Using Genetic Algorithm Based Distance Metric Learning in Intrusion Detection Systems

Ezat soleiman, Abdelhamid fetanat

Abstract—previous works on intrusion detection systems proved that hybrid method solutions have more advantages than single method solutions. This paper will introduce the method of how to join genetic algorithm with distance metric learning in intrusion detection system. With the rapid expansion of Internet in recent years, computer systems are facing increased number of security threats. Despite numerous technological innovations for information assurance, it is still very difficult to protect computer systems. Therefore, unwanted intrusions take place when the actual software systems are running. Different soft computing based approaches have been proposed to detect computer network attacks.

Index Terms— genetic algorithm, intrusion detection system,IDS, distance metric learning.

1 Introduction

Nowadays computers and the Internet are used almost in every part of our lives. Since the personal computer was invented it has been growing faster and faster and it is now impossible to imagine companies, universities or even a little shop that does not keep all the data of their customers, purchases and inventory in an electronic database or computer.

With the possibility of connecting several computers and networks was born the necessity of protecting all this data and machines from attackers (hackers) that would like to get some confidential information to use for their own benefit or just destroy or modify valuable information.

There are several security measures available to protect the computer resources of a company or a home user, but even if all expert recommendations are followed, our systems will never be safe against possible successful attacks. It is very difficult to get an invulnerable system, probably impossible and one may need to spend a lot of money designing and developing it. In companies, a very isolated system could drastically reduce productivity and for a not very experienced home user it may become a "hating technology" disease. For all these reasons the user or the security department should know what their values are, if they need to be protected and how much it costs, doing Risk Analysis [1].

According to the Edward Amoroso [2] the intrusion is "sequence of related actions by malicious adversary that results in the occurrence of unauthorized security threats to a target computing or networking domain".

An intrusion is considered as a sequence because it propagates under a current period. Actions causing the intrusion must be related, since unrelated ones are not of interest. As an intruder always has an attention to make an intrusion, he must be considered as a malicious adversary. Assuming there is one defined security policy, unauthorized security threat, is its violation.

A good security policy and a good risk analysis with well-educated users will make the system more secure to intrusions. An intrusion in the system will try to compromise one of the three main aspects in computer security.

• Confidentiality: the intruder has access to confidential information.

• Integrity: information can be modified or altered by the attacker.

• Availability: the system gets blocked so it cannot be used normally.

These three aspects are illustrated in figure1.

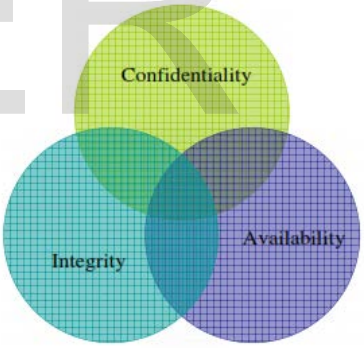

Fig. 1. The three main compromised aspects

2 Why use an IDS?

Intrusion detection allows protecting organization systems against threats that appear with increasing network connectivity and the interdependency of information systems [3].

IDSs have gained acceptance as a main part of the security infrastructure within an organization. There are several reasons for acquiring and using an IDS:

1- Avoid problems by dissuading hostile individuals

2- Detect attacks and other security violations not prevented by other protection measures.

3- Detect attack preambles

4- Record the organization risk

5- Provide useful information about the intrusions currently taking place

• Ph.D. Candidate, Dept of ICT Eng, Maleke Ashtar Univ. of Tech., Islamic Republic of Iran. Email: ezut_sol73@yahoo.com
• Assistant Professor, Maleke Ashtar Univ. of Tech, Islamic Republic of Iran. Email: abfetanat@gmail.com

3 Types of Computer Attacks

The standard attack classification that is widely used is by Kendall [3]. The attacks are grouped into five major categories.

3-1 Denial of service (DOS)

The example attacks include: Apache2arppoison, Back, Crashiis, dosnuke, Land, Mail-bomb, SYN Flood (Neptune), Ping of Death (POD), Process Table, selfping, Smurf, sshprocesstable, Syslogd, tcpreset, Teardrop, Udpstorm.

