Genetic Algorithm Optimization of Membership Functions for Mining Fuzzy Association Rules

Wengdong Wang
Susan M. Bridges
Department of Computer Science
Mississippi State University
{ww4, bridges}@cs.msstate.edu

1. INTRODUCTION

A large number of papers have been published regarding the combination of *fuzzy logic* (FL) and *genetic algorithms* (GA) (Herrera and Verdegay 1999). Fuzzy logic is a useful tool for modeling complex systems and deriving useful fuzzy relations or rules (Okamura et. al. 1998). However, it is often difficult for human experts to define the fuzzy sets and fuzzy rules used by these systems. GAs have proven to be a useful method for optimizing the membership functions of the fuzzy sets used by these fuzzy systems (Nawa et. al. 1998).

Luo (1999) has developed a method for mining fuzzy association rules for intrusion detection. This method is a fuzzy extension of the techniques used by Lee, Stolfo and Mok (1999) in which one compares the similarity of association rules mined from audit data when there are no intrusions with rules mined from audit data when there are intrusions. In this paper, we report on the use of genetic algorithms to tune the membership functions of the fuzzy variables used to mine the fuzzy association rules in order to improve the performance of the intrusion detection system. The evaluation functions used in our system differ from those used in fuzzy classification systems (Chen 1998) or fuzzy control systems (Tang et. al. 1998); our goal is to maximize the similarity of sets of *normal* rules and minimize the similarity of *normal* and *abnormal* rule sets. Non-intrusion is considered here as a *normal* situation and intrusion as an *abnormal* situation. Similarly, rules mined when there are no intrusions comprise a description of the *normal* situation while rules mined in the presence of an intrusion describe an *abnormal* situation.

2. FUZZY ASSOCIATION RULES FOR INTRUSION DETECTION

One approach to detecting intrusions in computer networks is to recognize deviations from normal patterns of behavior. Lee, Stolfo, and Mok (1999) reported the application of data mining techniques to this problem. They represent patterns as *association rules* that specify the correlation among various diagnostic features, or as *serial frequency episodes* that represent the re-occurrence characteristics of event sequences. A network intrusion can be detected by comparing a non-intrusion reference rule set with a test rule set, rather than by comparing individual rules. Luo (1999) modified this approach by introducing the mining of fuzzy association rules and fuzzy frequency episodes. The introduction of fuzzy logic can make network traffic patterns more abstract and flexible, and can overcome the "sharp boundary problem" identified by Kuok, Fu, and Wong (1998). In Luo's system, the fuzzy membership functions were determined by the expert building the system. Luo's prototype has demonstrated the effectiveness of this approach in alarming a network intrusion both online and offline. We show that the performance of the prototype can be improved by learning the membership functions from audit data rather than by empirically deriving them from human observation. Automatic derivation of membership functions from audit data will also allow different membership functions to be derived for different situations.

3. FUZZY LOGIC/GENETIC ALGORITHM SYSTEM

The GA system developed by Chen (1998) has been adapted in the current project to derive appropriate membership functions for the fuzzy intrusion detection system (Luo 1999). The original membership functions are first used to construct one chromosome. An initial population is then derived from the first chromosome by repeated application of the mutation operator (described in section 3.3). In each generation, the fitness of every new chromosome is first evaluated based on the performance of the intrusion detection system using the fuzzy membership functions represented by the chromosome. A specified percentage of the most fit chromosomes are retained for the next generation. Then parents are selected repeatedly from the current generation of chromosomes, and new chromosomes are generated from these parents by crossover and mutation. One generation ends when the number of chromosomes for the next generation has reached the

quota. This process is repeated for a pre-specified number of generations.

3.1. Data Structures

In this initial work, each fuzzy variable is defined by 3 fuzzy sets. We use the standard Z, Π, and S functions as the membership functions for these fuzzy sets (Orchard 1995). Each of these functions has two parameters as shown in Figure 1. A gene in this fuzzy logic/genetic algorithm is the set of 6 parameters that are used to define the standard membership functions of a fuzzy variable (Figure 1). In turn, a chromosome is a string of such genes where each gene represents a different fuzzy variable (Figure 2).

