Asian Journal of Information Technology 5(7): 778-783, 2006

An Evolutionary Support Vector Machine for Intrusion Detection

[1]Sung-Hae Jun and [2]Kyung-Whan Oh
[1]Department of Bioinformatics and Statistics, Cheongju University, Chungbuk, Korea
[2]Department of Computer Science, Sogang University, Seoul, Korea

Abstract: Today most information interchanges are performed in the internet. In the environment, internet attacks continue to increase. So, the methods used as intrusion detective tools for protecting network systems against diverse attacks are very important. The skills of intrusion are getting more powerful continuously. However, the detection techniques have been hard to catch up with these attacks. Therefore, we need good tools for intrusion detection. Many researches for intrusion detection have been studied. Most of them had a difficulty in classifying intrusions from networks accesses in the case of new patterns of intrusions which were not experienced by predictive models. In this study, we propose an efficient method to settle the problem. Our model is constructed by combining evolutionary programming into support vector machine. This model is able to detect new attacks as well as experienced attacks. We verify an improved performance of our model using KDD Cup-99 task data designed by DARPA.

Key words: Evolutionary programming, support vector machine, intrusion detection

INTRODUCTION

In recent years, internet attacks which are denial of service (Dos), distributed denial of service (DDoS), unauthorized access from a remote machine (R2L), unauthorized access to local root privileges (U2R) and probing continue to increase. So, the technologies to protect systems from the intrusive attacks are needed. Continuously the techniques of intrusion have been cleverer than the skills of detection. Attack programs have been widespread by anonymous sources and thus individuals without related knowledge can do intrusion. This is a reason why the crimes of information securities have been increased recently. Anybody can do cracking, DoS and so forth, to do considerable damage to network systems using attack programs from the internet. The paradigm change of intrusion has been already begun. We have found this seriousness from the cases which are DDoS in Yahoo and Amazon web sites harmed by attacks. Most existing models for information securities were constructed by training only known intrusive examples[1,2]. But these have had a difficulty to detect new intrusive patterns which were unknown. So, novel researches for intrusion detection system have been studied by multiple disciplines[1-4]. Many works for intrusion detection have been published using machine learning algorithms such as neural networks, fuzzy set theory and Support Vector Machine (SVM)[5-8,3,9-12]. But these models had some problems[3,13-16]. So, we propose an Evolutionary Support Vector Machine (ESVM) for intrusion detection. Our ESVM is constructed by combining evolutionary programming into SVM. Proposed model can overcome the problems of existing models. The approach of ESVM is to make the detectable model for new attacks patterns as well as known attack patterns. Using ESVM, we are able to detect intrusive patterns which are known and unknown. In the experimental results, we verify improved performances of ESVM using KDD Cup-99 task data designed by the Defense Advanced Research Projects Agency (DARPA)[17,18].

RELATED WORKS

Intrusion detection: Attacks on the internet are commonplace. A network intrusion called an attack is a sequence of related actions by a malicious adversary whose goal is to violate some stated or implied policy regarding appropriate use of a network. Examples include stealing protected data, denying service to a user or group of users, or performing probing actions in an attempt to gain information in preparation for an attack. Growing reliance on the internet and worldwide connectivity has greatly increased the potential damage that can be inflicted by such attacks[19]. Intrusion can be defined as any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource. In the network systems, it refers to any unauthorized access, unauthorized attempt to access or damage, or malicious use of information resources[11]. Detection of anomaly patterns is computationally expensive because of the overhead of keeping track of

Corresponding Author: Sung-Hae Jun, Department of Bioinformatics and Statistics, Cheongju University, Chungbuk, Korea

and possible updating several system profile metrics, as it must be tailored system to system and sometimes even user to user, due to the fact behavior patterns and system usage vary greatly.

Evolutionary Programming: Evolutionary Programming (EP) is not Genetic Algorithm (GA) because EP emphasizes the development of behavioral models and not genetic models. In evolutionary process, a simulation of adaptive behavior is able to derive EP. We find optimal behaviors from a space of observable behaviors in evolutionary process. So, the fitness function consists of measures by behavioral error of individual. The crossover of GA is not implemented in EP. The mutation is only used in EP. EP is another member of the EC (Evolutionary Computing) family. EC is a special type of computing, which draws inspiration from the process of natural evolution. The fundamental of EC relates powerful natural evolution to a particular style of problem solving, that of trial and error[20]. Environment, individual and fitness of the basic EC are linked respectively problem, candidate solution and quality of the natural evolution to problem solving. EP is originally developed to simulated evolution as a learning process with the aim of generating artificial intelligence[21,22]. Intelligence is viewed as the capability of a system to adapt its behavior in order to meet some specified goals in a range of environments. EP is typically used for continuous parameter optimization. In this study, we combine EP into SVM to construct an efficient tool for intrusion detection.