3-2 User to root (U2R)

U2R refers to a class of exploit in which the attacker breaks into the system as the normal user then eventually completely control the machine as the root user. The example attacks include anypw, casesen, Eject, Ffbconfig Fdformat, Loadmodule, ntfsdos, Perl, Ps, sechole, Xterm, yaga.

3-3 Remote to local (R2L)

R2L refers to the exploits that start from remote network-based access which intend to break into the machine and obtain the user account. Example attacks include: Dictionary Ftpwrite Guest, Httptunnel, Imap, Named, ncftp, netbus, netcat, Phf, ppmacro, Sendmail, sshtrojan, Xlock, Xsnoop

3-4 Probes

The example attacks include: insidesniffier, Ipsweep, ls domain,Mscan, NTinfoscan Nmap, queso, resetscan, Saint, Satan.

3-5 Data,

The example attacks include: Secret.

3-6 Trojan horses / worms – attacks:

That are aggressively replicating on other hosts

4 Genetic Algorithms

4-1 Introduction

GAs take their inspiration from biological evolution as proposed by Darwin. In biological evolution, individuals from species that adapt to their environment have a chance to survive and reproduce through natural selection. Species that survive usually develop new capabilities and capacities that can be inherited by offspring, if those prove to be worthwhile, and can be maintained through generations [4].

Table 1 shows a brief description of the correspondence between natural and artificial terminology.

TABLE1.
GA AND NATURAL TERMINOLOGY COMPARISON

Natural	Genetic Algorithm
Chromosome	String
Gene	Feature, character or detector
Allele	Feature value
Locus	String position
Genotype	Structure, or population
Phenotype	Parameter set, alternative solution, a decoded Structure

4-2 Initial Population

GA starts with a population of strings to be able to generate successive populations of strings afterwards. The initialization is usually done randomly [5]. Once a population is generated, all individual in that population has to be evaluated to distinguish between good and bad individuals.

4-3 Selection

The individuals that are chosen for mating (recombination) and how many offspring each individual produces are determined by the selection method.

4-4 Recombination (crossover)

The function of the crossover operator is to allow the advantageous qualities to be spread throughout the population in order that the population as a whole may benefit from this chance discovery.

4-5 Mutation

After recombination, every offspring undergoes mutation. Offspring variables are mutated by small perturbations (size of the mutation step), with low probability.

4-6 Reinsertion

After producing offspring, they must be inserted into the population. This is especially important, if less offspring are produced than the size of the original population [12].

5 Related Works

The early effort of using GAs for intrusion detection can be dated back to 1995, when Crosbie et al. [6] applied the multiple agent technology and GP to detect network anomalies. Each agent monitors one parameter of the network audit data and GP is used to find the set of agents that collectively determine anomalous network behaviors.

Bridges et al. [7] develop a method that integrates fuzzy data mining techniques and genetic algorithms to detect both network misuses and anomalies. In most of the existing GA based IDSs, the quantitative features of network audit data are either ignored or simply treated, though such features are often involved in intrusion detection.

Chittur et al. [8] 41 unique attributes were compiled from nine weeks of raw TCP dump data from a network. Five million separate connection records were created.

Li et al. [9] Applied a GA to network IDS. It considered both temporal and spatial information of network connections in encoding the network connection information into rules in the IDS. The final goal of the GA was to generate rules that matched only the anomalous connections. In this implementation, the network traffic used for GA is pre-classified data set that differentiates normal network connections from anomalous ones.

Lu et al. [10] used GP for detecting novel attacks on networks. The use of GP to detect unknown attacks was based on the hypothesis that the new rules would have better performance than the initial rules that were based on known attacks.

Xiao et al. [11] presented an approach that used information theory and GA to detect abnormal network behaviors. Based on the mutual information between network features and the types of network intrusions, a small number of network features are closely identified with network attacks. Then a linear structure rule is derived using the selected features and a GA.

6 Genetic Algorithm Based distance metric learning in intrusion detection systems

One of the fundamental questions of machine learning is how to compare examples. If an algorithm could perfectly determine whether two examples were semantically similar or dissimilar, most subsequent machine learning tasks would become trivial. For example, in classification settings, one would only require one labeled example per class and could then, during test-time, categorize all similar examples with the same class-label. An analogous reduction applies to regression if a continuous estimate of the degree of similarity were available. It is not surprising that many popular machine learning algorithms, such as Support Vector Machines, Gaussian Processes, kernel regression, k-means or k-nearest neighbors (kNN) fundamentally rely on a representation of the input data for which a reliable, although not perfect, measure of dissimilarity is known[12].

