Applying Local Search and Genetic Evolution in Concept Learning Systems to Detect Intrusion in Computer Networks

Mirko Mischiatti and Filippo Neri

DSTA - University of Piemonte Orientale
Corso Borsalino 54, 15100 Alessandria (AL), Italy
Tel:+39 011 6706783, Fax:+39 011 751603
Email: mischiatti@yahoo.it, neri@mfn.unipmn.it

Keywords: Intrusion detection, symbolic concept learning, genetic algorithms

Abstract. The detection of intrusions over computer networks (i.e., network access by non-authorized users) can be cast to the task of detecting anomalous patterns of network traffic. In this case, models of normal traffic have to be determined and compared against the current network traffic.
We compare models of network traffic acquired by a system based on a distributed genetic algorithm with the ones acquired by a system based on greedy heuristics. Also we show that representation change of the network data can result in a significant increase in the classification performances of the traffic models.
Network data made available from the Information Exploration Shootout project has been chosen as experimental testbed[1].

1 INTRODUCTION

The raise in the number of computer break-ins, virtually occurring at any site, determines a strong request for exploiting computer security techniques to protect the site assets. A variety of approaches to intrusion detection do exist [Denning, 1987]. Some of them exploit signatures of known attacks for detecting when an intrusion occurs. They are thus based on a model of virtually all the possible misuses of the resource. The completeness request is actually a major limit of this approach [Kumar and Spafford, 1994].

Another approach to intrusion detection tries to characterize the normal usage of the resources under monitoring. An intrusion is then suspected when a significant shift from the resource's normal usage is detected. This approach seems to be more promising because of its potential ability to detect unknown intrusions. However, it also involves major challenges because of the need to acquire a model of the normal use general enough to allow authorized users to work without raising alarms, but specific enough to recognized unauthorized usages [Lane and Brodley, 1997,Ghosh et al., 1999,Lee et al., 1999,Neri, 2000].

[1] An extended and revised version of this paper will appear in the Proceedings of ECML2000

Our approach follows the last philosophy for detecting intrusion and we describe here how it is possible to learn a model of normal use of a network from logs of the network activity. A distributed genetic algorithm REGAL [Giordana and Neri, 1995,Neri and Saitta, 1996] is exploited for mining the network logs searching for interesting traffic patterns.

We are well aware that many aspects of deploying in practice learning system to acquire useful traffic patterns are still open including: selecting or building informative data representations, improving recognition performances (i.e., reducing both the rate of false alarms and of undetected intrusions), representing the traffic models for real world deployment (real-time classification of packets), and dealing with the shift in the patterns of normal use of the resources [Lane and Brodley, 1998].

We concentrate here on the first two issues and we report our findings concerning the impact of different learning methods and of alternative data representation, with respect to the ones used in previous works, on the detection performances. As learning methods, we exploited two rule based systems: a heuristic one, RIPPER [Cohen, 1995], and an evolutive one (based on genetic algorithms), REGAL [Giordana and Neri, 1995,Neri and Saitta, 1996]. The first system has been selected because of its previous use [Lee et al., 1999]; it will thus act as benchmark. The second system has been selected because we believe that its intrinsically stochastic behavior should allow the acquisition of alternative robust and simpler models [Neri and Saitta, 1996].

In the following, a description of the systems REGAL and RIPPER (Section 2) and of the experiments performed in the Information Exploration Shootout (IES) (Section 3) are reported. Finally, the conclusions are drawn.

2 THE SYSTEMS REGAL AND RIPPER

For space reason we will provide here an abstract description of both learning systems REGAL and RIPPER as their full descriptions have already been published.

