The accuracy of each mining method is compared after applying supervised learning. In our proposed approach, a supervised learning algorithm is applied to classify if a file is considered spyware or benign. The files classified as spyware will then be categorized on a severity spectrum using an unsupervised algorithm.

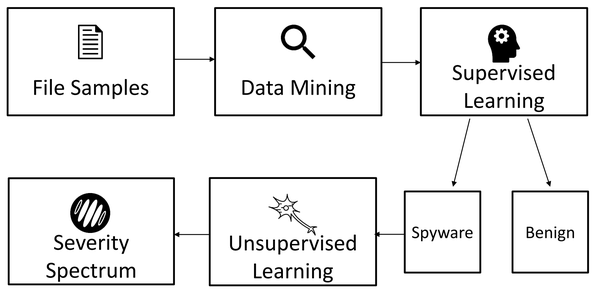


Figure 1. Flowchart of our research experimental process

Experimental Procedure

Data Mining:

The two general types of features we mined from each sample were n-grams and Portable Executable structure data.

N-grams are sequences of n bytes extracted from raw computer data as seen by the system memory. We "dump" the contents of the memory into a file and then sequence the data so that each segment is n bytes long.

The PE structure contains information about how the operating system manages a resources allocated to a program. Features from the PE Header, MZ Header, Data-Directory, and API function calls were extracted and compared from each sample's structure.

Machine Learning:

Machine learning can be split into two general techniques: supervised and unsupervised learning.

Our research introduces a new concept of hybrid machine learning, where we use both techniques.

In supervised learning, the computer is given a set of training data and how that data is classified. With that data, the computer builds a decision tree to decide how to classify the data. We select the best performing features by calculating the information gain when each node is split.

$$IG(T,a) = H(T) - H(T|a)$$

$$H = \sum_i -p_i \log_2 p_i$$

We use the self-organizing feature map as our unsupervised learning algorithm. The computer is not trained with a training set, but rather, it draws inferences from the correlated data by calculating the p-norm distance between the input and the node.

$$L_p = \left(\sum_{i=1}^{n} |q_i - p_i|^2\right)^{1/p}$$

Results and Analysis

To determine the effectiveness of our solution, we define the following evaluation metrics:

True Positive: spyware identified as spyware
False Positive: benign file identified as spyware
True Negative: benign file identified as benign
False Negative: spyware identified as benign
Sensitivity: true positive rate, $\mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$
Specificity: true negative rate, $\mathrm{TN}/(\mathrm{TN}+\mathrm{FN})$
ROC Curve: a graph of TPR vs. FPR

The overall accuracy of can be calculated by:

$$\frac{TP + TN}{TP + TN + FP + FN}$$

We compare the results of three supervised algorithms:

- *J48* is an implementation of the C4.5 algorithm in Java, which builds decision trees by splitting on the nodes that provide the highest normalized information gain.
- *Naïve Bayes* is a probabilistic classifier that applies Bayes' theorem with independent assumptions of each class.
- *Random Forest* is a collection of de-correlated decision trees that averages multiple models.

In each supervised algorithm, we also vary which features were used. Our best performing approach uses all of the features from the PE structure, in addition to the top 100 API function calls (selected through information gain).

Instead of calculating the Euclidean distance (2-norm) in the self-organizing feature map, we used a 6-norm calculation in our spectrum.

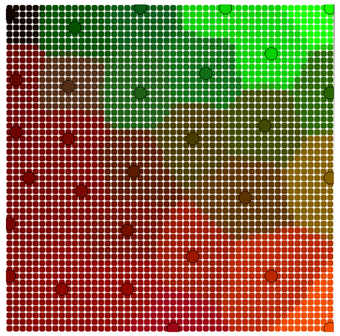
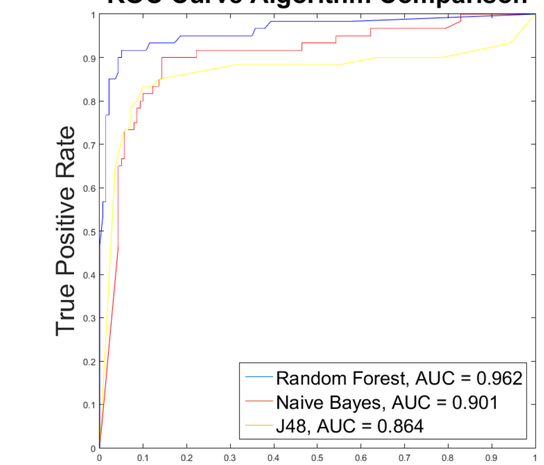
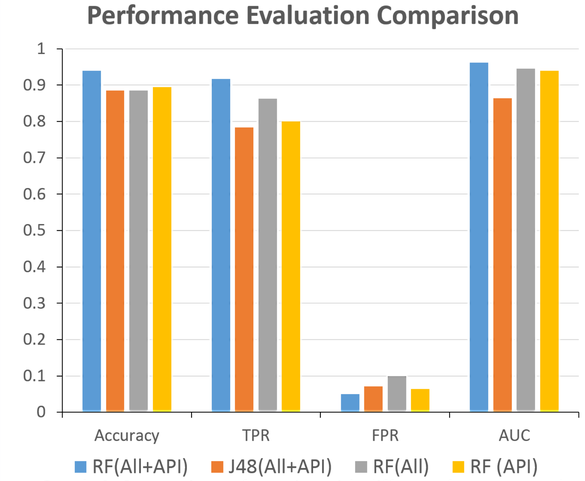


Figure 2. An output severity spectrum using 6-norm



Graph 3. Comparison of supervised algorithms with highest performing parameters



Graph 4. Comparison of metrics with different features used

Conclusions

In our research, we introduced a new method of hybrid machine learning to detect spyware and classify spyware on a severity spectrum. We explored the variations of parameters and their effect on the accuracy of detection. **It was discovered that the best performing solution uses features mined from the PE structure and API function calls selected through information gain applied with the Random Forest algorithm. We achieved a 94% overall accuracy with 91.7% TPR and 5% FPR.**

For future work, we suggest exploring other unsupervised learning algorithms to classify spyware, and additional depth in the self-organizing feature map algorithm.

This research work was conducted at Oakland University in the UnCoRe program - REU site supported by the National Science Foundation under Grant No. CNS-1460897. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.