On the Efficient Mining of Network Audit Data using Genetic Programming

Song D., Curry R., Heywood M.I., Zincir-Heywood A.N.

Dalhousie University, Faculty of Computer Science
6040 University Avenue, Halifax, NS. B3H 1W5 Canada
{mheywood, zincir}@cs.dal.ca

Abstract. Anomaly detection is often performed using models derived from off-line analysis of network audit data. Such datasets are typically very large. A method for efficiently applying GP to such audit data is presented in which training times for datasets with 500,000 exemplars is completed in 15 minutes. Six basic session features are demonstrated to be sufficient for detecting 95.15% Denial of Service attacks and 53.1% of Probe attacks in the DARPA 98 Intrusion Detection benchmark.

1 Introduction

Anomaly detection represents a widely used form for Intrusion Detection Systems (IDS) in which the principle objective is to distinguish between normal and abnormal behavior. Naturally, there has been considerable interest in such systems from machine learning practitioners. However, in order to build a practical anomaly detector (from audit data) it is not only important to achieve an acceptable detection and false positive rate, but also to retain concise solutions (i.e. efficient run time operation) and have an efficient training algorithm (i.e. timely solutions are required on large datasets, the content of which frequently change as new exploits are discovered). Genetic Programming is widely observed to both suffer from code bloat and have a computationally expensive inner loop. The code bloat problem is typically considered a run time problem, with a significant amount of simplification being available once the evolutionary cycle completes. The computational requirements of the inner loop (on large datasets) are classically addressed through the use of hardware solutions.

This paper begins by introducing a family of hierarchical Dynamic Subset Selection (DSS) algorithms of increasing complexity. Performance is then assessed in terms of the accuracy achieved whilst utilizing only the six simplest features. That is to say, the KDD cup 1999 competition utilized a feature set resulting from a knowledge driven data mining activity in which candidate features were designed to explicitly highlight behaviors considered to be representative of specific attacks [1]. In this work we detail performance for the case of detectors built on six basic session features alone. Previous works considering either all 41 features [1], [2] or the first 8 'basic' session features alone [3].

2 Hierarchical DSS algorithms

The basic hierarchical DSS algorithm divides the dataset into a series of roughly equal partitions sufficiently small to fit in the available RAM of the target computing platform. Such blocks may be selected with uniform probability [3] or in proportion to their 'difficulty' [4]. The exemplars from the block are then subsampled in proportion to their respective difficulty and age, as per the DSS algorithm [5], [3]. A further variant is introduced here. In the case of many datasets the distribution of class exemplars is not uniform, thus some classes are represented excessively and others very infrequently. This is typically the case with the anomaly detection problem on computer networks. Attacks – such as those for Denial of Service – will be very frequent (the basic objective being to overload the target network node) whilst others – such as a probe or a User to Root (U2R) attack – will be very infrequent, possibly only accounting for a fraction of a percent of the overall audit data. One approach to this problem might be to stratify the data such that each block had the same distribution as the original data. Unfortunately when exemplar classes are very rare this scheme also fails. Instead the data is organized first in terms of the corresponding attack category. Thus all instances of a DoS attack appear together, all instances of U2R appear together etc. A block is now composed from a fixed partition of each exemplar class, where such partitions are selected in proportion to their difficulty and age. The net effect is that every block consists of a balanced set of exemplars, independent of the initial exemplar data distribution.

3 Results

As indicated above, one of the first approaches to this problem was to create features specific to one of four attack categories (DoS, Probe, U2R and R2L) [1]. This method naturally required a significant degree of *a priori* knowledge in order to guide the data mining process. The result of this process was 41 features of four basic categories: Intrinsic session; Content (specific to U2R and R2L attacks); and two forms of temporal features. Intrinsic session features effectively took the original packet based DARPA 98 dataset and re-expressed this as session information. The first six of these features are completely independent of specific instances of attack.

In this work we report results using detectors based on the first six intrinsic session features alone. In order to provide the basis for representing temporal dependencies, a shift register is utilized in which 8 taps at every 8^{th} location provide the only source of data. Naturally, content-based attacks will be very difficult to identify. Performance is summarized in terms of median Detection and False Positive Rates and category specific detection rates on the 'corrected' KDD test set, Table 1 (14 attack types unseen during training and statistically different distribution of the attack types [2]). Detection rates of both systems are basically equivalent, however, the DSS-DSS algorithm consistently returns a 1-2% improvement in the False Positive rates. From the category specific detection rates it is apparent that this improvement is distributed across all classes and not associated with a single improvement to one class alone. Naturally, content-based attacks (U2L and R2L) are rarely detected, however, it is interesting to

note that some of these attacks are recorded through temporal behavioral properties (i.e. guessing a password is demonstrated through repeated log in attempts).

Table 1. Median Detection, False Positive Rates and Category Specific Detection Rates on Test

Algorithm	Train (10% KDD)		Test (Corrected) KDD	
	Detection	False Positive	Detection	False Positive
RSS-DSS	98.4	3.64	88.98	3.83
SSS-DSS	98.9	1.3	88.8	1.7
RSS-DSS (Corrected KDD Test Set – Category Specific Detection rates)				
Algorithm	Normal	DoS	Probe	U2R	R2L
RSS-DSS	96.76	95.18	52.5	10.31	2.36
SSS-DSS	97.94	95.78	53.1	9.65	2.61

4 Conclusion

The capacity for developing anomaly detection systems efficiently from audit data using GP is demonstrated. Six basic session features are demonstrated to be sufficient for detecting 95.15% Denial of Service attacks and 53.1% of Probe attacks. The hierarchical DSS algorithms central to achieving this are generic, and therefore applicable to a wide class of machine learning algorithms, both of a supervised and unsupervised nature.

Acknowledgements

The support of IRIS Emerging Opportunities, CFI New Opportunity and NSERC Discovery Grants from the Canadian Government are gratefully acknowledged.

References

1. Gathercole C., Ross P.: Dynamic Training Subset Selection for Supervised Learning in Genetic Programming. PPSN III. LNCS 866 (1994) 312-321.
2. Elkan C.: Results of the KDD-99 Classifier Learning Contest. SIGKDD Explorations. ACM SIGKDD 1(2). (2000) 63-64.
3. Song D., Heywood M.I., Zincir-Heywood A.N.: A Linear Genetic Programming Approach to Intrusion Detection. GECCO'03, Cantu-Paz E., *et al*. (eds), LNCS 2724 (2003) 2325-2336.
4. Lee S., Stolfo S.J.: A framework for constructing features and models for Intrusion Detection Systems. ACM Transactions on Information and System Security 3(4) (2000) 227-261.
5. Curry R., Heywood M.I.: Towards Efficient Training on Large Datasets for Genetic Programming. Canadian Conference on Artificial Intelligence. LNAI (2004) to appear.