REGAL [Giordana and Neri, 1995,Neri and Saitta, 1996] is a learning system, based on a distributed genetic algorithm (GA). It takes as input a set of data (training instances) and outputs a set of symbolic classification rules characterizing the input data. As usual, learning is achieved by searching a space of candidate classification rules. The language L used to represent classification rules is a Horn clause language in which terms can be variables or disjunctions of constants, and negation occurs in a restricted form [Michalski, 1983]. An example of an atomic expression containing a disjunctive term is *color(x,[yellow, green])*, which is semantically equivalent to *color(x,yellow) or color(x,green)*. Such formulas are represented as bitstrings that are actually the population individuals processed by the GA. Classical genetic operators, operating on binary strings, with the addition of task oriented *specializing* and *generalizing* crossovers are exploited, in an adaptive way, inside the system (for details see [Giordana and Neri, 1995].

In summary, REGAL is a distributed genetic algorithm that effectively combines the Theory of Niches and Species of Biological Evolution together with parallel processing. The system architecture is made by a set of extended Simple Genetic Algorithms

(SGA) [Goldberg, 1989], which cooperates to sieve a description space, and by a Supervisor process that coordinates the SGAs efforts by assigning to each of them a different region of the candidate rule space to be searched. In practice this is achieved by dinamically devising subsets of the dataset to be characterized by each SGA. In other words, REGAL does include a form of meta-learning, i.e. the ability to combine several classifiers into a single one. Such a form of meta-learning can be easily realized by means of a cooperative genetic algorithms [Potter et al., 1995,Giordana and Neri, 1995]. That is exactly what the Supervisor process does.

The system RIPPER [Cohen, 1995] is based on the iterated application of a greedy heuristic, similar to the Information Gain measure [Quinlan, 1993], to build conjunctive classification rules. At each iteration, those training instances correctly classified by the found rules are removed and the algorithm concentrate on learning a classification rule for the remaining one. The system outputs an ordered list of classification rules (possibly associated to many classes) to be applied in that same order to classify a new instance. An interesting features of the method is that it exploits on-line rule pruning while incrementally building a new classification rule to avoid overfitting.

3 INTRUSION DETECTION IN THE INFORMATION EXPLORATION SHOOTOUT CONTEST

An evaluation of REGAL over an intrusion detection task, by exploiting data from the Information Exploration Shootout Project (IES), is reported in this section. The IES made available network logs produced by 'tcpdump' for evaluating data mining tool over large set of data. These logs were collected at the gateway between an enterprise LAN and the outside-network (Internet). In the IES context, detecting intrusions means to recognize the possible occurrence of unauthorized ('bad') data packets interleaved with the authorized ('good') ones over the network under monitoring. The IES's project makes available four network logs: one is guarantee not to contain any intrusion attempts, whereas the other ones do include both normal traffic and intrusions attempts. In the IES context, no classification for each data packets is requested, instead an overall classification of a bunch of the network traffic, as containing or not attacks, is desired.

An approach to intrusion detection, based on anomaly detection, has been selected. We proceed as follows. IES data can be partitioned, on the base of their IP addresses, into packets exiting the reference installation (Outgoing), entering the installation (Incoming) and broadcasted from host to host inside the installation (Interlan). Three models of the packet traffic, one for each direction, have been built from the intrusion-free dataset. Then, these models have been applied to the three datasets containing intrusions. We expect to observe a significant variation in the classification rate between intrusion-free logs and logs containing intrusions because of the *anormal* characteristics of the traffic produced by the intrusive behavior. If this would actually occur, we could assert that the learned traffic models correctly capture the essential characteristics of the intrusion-free traffic. Experiments have been performed both with RIPPER and REGAL.

When RIPPER is applied to the IES data, the classification rate appearing in Table 1 becomes evident [Lee et al., 1999]. This results have been obtained by applying

RIPPER to the data as available from the tcpdumped files (see Appendix A). No preprocessing over the data, such as feature construction, has been applied. The experimental findings shows that the acquired models do not exhibit very different classification rate when applied to logs containing intrusions with respect to intrusion-free logs. These findings may suggest that the exploited data representation is too detailed with respect to the capability of the learning system. In turn, this causes the learned models to miss the information characterizing intrusion-free traffic.

