Journal of Machine Learning Research 7 (2006) 2669-2672 Submitted 12/06; Published 12/06

Machine Learning for Computer Security*

Philip K. Chan PKC@CS.FIT.EDU
Department of Computer Sciences
Florida Institute of Technology
Melbourne, FL 32901, USA

Richard P. Lippmann LIPPMANN@LL.MIT.EDU
Lincoln Lab, MIT
244 Wood Street
Lexington, MA 02173, USA

Editors: Philip K. Chan and Richard P. Lippman

Abstract

The prevalent use of computers and internet has enhanced the quality of life for many people, but it has also attracted undesired attempts to undermine these systems. This special topic contains several research studies on how machine learning algorithms can help improve the security of computer systems.

Keywords: computer security, spam, images with embedded text, malicious executables, network protocols, encrypted traffic

1. Introduction

As computers have become more ubiquitous and connected, their security has become a major concern. Attacks are more pervasive and diverse—they range from unsolicited email messages that can trick users in providing personal information to dangerous viruses that can erase data and shut down computer systems. Consequently, security breaches are not rare topics in the news.

Conventional security software requires a lot of human effort to identity threats, extract characteristics from the threats, and encode the characteristics into software to detect the threats. This labor-intensive process can be more efficient by applying machine learning algorithms. As a result, a number of researchers have investigated various machine learning algorithms to detect attacks more efficiently and reliably. Two edited books (Barbara and Jajodia, 2002; Maloof, 2006) have been published and two workshops at research conferences (Chan et al., 2003; Brodley et al., 2004) have been conducted in recent years. Due to the level of interest from the researchers and maturity of some of their studies, we decided to organize a special topic on "Machine Learning for Computer Security" for this journal.

*. This work is sponsored by the U.S. Air Force under Air Force Contract FA 8721-05-C-0002. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government.

Figure 1: Adversarial spam image designed to defeat OCR text extraction

2. JMLR Special Topic

We received nineteen submissions for this special topic. After considering the reviews for each submission, we selected four papers to be included in this special topic.

Bratko et al. (2006) describe a recent advance in the ongoing battle between those that generate and those that want to block unwanted spam email. They apply adaptive statistical compression algorithms (Dynamic Markov Compression (DMC) and Prediction by Partial Matching (PPM)) to build models for email messages. DMC learns a Markov model incrementally via a cloning technique to introduce new states in the model. PPM learns a table of contexts and the frequency of the symbol following the contexts. To classify if a message is spam, they use minimum cross entropy (MCE) and minimal description length (MDL). MCE calculates the number of bits to encode a message based on competing models learned from normal and spam messages, and classifies the message to the class whose model requires fewer encoded bits. MDL measures the additional number of bits needed to encode a message after adding it to the competing models, and classifies the message to the class whose model needs fewer additional bits. The authors evaluated their techniques on three datasets and against six open source spam filters. For both DMC and PPM models, they found that MDL yields lower 1-AUC (1 - Area under ROC) than MCE. They also reported that DMC consistently outperforms the six open source spam filters.

Similar to Bratko et al. (2006), Fumera et al. (2006) tackle the problem of spam email, however, they consider spam messages with embedded images. They developed an approach to analyze spam email when spam text messages are embedded in attached images instead of in the text email body (for example, Figure 1). Standard optical character recognition (OCR) software is used to extract words embedded in images and these extra words are used in addition to text in the email header and body to improve performance of a support vector machine spam classifier. At a false alarm rate of 1%, this technique often reduced the miss rate by a factor of two for spam email that contained embedded images. Evidently, this approach has been adopted by commercial spam filtering companies. Spammers have reacted by adding varied background and distorting text embedded in images

to make it difficult for OCR systems to extract spam messages but easy for humans to interpret these messages. Figure 1 shows an example of an image from a recent spam email suggesting a stock to purchase. This paper illustrates that pattern classification techniques can be effective for complex problems such as spam, but that it can be difficult to obtain a long-standing advantage in adversarial environments.

Instead of email messages, Kolter and Maloof (2006) analyze executables. They demonstrate that N-gram analysis of executables can be used to distinguish between normal computer programs and malicious virus, worm, and Trojan horse programs. Even though roughly 20% of the malicious software samples used were obfuscated with either compression or encryption, detection accuracy for 291 previously unseen malicious executables was roughly 98% correct at a false alarm rate of 5%. These good results were made possible by collecting and carefully confirming and labeling a training corpus of 1971 benign and 1651 malicious executables and using 10-fold cross-validation to select both the top-performing N-grams and the best performing classifier which in this case was a boosted tree classifier.