Support vector machine: The learning is to construct a claim by observing data. The learning procedure contains from this till performing experiments and making conclusion. Statistical Learning Theory (SLT) developed by Vapnik[16]. SLT is perhaps the best currently available theory for finite sample statistical estimation and predictive learning. It has three types which are SVM, Support Vector Regression (SVR) and Support Vector Clustering (SVC). SVM, SVR and SVC are respectively classification, prediction and clustering tools[23,24,16]. All types of SLT are based on support vector. The approaches of support vector are projection instances into high dimensional spaces, learning linear separators with maximum margin and learning as optimizing upper bound on expected error. The classification problem of SLT can be restricted to consideration of the two-class problem. In this problem the goal is to separate the two classes by a function which is induced from available examples. The goal is to produce a classifier that will work well on unseen examples, that is, it generalizes well. Consider the problem of separating the set of training vectors belonging to two separate classes,

$$D = \{(x_1, y_1), \mathrm{L}\ , (x_l, y_l)\}, \quad x \in R^n, \ y \in \{-1, 1\} \tag{1}$$

with the hyperplane,

$$\langle w, x \rangle + b = 0 \tag{2}$$

The set of vectors is said to be optimally separated by the hyperplane if it is separated without error and the distance between the closest vector to the hyperplane is maximal. There is some redundancy in (2) and without loss of generality it is appropriate to consider a canonical hyperplane[16]. Where the parameters w, b are constrained by,

$$\min_i \left| \langle w, x_i \rangle + b \right| = 1 \tag{3}$$

This incisive constraint on the parameterization is preferable to alternatives in simplifying the formulation of the problem. In words it states that: the norm of the weight vector should be equal to the inverse of the distance, of the nearest point in the data set to the hyperplane. A separating hyperplane in canonical form must satisfy the following constraints,

$$y_i \left[\langle w, x_i \rangle + b \right] \geq 1, \quad i = 1, \mathrm{K}, 1 \tag{4}$$

The distance d (w,b; x) of a point x from the hyperplane (w,b) is,

$$d(w, b; x) = \frac{\left| \langle w, x_i \rangle + b \right|}{\|w\|} \tag{5}$$

The optimal hyperplane is given by maximizing the margin, subject to the constraints of (4). This approach of SVM is used for the prediction model, SVR and the clustering model, SVC[23,13,25,26]. So, SVM is a powerful learning machine. But, there are some problems in SVM.

AN INTRUSION DETECTION MODEL USING EVOLUTIONARY SUPPORT VECTOR MACHINE

To develop an efficient model for intrusion detection, we combine evolutionary programming into SVM. SVM is a learning machine to classify data by determining a set of support vectors, which are members of the set of training inputs that outline a hyper plane in feature space[16]. SVM is based on the idea of structural risk minimization, which minimizes the generalization error, that is, true error on unseen

examples. The primary advantage of the SVM is binary classification and regression that they provide to a classifier with a minimal VC-dimension[16], which implies low expected probability of generalization errors. SVM provides a generic mechanism to fit the surface of the hyper plane to the data through the use of a kernel function. The user may provide a function, such as a linear, polynomial, or sigmoid curve, to the SVM during the training process, which selects support vectors along the surface of this function. SVM offers a good performance because the optimization is convex[16]. But the optimal determinations of kernel parameters and regularization parameter are the problems of SVM. In general, these are selected by the arts of researchers. In this study, we overcome the problems using combining EP into SVM. That is, the parameters of SVM are determined by optimal search of EP.