A common choice of dissimilarity measure is an uninformed norm, like the Euclidean distance. Here it is assumed that the features are represented in a Euclidean subspace in which similar inputs are close and dissimilar inputs are far away. Although the Euclidean distance is convenient and intuitive, it ignores the fact that the semantic meaning of “similarity” is inherently task and data-dependent. To illustrate this point, imagine two researchers who use the same data set of written documents for clustering. The first one is interested in clustering the articles by author, whereas the second wants to cluster by topic. Given the nature of their respective tasks, both should use very different metrics to measure document similarity, even if the underlying features are computed through similar means. Often, domain experts adjust the feature representations by hand — but clearly, this is not a robust approach. It is therefore desirable to learn the metric (or data representation) explicitly for each specific application [13].

For example, suppose we have five classes in figure2. If we find some transforms (or metrics) that separate these classes so that they have minimum overlap and maximum integration, we can then apply a simple classification method to classify these classes. Figures 2 to 6 show steps of metric learning.

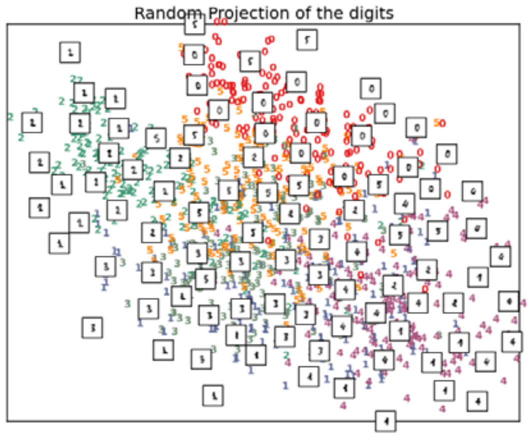

Fig. 2. First step of metric learning

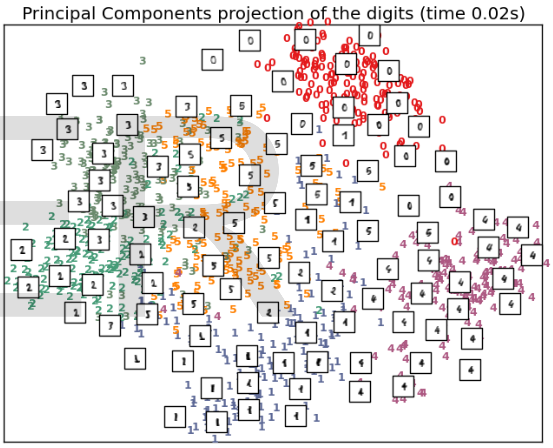

Fig. 3. Second step of metric learning

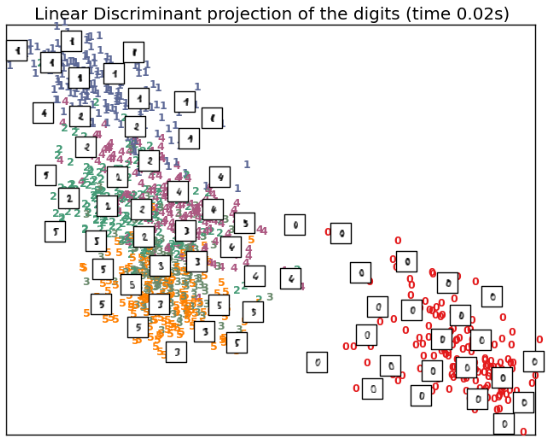

Fig. 4. Third step of metric learning

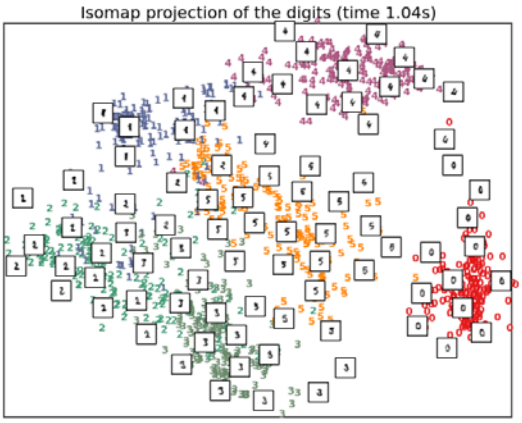

Fig. 5. Fourth step of metric learning

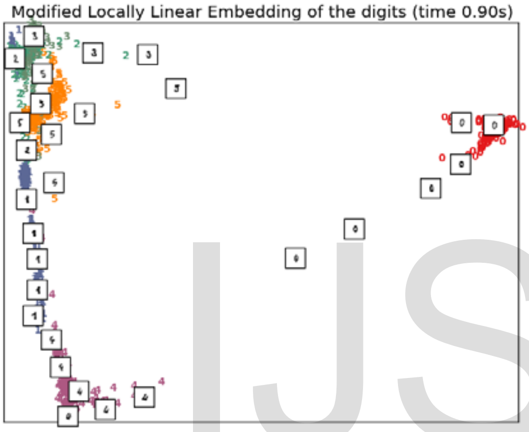

Fig. 6. Fifth step of metric learning

