The Making of a Malware Hound

Using Machine Learning Techniques to smell out familiar indicators of Malware Families

Mitchell Mays

Taylor University

Introduction

With highly motivated and intelligent attackers creating malware, many common defenses have become insufficient to protect against them. The common Anti-virus softwares are unable to guarantee protection against new malware, even if it has only been slightly modified from a similar known malware. With this in mind, research has turned to machine learning techniques as a possible solution. Instead of relying on specific malware signatures, machine learning attempts to identify particular features which indicate shared code or techniques between similar malware executables. These features can be gathered without running the executable which could allow the identification and removal of never before seen malware on a user's machine.

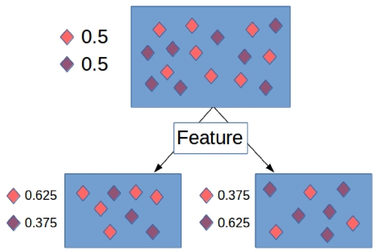

Figure 1. The information gain algorithm used to calculate the helpfulness of a possible N-gram feature.

Related Work

Schultz, Eskin, Zadok, and Stolfo [1] accomplished the first work which applied machine learning to malware. They looked at taking the executable files and finding features for classification. As with many developments in machine learning, a lot of effort initially was spent on feature discovery. Many components of the executable file were tested as features, including gathering n-grams from the byte files. Abou-Assaleh, Cercone, Keˇselj, and Sweidan [2] extracted these sequences of bytes and used their frequency within the file as a feature.

In more recent years, research in the topic has directed itself toward differing classifiers. New research has offered up an ensemble classifier [3, 4], a classification system which utilizes multiple classifiers to come to a final class decision.

Problem Statement

- Gather all possible N-gram features from assembly files and limit set to those useful for classifying
- Make image features from byte files
- Create multiple classifiers, which use different features to come at the problem from different angles.
- Combine the results of each classifier as an ensemble classifier for a final more accurate result.

Experimental Design

After gathering all possible n-grams, information gain (Figure 1) was used to pare down the resulting features. These were the input to the Feed-Forward Neural Network, while a byte-image was input into the convolutional neural network (Figure 3). Below is the structure of the experiment.

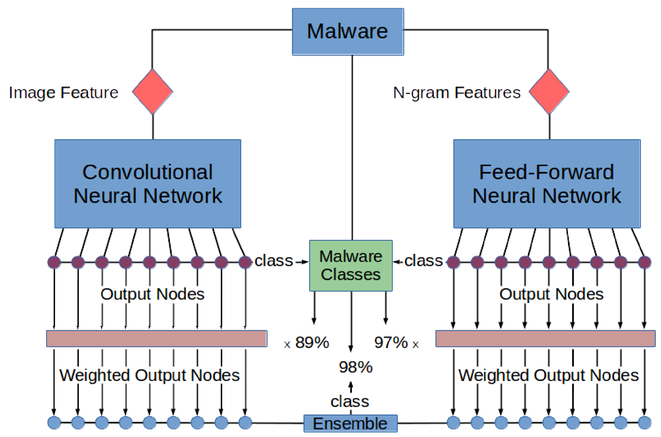

Figure 2. The testing structure of the classifiers. Performed with 10-fold cross validation. Outputs of each classifier are weighted and combined for a final more accurate classification through the ensemble classifier.

Results

After running the cross validations on both of the networks, I was left with a final average accuracy for each. The deep feed-forward neural network using n-gram features trained to be 96.7% accurate. The convolutional neural network, on the other hand, trained to 86.4% accuracy on the test data. By multiplying the nodes by these accuracies, I created weighted nodes. Combining these nodes as an ensemble classifier returned an accuracy of 97.7%.

Future Work

Due to the amount of time needed for gathering and assessing the 14 million possible n-grams, there was not much time for manipulating the classifiers. In the future, making changes to or adding new classifiers could make dramatic changes on the final accuracy of the system.

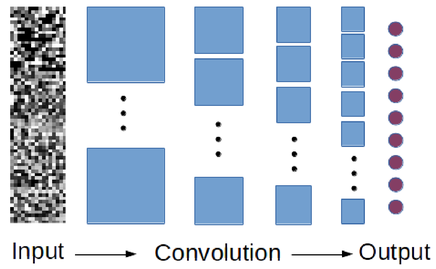

Figure 3. The convolutional neural network takes in byte data as a black and white pixel image and uses convolutions to produce a final classification.

Conclusion

The accuracy of the ensemble classifier was 98% on the 10,000 malwares. This value is one percent higher than the best individual classifier in the coalition (97% percent from the feed-forward neural network). Yi-Bin, Shu-Chang Din, Chao-Fu Zheng, and Bai-Jian Gao [3] showed that simpler classifiers such as support vector machines and decision trees could be used as components to build an improved ensemble classifier. My work shows that neural networks can be used as constituents to an ensemble classifier and still create an improved classification.

Acknowledgements

Many thanks to Dr. Brandle for his advising throughout the process, and to Dr. White for help with the information gain algorithm.

[1]. Schultz, M.g., E. Eskin, F. Zadok, and S.j. Stolfo. "Data Mining Methods for Detection of New Malicious Executables."

[2]. Abou-Assaleh, T., N. Cercone, V. Keselj, and R. Sweidan. "N-gram-based Detection of New Malicious Code."

[3]. Lu, Yi-Bin, Shu-Chang Din, Chao-Fu Zheng, and Bai-Jian Gao. "Using Multi-Feature and Classifier Ensembles to Improve Malware Detection."

[4]. Kolter, Jeremy Z., and Marcus A. Maloof. "Learning to Detect Malicious Executables in the Wild."