In our research, all attacks are classified as +1 and a normal is classified as +1. In ESVM, our goal is to construct the hyper plane for optimal separating. To make this hyper plane, we need to maximize the quadratic form (6) subject to constraints (7).

$$W(\alpha) = \sum_{i=1}^{1} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{1} \alpha_i \alpha_j K(x_i, x_j) y_i y_j \qquad (6)$$

$$\sum_{i=1}^{1} y_i \alpha_i = 0, \quad 0 \le \alpha_i \le C, \quad i = 1, 2, \ldots 1 \qquad (7)$$

where, x and y are shown in (1). K( ) is a kernel function. The kernels for three common types of SVM are polynomial function, radial basis function(RBF) and sigmoid function[22]. In (6) and (7), the kernel parameters of K( ) and regularization parameter C are determined efficiently. So, we propose ESVM for solving the problem of optimal determination. EC including EP are search method for optimization problems, in which a mechanics of natural evolution principle is used to obtain the global optimal solution. They have been demonstrated considerable success in combination with other machine learning methods[27-29]. In the following, we show an algorithm for efficient selection of the parameters.

Step 1. (Initialize)
(1-1) Let t=0;
(1-2) Create an initial population;
$\{i_t = (w_{p\times1}, C)\} \in I_t$
$W_{p\times1}$: parameters of kernel function
C: regularization parameter
Step 2. (Repeat until stop conditions)
(2-1) Evaluate the fitness of as $f(i_t)$;
(2-2) Mutate to produce $o_t \in O_t$;
(2-3) Let t=t+1;

The stop conditions in the above consist of two cases of terminated requirements. Firstly, the process of ESVM algorithm is stopped when the total number of fitness evaluations researched a given limit. Secondly, for a given period of time, until the fitness improvement is remained under a threshold value, our algorithm has been stopped. In above mutation in step 2, we draw z_i from N(0,1) which is standard normal distribution. According to the comparison of fitness value, we are able to select i_{t+1} as the following.

Draw $z_i \sim N(0,1)$;
$y_t = i_t + z_i$ for all ; $i \in \{1, \ldots, n\}$
If $(f(i_t) < f(y_t))$ then $i_{t+1} = i_t$; else $i_{t+1} = y_t$;

In this study, we use lift value(LV) measure as fitness function. Lift is typically calculated by dividing the percentage of expected response predicted by the data mining model by the percentage of expected response predicted by a random selection. For example, suppose that 2% of the customers mailed a catalog without using the model would make a purchase. However, using the model to select catalog recipients, 10% would make a purchase. Then the lift is 10/2 or 5. Lift may also be used as a measure to compare different intrusion detection models. Since lift is computed using a data table with actual outcomes, lift compares how well a model performs with respect to this data on predicted outcomes. Lift indicates how well the model improved the predictions over a random selection given actual results. Lift allows a researcher to infer how a model will perform on new data. Generally the lift value is defined as the following[30,31].

$$\text{fitness}_{LV} = \frac{\text{response\_percentile}}{\text{BaseLine\_LiftValue}} \qquad (8)$$

In above equation, response_percentile is percentage of the number of correctly predicted attacks using constructed model and BaseLine_LiftValue is the base line lift value which is the predicted result by random selection without modeling. For example, a model has twice improved performance when its LV is 2. We summarize our algorithm in (Fig. 1).

In above process, in EP step, we efficiently determine kernel parameters and regularization parameter using EP. Using optimal parameters from EP step, an improved model for intrusion detection is constructed in SVM step. Next, we are able to classify attacks from diverse intrusions.

Our ESVM has a problem of computing time because EP search cost adds to SVM computing time.

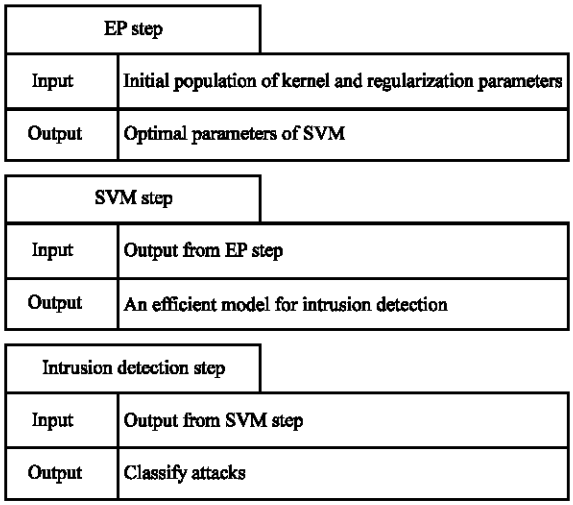

Fig. 1. ESVM Process

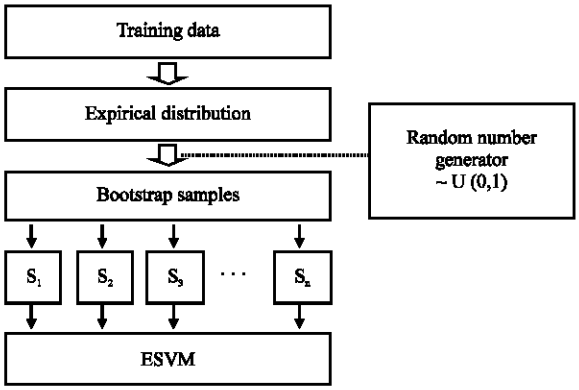

Fig. 2. Bootstrap for ESVM

To solve our computing burden, we consider bootstrap re-sampling[32].