For learning an appropriate distance metric we should have a set of information about position of each data relative to each other in feature space. Before that, we should be familiar with must-link and cannot-link. Suppose that data in the feature space construct a graph so that nodes of the graph are the data. We can assume that each node can connect to every node in feature space. We have two types of vertex; inter class (cannot-link) and intra-class (must-link). Figure 7 shows the must-links and cannot-links. In clustering problems, both have equal importance because we have no information about labels and number of clusters. But in classifications, cannot-links have more importance than must-links.

Selection of all possible cannot-links and must-links causes over fitting during learning of distance metric. So, we should establish a tradeoff between generality and efficiency. We can use genetic algorithms to find an appropriate subset of vertices to learn distance metric.

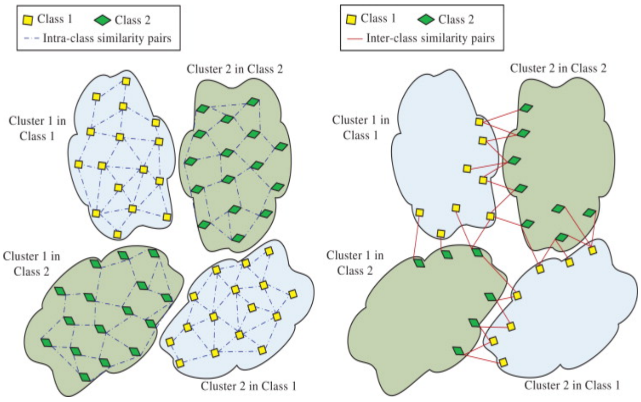

Fig. 7. Classification of must_link and cannot link

Suppose we want to classify the data of figure 8 using metric learning.

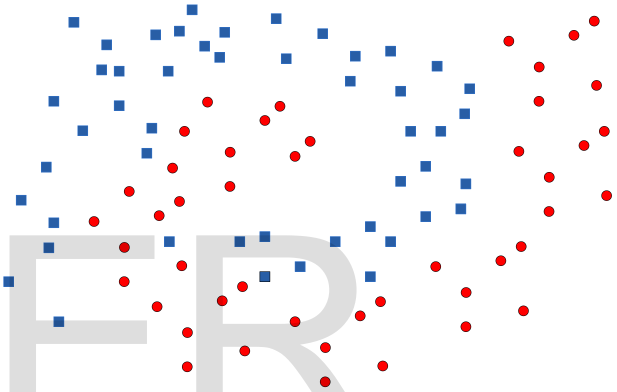

Fig. 8. Initial data

For classification using metric learning, we want a set of must-link and cannot-link among data set. Many questions appear in mind. How many links should we give to metric learner? We have n*(n-1)/2 links. How do we choose these links? What is the goodness criterion of a link?

We will answer to these questions one by one. A straightforward answer to the question that how many links we should give to metric learner is the more the better. If we have more knowledge about the must-links and cannot-links, we can accurately classify the data but we lose generality of classification. So, we should eliminate some unnecessary links.