Table 1. Experimental results of applying RIPPER to IES datasets using the raw data representation.

Dataset	interlan	incoming	outgoing
normal	0.04	0.04	0.04
intrusion1	0.23	0.07	0.04
intrusion2	0.09	0.07	0.05
intrusion3	0.08	0.14	0.04

Table 2. Experimental results of applying RIPPER to IES datasets using a compressed data representation.

Dataset	interlan	incoming	outgoing
normal	0.02	0.05	0.04
intrusion1	0.11	0.11	0.21
intrusion2	0.03	0.13	0.12
intrusion3	0.11	0.21	0.12

Table 3. Experimental results of applying REGAL to IES datasets using a compressed data representation.

Dataset	interlan	incoming	outgoing
normal	0.02	0.04	0.04
intrusion1	0.12	0.15	0.11
intrusion2	0.06	0.11	0.12
intrusion3	0.12	0.15	0.11

Following this observation, we develop a more compact representation for the packets that consists in mapping a subset of feature's values into a single value, thus reducing the cardinality of possible features values.

As an instance of reducing the range of the feature values, considers that the feature 'srcport' (see Appendix A for a description) may virtually assume any integer number from 0 to 65536. Also, the feature 'op' may assume hundreds of discrete values. Taking

into account basic knowledge about the domain, we manually developed the reduction mapping shown in Table 4. This mapping is not to be considered as the best one but

Original Value	New Value
0≤srcport<50	srcport=0
50≤srcport<100	srcport=0
<... skipped test ... >	<... skipped text ... >
srcport>20000	srcport=10
<... skipped text ... >	<... skipped text ... >
op contains ”DF”	op=1
op contains ”NXDomain”	op=2
op contains ANY OTHER VALUE	op=3

Table 4. Compression mapping applied when dealing with IES network data.

as a proof that a simple reduction of the feature values may positively impact over the recognition capabilities.

Exploiting this representation, RIPPER's performances become the ones reported in Table 2 and REGAL's performances exploiting the same compact data representation appear in Table 3. The observed figures show a more stable classification behavior of the models across different traffic conditions. Also a more distinct classification performance between the intrusion-free log and the logs including intrusions is evident. A compression-based representation is then a valuable way of increasing classification performances without introducing complex feature that may involves additional processing overhead. An evaluation of the effect caused by the addition of complex features to the raw network data representation has been performed in [Lee et al., 1999].

```
IF    srcprt(x,[[0,20],[40,100],[150,200],[>500]]) and
      dstprt(x,[>1024]) and flag(x,[FP,pt]) and
      seq1(x,[[100,150],[200,300],[500,5000],[>10000]]) and
      seq2(x,[[50,100],[200,300],[500,20000]]) and
      ack(x,[[0,3000],[5000,10000]]) and
      win(x,[[0,2000],[>3000]]) and
      buf(x,[<=512])
THEN  IncomingPacket(x)
Coverage: (Interlan, Incoming, Outgoing) = (0, 7349, 0)
```

Fig. 1. Example of a rule characterizing part of the incoming traffic. The rule describes 7349 incoming packets without confusing them with any outgoing or interlan packet.

For the sake of clarity, an example of rule characterizing intrusion-free Incoming packets, learned by REGAL, appears in Figure 1. The Incoming packets are characterized in term of the values of the features from their TCP/IP header. This rule successfully covers 7349 Incoming packets without being fooled by any Interlan or Outgoing ones. A description of the predicates appearing in the rule is provided in Appendix A.

4 CONCLUSIONS

We started to investigate the potentiality of two concept learners to the modeling of network data for detecting intrusion. Different set-ups to deal with detecting intrusions have been explored. In particular, we analyzed a packet representation exploiting compression of the feature's values in the effort to reduce the complexity of learning models of the traffic. We believe this being an important requisite for the automatic modeling and the on-line deployment of intrusion detection system.