For our intrusion detection task, the fuzzy variables SN, FN, and RN are the number of SYN, FIN, and RST flags appearing in TCP packet headers during last 2 seconds. The fuzzy variable PN is the number of different destination ports during the last 2 seconds. These quantitative features of network traffic are thought to be diagnostic for network intrusions (Porras and Valdes 1998; Lee and Stolfo 1998).

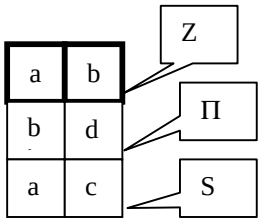

Fig. 1. Gene data structure

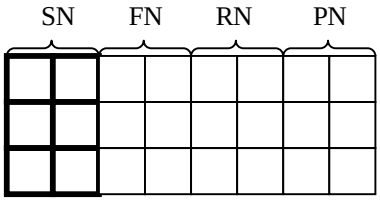

Fig. 2. Chromosome data structure

3.2. Fitness Functions

The fitness function for the genetic algorithm is based on the similarity of rule sets mined from different sets of audit data. Luo (1999) defines the similarity between two fuzzy association rules and between two fuzzy rule sets.

An association rule is of the form: $X \rightarrow Y$, c, s, where X and Y are two disjoint item sets ($X \cap Y = \varnothing$), s is the *support* for the association rule, and c is the c*onfidence* for the rule. The support for a rule is the percentage of transactions in which X and Y appear in the same transaction: $s = n'/n$ (n' is the number of transactions that contain both X and Y, n is the total number of transactions). The confidence in the rule is the percentage of the transactions containing X that also contain Y: $c = n'/n''$ (n'' is the number of transactions that contains X). Algorithms for finding association rules in transaction data use two parameters called the *minimum support threshold* and the *minimum confidence threshold* (Agrawal and Srikant 1994). Only rules with s and c values above the thresholds are mined from the data.

Given two association rules, $R1$: $X \rightarrow Y$, c, s, and $R2$: $X' \rightarrow Y'$, c', s', if $X=X'$ and $Y=Y'$, then the similarity between $R1$ and $R2$ is:

$$similarity(R1, R2) = \max\left(0, 1 - \max\left(\frac{|c-c'|}{c}, \frac{|s-s'|}{s}\right)\right)$$

otherwise, $similarity(R1, R2) = 0$.

The similarity between two rule sets $S1$ and $S2$ is:

$$similarity(S1, S2) = \frac{s}{|S1|} * \frac{s}{|S2|}$$

where $s = \sum_{\substack{\forall R1 \in S1 \\ \forall R2 \in S2}} similarity(R1, R2)$, and

$|S1|$ and $|S2|$ are the total number of rules in $S1$ and $S2$, respectively.

Three different sets of audit data were available for testing the fuzzy logic genetic algorithm: a normal data set with no intrusions and two "abnormal" data sets with different types of intrusions. The normal data set was partitioned into two sets. One partition of the normal audit data is called the reference data and one set is called normal data. This data is described in more detail in section 4. The following five fitness functions have been designed and tested:

$$F_1 = \frac{S_{rn}}{S_{ra1}} * \frac{S_{rn}}{S_{ra2}}$$

$$F_2 = \frac{S_{rn}}{S_{ra1}} + \frac{S_{rn}}{S_{ra2}}$$

$$F_3 = 2S_{rn} - S_{ra1} - S_{ra2}$$

$$F_4 = \frac{S_{rn}}{S_{ra1}}$$

$$F_5 = S_{rn} - S_{r1}$$

where S_{rn} is the similarity between a reference rule set and the "normal" rule set, S_{ra1} and S_{ra2} are respectively the similarity between the reference rule set and abnormal rule set 1 and between the reference rule set and abnormal rule set 2. In practice, a small constant has been added to all denominators in the above fitness functions to avoid division by zero.

Functions 1-3 will drive a co-evolution of the system's sensitivity to the two types of intrusions, because the similarities between the reference rule set

and the two anomaly rule sets must be minimized at the same time. Functions 4-5 are designed to test the versatility of our intrusion detection approach, i.e., a system trained with only one type of intrusion is used for detecting another type of intrusion.