For eliminating unnecessary links, we should have some criterion to measure the value of the links. Which is better link than others? A reasonable answer is that a must-link that minimizes intra class distance is better than other must-links. Must-links may have short length or long length. Which is better? In figure 9 we showed two approaches.

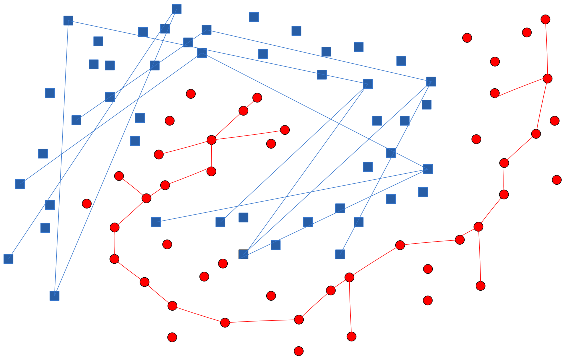

Fig. 9

According to figure 9, it is reasonable that we chose long length must-links as good links because they can effectively determine boundaries and skeletons of class simultaneously.

For cannot-links, we have the same approaches as the must-links. But here, the short length cannot-links have more information than long length cannot-links.

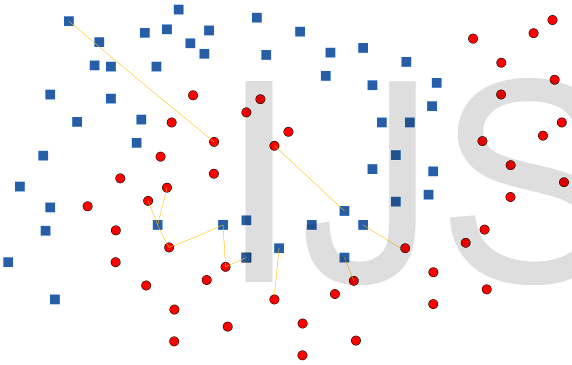

Fig. 10

In figure 10, we can see that the long length cannot-link will produce more error than short length CL (cannot-link) when we train metric learner.

Sometimes, good links may concentrate around a node. To prevent this situation, we should import a scattering term. Link goodness value decrease if degree of each connected nodes of link increase [14].

Now we define the problem and propose a genetic algorithm answer to find near optimal solution.

7 Definition of Problem

We have four classes' data and we have N samples of it with feature vector $(f_1,f_2,....,f_n)$. Find a subset S from all possible links subset ($2^{N(N-1)/2}-1$) that maximize total value of subset.

Total value of subset is:

$$fitness(S)=\alpha_1*\sum CL+\alpha_2*\sum ML-\alpha_3*M$$

The terms $\sum CL$ and $\sum ML$ are the total sum of all cannot-links value and must-links value. M is the number of all nodes that are included in the links set. w_1, w_2 and w_3 are the weights that user specifies. It was better that these weights are assigned automatically based on the accuracy of classifier. At first step we assumed that these are manual.

7-1 Suggested solution:

We can use GA to find near optimal solution. Now, we describe the GA model in detail.

7-2 Chromosome:

Each link is a chromosome that can be ML or CL.

7-3 Fitness function:

If link was a CL then its fitness value is:

$$fitness(CL)=\frac{1}{\sqrt{\frac{(f_1^{/}-f_1)^2}{I_1+1}+\frac{(f_2^{/}-f_2)^2}{I_2+1}+...+\frac{(f_n^{/}-f_n)^2}{I_n+1}}}+\frac{1}{D+D^{/}}$$

And if the link was ML, its fitness value is:

$$fitness(ML)=\sqrt{\frac{(f_1^{/}-f_1)^2}{I_1+1}+\frac{(f_2^{/}-f_2)^2}{I_2+1}+...+\frac{(f_n^{/}-f_n)^2}{I_n+1}}+\frac{1}{D+D^{/}}$$

Where $(f_1,f_2,....,f_n)$ and $(f_1^{/},f_2^{/},....,f_n^{/})$ are feature vectors of two nodes of the link. The term $\frac{(f_1^{/}-f_1)^2}{I_1+1}$ is the weighted distance in first dimension. The information of the first feature I_1 is the entropy of the feature. And finally $D+D^{/}$ is the sum of degrees of the nodes of link.