In Fig. 2, we construct ESVM model by bootstrap. Using re-sampling by random number generated from uniform distribution between 0 and 1, small training samples are made. We make ESVM model using small samples instead of large training data. So, we are able to reduce the computing time of ESVM.

RESULTS

Our experimental data which were the network packet data from KDD Cup-99 to evaluate the accuracy of intrusion detection were acquired from the 1998 DARPA intrusion detection evaluation program[17]. The data were designed by DARPA for the program of intrusion detection evaluation[18]. They were based on an environment to acquire raw TCP/IP dump data for a local area network. The data consisted of diverse attacks. In Table 1, the attacks and non-attack classes and their data size are shown.

Table 1. Data summary

Classes	Data size
DoS	194,169 from 3,883,370
R2L	56 from 1,126
U2R	42 from 42
Probe	2,055 from 41,102
Non-attack	48,639 from 972,780

Table 2: Evaluated results in competitive models

Models	Accuracy (%)	Computing time (sec)
ESVM	96.4	4,593
ESVM with Bootstrap	95.6	1,736
SVM Polynomial	78.5	2,439
RBF	93.9	2,543
Sigmoid	80.4	2,675
Logistic	89.8	1,844
Gaussian mixture	90.4	1,934

In Table 1, Dos, R2L, U2R and Probe are denial of service, unauthorized access from a remote machine, unauthorized access to local root privileges and surveillance and other probing respectively. Non-attack is not included in any attacks. It is difficult to analyze the data because above data are very unbalanced. Dos attack patterns dominate other attacks and non-attack patterns. So, we consider a solution of the problem. To construct an effective model for intrusion detection, we use all U2R instances and sampled with probabilities proportional to size among DoS, R2L, Probing and Non-attack. Consequently we experiment using sampled data set in Table 1. Our experimental result of performance evaluation in the comparative models is shown in the Table 2. We compare ESVM with popular models which are logistic regression, Gaussian mixture model and SVMs with different kernels[13,16,33-37].

In the experiment, we compare the accuracy and computing time in the competitive models. The accuracy of ESVM is the best in the models in above results. But, it takes a lot of computing time. So, ESVM using total training data is not efficient. The performance of each SVM is severely varied according to its kernel type. The RBF kernel of SVM is the best among 3 kernels in our experiment. Generally the kernels of SVM are dependent on given training data set. To conclude, we verify an improved performance of ESVM with bootstrap re-sampling by accuracy and computing time.

CONCLUSION

In this study, we propose ESVM with bootstrap re-sampling model for an efficient model for intrusion detection. EP and SVM have been used in machine learning including pattern recognition, optimization and

so forth. In our ESVM, we combine EP into SVM to construct an intrusion detection model. We verify an improved performance of our model by the accuracy and computing time using KDD Cup-99 data set which is DARPA project. In our experimental results, we find that the performance of ESVM is improved. In future works, we will study on combining diverse evolutionary algorithms into support vector regression(SVR) and support vector clustering(SVC) to efficient regression and clustering models.