3.3. Other GA Parameters

The selection strategy is a combination of fitness-proportionate "roulette wheel" selection, re-scaling, and elitism. The single-point crossover strategy is used with the crossover point determined randomly. The fixed crossover probability was 0.6 and the fixed mutation probability was 0.0333. When mutation occurs, all parameters within a gene are changed by a percentage of their current values. This percentage is chosen randomly from a preset range at run-time. The direction of change (positive or negative) is also determined randomly. A population size of 30 was used in all experiments.

4. EXPERIMENTAL RESULTS

The network intrusion detection system has been trained and tested using three sets of network traffic data downloaded from http://iris.cs.uml.edu:8080. These three data sets were collected by tcpdump. The first data set, *baseline*, was collected in normal situations (no network intrusion), the second data set, *abnormal1*, was collected when "IP spoofing" intrusions were simulated, and the third data set, *abnormal2*, was collected when "port scanning" intrusions were simulated. Four diagnostic flags (SN, FN, RN, PN) were extracted from the raw data sets. The data set *Baseline* was divided into two partitions, one of which will be called *reference* and the other *normal*. *Reference* was used to mine the reference association rule set that represents the normal situations, while *normal*, *abnormal1* and *abnormal2* were used to mine new association rule sets to be compared with the reference rule set.

The goal is to use the GA to derive membership functions that maximize *normal similarity*, and minimize both *abnormal1 similarity*, and *abnormal2 similarity*. *Normal similarity*, *abnormal1 similarity*, and *abnormal2 similarity* are short names for the similarity between the reference association rule set mined from *reference* and one of the three rule sets mined from *normal*, *abnormal1*, and *abnormal2*.

Figure 3 presents the results of an experiment comparing the similarity of the three rule sets and the reference set before and after the GA was used to optimize the membership functions. Fitness function *F4* was used in this experiment.

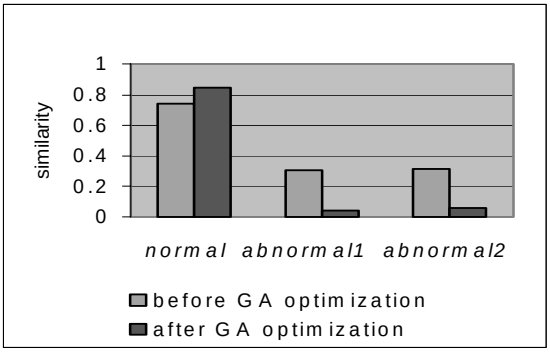

Fig. 3 Similarities between the reference rule set and the 3 test rule sets (*normal*, *anormal1*, *abnormal2*) before and after GA optimization using fitness function F4.

Table 1 shows the similarities before and after optimization with all five fitness functions. In all cases the similarities of the references rule set and normal rule set are increased while the similarities of the reference rule set and abnormal sets are decreased. The results using fitness functions F_4 and F_5 indicate the generality of our detection approach, since the system was trained in those experiments with only one type of intrusion, but it was able to recognize another type of intrusion. As a matter of fact, *abnormal1 similarity* and *abnormal2 similarity* were found close to each other in all experiments, regardless of the fitness function used or the parameter settings.

	Original	F1	F2	F3	F4	F5
Normal	0.74	0.81	0.81	0.84	0.85	0.84
Abnor-mal1	0.31	0.05	0.07	0.05	0.04	0.04
Abnor-mal2	0.32	0.00	0.00	0.03	0.06	0.04

Table 1. Similarities of rule sets to the reference rule set before optimization and using different fitness functions.

In addition to the effect of the fitness functions, the settings of *minimum support* and *minimum confidence* obviously influence the system performance. Figure 4 shows the effect of two different minimum support thresholds when abnormal set 1 is compared to the reference set for all fitness functions. The lower minimum support value results in a larger number of association rules with a greater degree of variation.