7-4 Population:

A population is the set of initially random links between data samples. If we have K classes, the number of links can be K*N because each node has a ML and (K-1) CL for the average; but this condition is not necessary.

7-5 Selection strategy:

Selection strategy is arbitrary but, we can use roulette wheel selection based on link fitness.

7-6 Crossover:

Suppose links in figure 11(a):

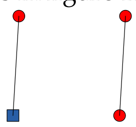

Fig. 11(a)

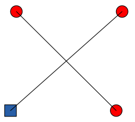

Fig. 11(b)

The crossover action will be figure 11(b).

7-8 Mutation:

Mutation with probability α is done by replacing a node of selected link with a random node.

References

[1] Dieter ollmann. Computer Security, Second Edition. Wiley, New Jersey, 2002.

[2] Edward G.Amoroso, "Intrusion Detection – An Introduction to Internet Surveillance, Correlation, Trace Back, Traps and Response", Intrusion.net Books, 1999.

[3] Ignacio Porres Ruiz, "an evaluation of current ids", master of science, university of Linköping, February, 2008.

[4] Dieter Gollmann . Computer Security, Second Edition. Wiley, New Jersey, 2002.

[5] Maiwald, Eric. "Network Security a Beginners Guide". Mc Graw Hill Professional, 2002.

[6] Brenton, Chris, and Hunt Cameron. "Mastering Network Security (2nd Edition)". Sybex, Incorporated, 2002.

[7] "Cisco Security Professional´s Guide to Secure Intrusion Detection Systems (IDS)". Syngress, 2003

[8] Stewart, J.m. "CISSP Professional: Certified Information Systems Security Professional Study Guide". Sybex, Incorporated, 2005

[9] Bace, R., Peter Mell. "Intrusion Detection Systems" 20 June 2007 http://csrc.nist.gov/publications/nistpubs/800-31.pdf

[3] Kendall, K., "A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems, in C.S. 1998, Massachusetts Institute of Technology: Boston

[4] Pedro A.Diaz-Gomez, "optimization of parameters for binary genetic algorithms", Doctor of philosophy, University Of Oklahoma, 2007.

[5] Kafi I.Hassan, "Adaptive algorithm for obtaining in-phase (I) and quadrature-phase (Q) pseudo-noise (PN) sequences in CDMA", Doctor of philosophy, The City University Of New York, 2005.

[6] M. Crosbie and E. Spafford, "Applying Genetic Programming to Intrusion Detection", Proceedings of the AAAI Fall Symposium, 1995

[7] S. M. Bridges and R. B. Vaughn, "Fuzzy Data Mining and Genetic Algorithms Applied to Intrusion Detection", Proceedings of 12th Annual Canadian Information Technology Security Symposium, pp. 109-122, 2000

[8] Chittur, A. "A Model Generation for an Intrusion Detection System Using Genetic Algorithms". http:// ww1.cs.columbia.edu/ids/publications/ gaids-thesis01.pdf. Accessed January, 2005

[9] Li, W. "A Genetic Algorithm Approach to Network Intrusion Detection". http://www.giac.org/practical/GSEC/Wei_Li_GSEC.pdf. Accessed January 2005

[10] W. Lu and I. Traore, "Detecting New Forms of Network Intrusion Using Genetic Programming", Computational Intelligence, vol. 20, pp. 3, Blackwell Publishing, Malden, pp. 475-494, 2004

[11] T. Xiao, G. Qu, S. Hariri, and M. Yousif, "An Efficient Network Intrusion Detection Method Based on Information Theory and Genetic Algorithm", Proceedings of the 24th IEEE International Performance Computing and Communications Conference (IPCCC '05), Phoenix, AZ, USA. 2005

[12] Weinberger, K. Q.; Blitzer J. C., Saul L. K. (2006). "Distance Metric Learning for Large Margin Nearest Neighbor Classification". Advances in Neural Information Processing Systems 18 (NIPS): 1473–1480.

[13] Weinberger, K. Q.; Saul L. K. (2009). "Distance Metric Learning for Large Margin Classification". Journal of Machine Learning Research 10: 207–244.

[14] Kumar, M.P.; Torr P.H.S., Zisserman A. (2007). "An invariant large margin nearest neighbour classifier". IEEE 11th International Conference on Computer Vision (ICCV), 2007: 1–8.