REFERENCES

1. Emran, S.M., M. Xu, N. Ye, Q. Chen and X. Li, 2001. Probabilistic techniques for intrusion detection based on computer audit data, IEEE Transactions on Systems, Man and Cybernetics, Part A, 31: 266-274.
2. Lee, W., S.J. Stolfo and K.W. Mok, 1999. A data mining framework for building intrusion detection models, Proceedings of the 1999 IEEE Symposium on Security and Privacy, pp: 120-132.
3. Jun, S.H., 2003. Hybrid statistical learning model for intrusion detection of networks, The KIPS Transaction: Part C, 10: 705-710.
4. http://www.ll.mit.edu
5. Cannady, J., 1998. Artificial neural networks for misuse detection. National Information Systems, Proceedings of Security Conference.
6. Debar, H., M. Becke and D. Siboni, 1992. A neural network component for an intrusion detection system. Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, pp: 240-250.
7. Debar, H. and B. Dorizzi, 1992. An application of a recurrent network to an intrusion detection system. Proceedings of the International Joint Conference on Neural Networks, pp: 78-483.
8. Ghosh, A.K., 1999. Learning program behavior profiles for intrusion detection, USENIX.
9. Kumar, S. and E.H. Spafford, 1994. An application of pattern matching in intrusion detection, technical report CSD-TR-94-013, Purdue University.
10. Luo, J. and S.M. Bridges, 2000. Mining fuzzy association rules and fuzzy frequency episodes for intrusion detection. Intl. J. Intelligent Systems, John Wiley and Sons, pp: 687-703.
11. Mukkamala, S., G. Janoski and A. Sung, 2003. Intrusion detection using neural networks and support vector machines. Proceedings of International Symposium on Applications and the Internet Technology, pp: 209-216.
12. Ryan, J., M.J. Lin and R. Miikkulainen, 1998. Intrusion Detection with Neural Networks, Advances in Neural Information Processing Systems 10, Cambridge, MA: MIT Press.
13. Haykin, S., 199. Neural Networks, Prentice Hall.
14. Jun, S.H., 2005. Web usage mining using support vector machine. Lecture Note in Computer Science, 3512: 349-356.
15. Mitchell, T.M., 1998. An introduction to Genetic Algorithms, MIT Press.
16. Vapnik, V.N., 1998. Statistical Learning Theory, John Wiley and Sons, Inc.
17. http://www.ecn.purdue.edu/KDDCUP
18. http://www.ll.mit.edu/IST/ideval/data
19. Haines, J.W., R.P. Lippmann, D.J. Fried, M.A. Zissman, E. Tran and S.B. Boswell, 2001. DARPA intrusion detection evaluation: Design and Procedures, Technical Report, Lincoln Laboratory, MIT.
20. Eiben, A.E. and J.E. Smith, 2003. Introduction to Evolutionary Computing, Springer.
21. Fogel, D.B., 1995. Evolutionary Computation, IEEE Press.
22. Fogel, L.J., A.J. Owens and M.J. Walsh, 1996. Artificial Intelligence through Simulated Evolution, Wiley, Chichester, UK.
23. Ben-Hur, A., D. Horn, H. Siegelmann and V.N. Vapnik, 2001. Support vector clustering, J. Machine Learning Res., 2: 125-137.
24. Vapnik, V.N., S. Golowich and A. Smola, 1997. Support vector method for function approximation, regression estimation and signal processing. Advances in Neural Information Processing Sys., 9: 281-287.
25. Smits, G.F., E.M. Jordaan, 2002. Improved SVM regression using mixtures of kernels. Proceesings of International Joint Conference on Neural Networks, 3: 2785-2790.
26. Smola, A.J., 1996. Regression estimation with support vector learning machines, Master's thesis, Technische University.
27. Liu, B., 2001. Fuzzy Kandom chance-constrained Programming, IEEE Transactions on Fuzzy Systems, 9: 713-720.
28. Quang, A.T., Q.L. Zhang and X. Li, 2002. Evolving support vector machine parameters. Proceedings of the First International Conference on Machine Learning and Cybernetics, pp: 548-551.
29. Yao, X., 1999. Evolving artificial neural networks. Proceedings of the IEEE, 87: 1423-1447.
30. SAS., 1998. Enterprise miner software. Applying Data Mining Techniques, SAS Institute Inc., (1998)
31. http://technet.microsoft.com

32. Davison, A.C., 1998. Bootstrap Methods and their Application. Cambridge University Press.
33. Breiman, L., J.H. Friedman, R.A. Olshen and C.J. 1984. Stone, Classification and Regression Trees, Wadsworth Inc.,
34. Cherkassky, V. and F. Mulier, 1998. Learning From Data Concepts, Theory and Methods, John Wiley and Sons.
35. Huet, S., A. Bouvier, M.A. Poursat and E. Jolivet, 2003. Statistical Tools for Nonlinear Regression, Springer Series in Statistics, Springer.
36. Mclachlan, G. and D. Peel, 2000. Finite Mixture Models, John Wiley and Sons, Inc.
37. Myers, R.H., 1990. Classical and Modern Regression with Applications, Duxbury Press.