The experimental results seems to support the use of a compression in the feature values as a method to increase detection performances while avoiding the use of derived and complex features whose computation results in a costly overhead.

A Appendix. The Information Exploration Shootout raw data representation

The IES data (available on line at http://iris.cs.uml.edu) have been collected by means of the TCPDUMP utility. Taking into account privacy concerns, the data portiong of each packet has been dropped. For each packet in the datasets the following attributes are available:
time - converted to floating pt seconds .. hr*3600+min*60+secs.
addr and port - (just get rid of x.y.256.256.port) The first two fields of the src and dest address make up the fake address, so the converted address was made as: x + y*256.
flag - added a "U" for udp data (only has ulen) X - means packet was a DNS name server request or response. The ID# and rest of data is in the "op" field. (see tcpdump descrip.) XPE - means there were no ports... from "fragmented packets".
seq1 - the data sequence number of the packet.
seq2 - the data sequence number of the data expected in return.
buf - the number of bytes of receive buffer space available.
ack - the sequence number of the next data expected from the other direction on this connection.
win - the number of bytes of receiver buffer space available from the other direction on this connection.
ulen - if a udp packet , the length.
op - optional info such as (df) ... do not fragment.
Particular attention has to be taken when dealing with fields like 'op' that contains a large amount of values.

References

[Cohen, 1995] Cohen, W. (1995). Fast effective rule induction. In *Proceedings of International Machine Learning Conference 1995*, Lake Tahoe, CA. Morgan Kaufmann.

[Denning, 1987] Denning, D. (1987). An intrusion detection model. *IEEE Transaction on Software Engineering*, SE-13(2):222–232.

[Ghosh et al., 1999] Ghosh, A., Schwartzbard, A., and Schatz, M. (1999). Learning program behavior profiles for intrusion detection. In *USENIX Workshop on Intrusion Detection and Network Monitoring*. USENIX Association.

[Giordana and Neri, 1995] Giordana, A. and Neri, F. (1995). Search-intensive concept induction. *Evolutionary Computation*, 3 (4):375–416.

[Goldberg, 1989] Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, Ma.

[Kumar and Spafford, 1994] Kumar, S. and Spafford, E. (1994). A pattern matching model for misuse detection. In *National Computer Security Conference*, pages 11–21, Baltimore.

[Lane and Brodley, 1997] Lane, T. and Brodley, C. (1997). An application of machine learning to anomaly detection. In *National Information Systems Security Conference*, Baltimore.

[Lane and Brodley, 1998] Lane, T. and Brodley, C. (1998). Approaches to online learning and conceptual drift for user identification in computer security. Technical report, ECE and the COAST Laboratory , Purdue University, Coast TR 98-12.

[Lee et al., 1999] Lee, W., Stolfo, S., and Mok, K. (1999). Mining in a data-flow environment: experience in network intrusion detection. In *Knowledge Discovery and Data Mining KDD'99*, pages 114–124. ACM Press.

[Michalski, 1983] Michalski, R. (1983). A theory and methodology of inductive learning. In Michalski, R., Carbonell, J., and Mitchell, T., editors, *Machine Learning, an Artificial Intelligence Approach*, volume I, pages 83–134. Morgan Kaufmann, Los Altos, CA.

[Neri, 2000] Neri, F. (2000). Mining tcp/ip traffic for network intrusion detection by using a distributed genetic algorithm. In *Proc. of European Conference on Machine Learning*, page In press, Barcelona, Spain. Springer-Verlag.

[Neri and Saitta, 1996] Neri, F. and Saitta, L. (1996). Exploring the power of genetic search in learning symbolic classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-18:1135–1142.

[Potter et al., 1995] Potter, M. A., Jong, K. A. D., and Grefenstette, J. J. (1995). A coevolutionary approach to learning sequential decision rules. In *Sixth International Conference on Genetic Algorithms*, pages 366–372, Pittsburgh, PA. Morgan Kaufmann.

[Quinlan, 1993] Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann, California.