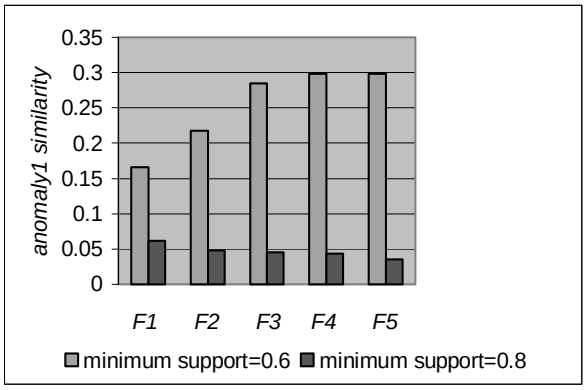

Fig. 4. Different *minimum support*s lead to different sets of *anormal1 similarities*.

CONCLUSIONS

A GA-optimized fuzzy logic system has been experimentally shown to have an improved performance in detecting network intrusion, i.e., maximizing the similarity between normal association rule sets while minimizing the similarity between a normal and an abnormal association rule set. Further experiments are required to determine the effectiveness of the resulting membership functions when used with different normal network situations and with other types of intrusions. Experiments will also be conducted to determine the effects of additional variations in the GA parameters and algorithm.

In addition to deriving membership functions for fuzzy association rules, our approach can also be used to optimize the membership functions for *fuzzy serial frequency episodes*, and to tune other association rule mining parameters such as *minimum support* and *minimum confidence*.

ACKNOWLEDGEMENT

A portion of this work was done as part of a class project in a Genetic Algorithms class taught by Dr. Gene Boggess.

REFERENCES

Agrawal, R. and R. Srikant. 1994. Fast algorithms for mining association rules. In *Proceedings of the 20th international conference on very large databases held in Santiago, Chile, Sept. 12-15, 1994*, 487-99. San Francisco, CA: Morgan Kaufmann.

Chen, J. 1998. Derivation of membership functions for fuzzy variables using genetic algorithms. M.S. Thesis, Mississippi State University.

Herrera, F., and J.L.Verdegay. 1997. *Genetic algorithms and soft computing*. http://decsai.ugr.es/~herrera/pub-int.html (Accessed 1 August 1999).

Kuok, C., A. Fu, and M. Wong. 1998. Mining fuzzy association rules in databases. SIGMOD Record 17(1): 41-6. (Downloaded from http://www.acm.org/sigs/sigmod/record/issues/9803 on 1 March 1999).

Lee, W., and S. Stolfo. 1998. Data mining approaches for intrusion detection. In *Proceedings of the 7th USENIX security symposium, 1998.* (Downloaded from http://www.cs.columbia.edu/~sal/recent-papers.html on 10 March 1999.)

Lee, W., S. Stolfo, and Mok. 1999. A data mining framework for building intrusion detection models. (Downloaded from http://www.cs.columbia.edu/~sal/recent-paper.html on 10 March 1999).

Luo, J. 1999. Integraing fuzzy logic with data mining methods for intrusion detection. M.S. Thesis, Mississippi State University.

Nawa, N. E., F. Takeshi, T. Hashiyama, and Y. Uchikawa. 1998. A study on the discovery of relevant fuzzy rules using pseudo-bacterial genetic algorithm. http://www.bioele.nuee.nagoya-u.ac.jp/~eiji/ papers/transIE99/ (Accessed 29 July 1999).

Okamura, M., H. Kikuchi, R. R. Yager, and S. Nakanishi. 1998. *Character diagnosis of fuzzy system by genetic algorithm and fuzzy inference.* http://www.ep.utokai.ac.jp/~masakazu/vietnam/vpaper.htm (Accessed 30 July 1999).

Orchard, R. 1995. *FuzzyCLIPS version 6.04 user's guide.* Knowledge System Laboratory, National Research Council Canada.

Porras, P., and A. Valdes. 1998. Live traffic analysis of TCP/IP gateways. In *Proceedings of the 1998 ISOC symposium on network and distributed systems security held in March, 1998.* (downloaded from http://www2.csl.sri.com/emerald/downloads.html on 1 March 1999.)

Tang, K. S., K. F. Man, Z. F. Liu, and S. Kwong. 1998. Minimal fuzzy memberships and rules using hierarchical genetic algorithms. *IEEE Transactions on Industrial Electronics* 45 (1): 162-69.