As network traffic is increasingly encrypted, Wright et al. (2006) address the problem of inferring application protocol behaviors in encrypted traffic to help intrusion detection systems. The authors first propose using k-nearest neighbor methods for identifying protocols in data instances, each of which is known to belong to one protocol. Experiments on eight protocols indicate 75-100% true detection rate. They then propose Hidden Markov Models (HMMs) for identifying protocols in traffic with *mixed* protocols. Each protocol has an HMM model. Each model has a group of states (a pair of client and server states, and a pair of insert and delete states) and the number of groups is equal to the average number of packets in a connection for the protocol. The emitting symbols are codewords for packet size and inter-arrival time. To classify, they pick the protocol, whose model has the best Viterbi path that explains the observation. Their empirical results indicate that HMMs can achieve 58-87% true detection rate on eight protocols. They last propose methods for identifying the number of connections in encrypted *tunnels*. They assume the number of connections is Gaussian and the number of packets of a certain type is Poisson. Each HMM state corresponds to a connection count, the output is a tuple of counts for the different types of packets. They use the Gaussian and Poisson assumptions to estimate standard deviations of the number of connections and rates of each packet type. The number of connections at a certain time is predicted by the most probable state at that time. They evaluated their techniques on four tunnels.

3. Concluding Remarks

These four papers demonstrate the need for carefully constructed training and test corpora, effective feature extraction and selection, and valid evaluations on representative corpora when applying pattern classification to computer security problems. They also suggest a new important direction for pattern classification research. This is to develop approaches that provide sustained good performance in adversarial environments where a malicious adversary takes actions to subvert a classifier. Some of these actions could be: (1) obscure important discriminating input features, for example by modifying text in images to be difficult for an OCR to extract, (2) add extraneous additional features to make an input appear more normal, for example by adding sentences extracted from normal emails to the end of spam emails, (3) alter the prior probabilities of abnormal inputs, and (4) take all of these actions over time in a way designed to thwart systems that learn and adapt over time. The paper on spam detection (Fumera et al., 2006) mentions this problem and another recent paper

(Newsome et al., 2006) shows how an adversary can defeat a system that learns to automatically extract signatures to detect computer worms. Further research is needed to determine if there are any systematic approaches that can lead to classifiers that are more robust in adversarial environments.

Acknowledgments

We would like to acknowledge the time and effort of the reviewers: Wei Fan (IBM Watson Research Center), Anup Ghosh (DARPA), Tom Goldring (NSA), Sushil Jajodia (George Mason University), Chris Kruegel (Technical University Vienna), Vipin Kumar (University of Minnesota), Terran Lane (University of New Mexico), Wenke Lee (Georgia Institute of Technology), Matthew Mahoney (Florida Institute of Technology), Roy Maxion (Carnegie Mellon University), Chris Michael (Cigital), Srinivasan Parthasarathy (Ohio State University), R. Sekar (Stony Brook University), Jude Shavlik (University of Wisconsin), Marius Silaghi (Florida Institute of Technology), Salvatore Stolfo (Columbia University), and Alfonso Valdes (SRI). Their diligence makes this special topic a reality.

References

D. Barbara and S. Jajodia, editors. *Applications of Data Mining in Computer Security*. Kluwer, 2002.

A. Bratko, B. Filipic, G. Cormack, T. Lynam, and B. Zupan. Spam filtering using statistical data compression models. *Journal of Machine Learning Research*, 7:2673–2698, 2006.

C. Brodley, P. Chan, R. Lippmann, and B. Yurcik, editors. *Workshop Notes of Visualization and Data Mining for Computer Security*, 2004. ACM Intl. Conf. Computer and Communications Security (CCS).

P. Chan, V. Kumar, W. Lee, and S. Parthasarathy, editors. *Workshop Notes of Data Mining for Computer Security*, 2003. IEEE Intl. Conf. Data Mining (ICDM).

G. Fumera, I. Pillai, and F. Roli. Spam filtering based on the analysis of text information embedded into images. *Journal of Machine Learning Research*, 7:2699–2720, 2006.

J. Kolter and M. Maloof. Learning to detect and classify malicious executables in the wild. *Journal of Machine Learning Research*, 7:2721–2744, 2006.

M. Maloof, editor. *Machine Learning and Data Mining for Computer Security*. Springer, 2006.

J. Newsome, B. Karp, and D. Song. Paragraph: Thwarting signature learning by training maliciously. In D. Zamboni and C. Kruegel, editors, *Recent Advances in Intrusion Detection (RAID) 2006 (LNCS 4219)*, pages 81–105, Berlin, 2006. Springer-Verlag.

C. Wright, F. Monrose, and G. Masson. On inferring application protocol behaviors in encrypted network traffic. *Journal of Machine Learning Research*, 7:2745–2769, 2006.