Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 12, December 2014, pg.559 – 566

RESEARCH ARTICLE

Analysis on: Intrusions Detection Based On Support Vector Machine Optimized with Swarm Intelligence

Pankaj Shinde[1]
Thaksen Parvat[2]

[1]Department of Computer Network Engineering & Sinhagad Institute of Technology, Lonavala, savitribai phule pune university, India

[2]Department of Computer Engineering & Sinhagad Institute of Technology, Lonavala, savitribai phule pune university, India

[1] pankaj.mb.shinde@gmail.com, [2] pthaksen@yahoo.com

Abstract— This Intrusion detection system has become popular security intelligence component to provide security with capability of detecting attacks and patterns. Now day's globally use of IDS raising some lagging points like detecting false alert to be checked. Here new approach of support vector mechanism with swarm intelligence for selecting appropriate parameters to achieve high rate of attack detection and lower the false alarm than regular IDS. Recently, Support Vector Machines (SVM) has been employed to provide potential solutions for IDS. With its many variants for classification SVM is a state-of-the-art machine learning algorithm. However, the performance of SVM depends on selection of the appropriate parameters. In this paper we propose an IDS model based on Information Gain for feature selection combined with the SVM classifier. The parameters for SVM will be selected by a swarm intelligence algorithm (Particle Swarm Optimization or Artificial Bee Colony). We use the NSL KDD data set and show that our model can achieve higher detection rate and lower false alarm rate than regular SVM.

Keywords— Intrusion Detection, SVM, PSO, ABC and NSLKDD

I. INTRODUCTION

Security is major concern critical issue as the Internet applications and networking is growing. The current security technologies are going with on encryption, firewall and access control But still these technologies cannot assumable security. The system security can be enhanced by Intrusion detection. The ability of IDS to classify a large variety of intrusions in real time with accurate results is important. The process, patterns of user activities and log records are examined and the intrusions are located. IDSs are classified, based on their functionality, as misuse detectors and anomaly detectors. Misuse detection system uses well defined patterns of attack which are matched against user behaviour to detect intrusions. Nowadays, the use of networks and especially the Internet has become a big part of daily life. Various private as well as government organizations store valuable data over the network. Almost every activity has a corresponding term that begins with an e (e-banking, e-learning). According to rapid development and widespread use of network systems, diverse

intrusive approaches have grown extensively in the recent years. Multiple protection techniques have been used in order to manage the security network risks (encrypting sensitive data, access control, firewall policies). These methods do not suffice, as each of them have proven their inefficiency. Therefore, the use of intrusion detection systems as an additional defence mechanism is almost indispensable. An Intrusion Detection System (IDS) dynamically monitors the events taking place in a system, and decides whether these events are symptomatic of an attack (intrusion) or constitute a legitimate use of the system [11]. Based on the adopted data analysis approach, an IDS may belong to one of the two main groups : misuse detection(or signature based detection) or anomaly detection [17]. The first approach is most widely used and it detects only known attacks that have their signature included in the database. The anomaly detection approach creates a normal behaviour profile and detects intrusions based on significant deviations from the normal profile. Many challenges need to be considered when building an intrusion detection model, such as obtaining a high attack Detection Rate without generating many false alarms (low False Alarm Rate). Since the appearance of IDS [12], multiple techniques have been proposed in order to improve the performances of these systems. Recently, several machine learning techniques have been applied. Kruegel and Toth [15] proposed an approach based on decision trees for classifying the detection rules. Xiao et. al. [10] present a hybrid model based on information theory and genetic algorithm to detect network attacks. Their approach considers only discrete features. Chen, Abraham and Yang [9] proposed a flexible neural tree (FNT) model for intrusion detection based on neural tree for attribute selection and particle swarm optimization(PSO) for parameter optimization. In 2012 Pu, Xiao and Dong [3] use an improved ant colony algorithm in order to determine the best parameters for the SVM classifier. Support Vector Machines (SVM) is a novel machine learning approach that has become a popular research method in intrusion detection [8][3]. The main reasons are its good generalization performance, absence of local minimal and fast execution time. This paper proposes an anomaly network intrusion detection approach using Information Gain for feature selection and Support Vector Machine optimized with Swarm Intelligence for classification. The rest of this paper is organized as follows: Section II presents a background of the SVM algorithm and parameters selection based on PSO and ABC. Section III describes the proposed model, including the feature selection approach. Section IV shows the implementation results and analysis. Finally, the conclusion is presented in Section V.

II. SUPPORT VECTOR MACHINE

Support Vector Machine (SVM or Support Vector Network)[8] is a linear machine learning method based on structural risk minimization of statistical learning theory (STL). SVM shows good generalization skill, without the need of any priori knowledge. Furthermore, it does not suffer from the local minimum and it can handle noisy datasets. These attributes correspond with the requirements needed for implementing an efficient IDS.

A. Overview

The main objective of SVM is to find the optimum hyperplane to separate the two classes. For this, it only needs a small amount of support vector quantities in order to define this hyperplane. Given a training dataset (x 1, y 1), (x 2, y 2), ..., (xN, yN), where xi ∈ Rn is the n dimensional characteristic vector, y i ∈ {− 1, +1} is the class label and N denotes the total number of records from the training dataset. The hyperplane is defined by (w, b), where w is the weight vector and b is the bias. A new object x, can be classified with the following function g(x)=sign(wT·x+b)=sign(f(x)). Linear SVM is solved by formulating the quadratic optimization problem as: min w,b(wTw2), s. t. yi(wTxi + b) ≥ 1 (i = 1, ..., N) (1) The dataset is not always linearly separable. In these cases we can introduce a slack variable ξi ≥ 0 for each xi (i = 1, ..., N). Furthermore, we can map the dataset into a higher-dimension feature space and try to find the hyperplane that linearly separates the mapped vectors. In other words xi will be replaced with K(xi) where K provides the higher dimensional mapping (K is also called the kernel function). Usually, there are three main types of kernel functions: polynomial kernel function, radial-basis kernel function (RBF) and sigmoid function. In this paper we will use SVM constructed by radial basis function (RBF), that is a universal kernel function and has fewer controllable parameter. This means the kernel function will be K(xi, x) = exp(− 2σ1 2 k xi − xk 2).

B. Impacts Of The Parameters

For the SVM based on RBF as the kernel function, the parameters include adjusting C and σ.

• C - is a regularization parameter that controls the "flexibility" of the hyper plane. Finite C allows misclassifying some points and changes the problem of perfectly separable data to finding a "soft-margin" classifier. Lower C allows softer constraints and corresponds to a larger margin. On the other hand, larger C forces the creation of a more accurate model, with a narrow margin.

• σ - is the kernel parameter. This is a tunable parameter that controls the correlation among support vectors. In other words, selecting an improper value for σ may cause over fitting. The choice of SVM parameters has generated many debates and many solutions have been proposed, such as enumeration, the three-step search strategy, max-min nuclear parameter selection or intelligent optimization. During the recent years, intelligent optimization has been applied for training SVM or other neural networks algorithms and has shown great results involving PSO algorithm [8][18][10], genetic algorithm [7], ABC algorithm [11] etc.. In this paper we propose to apply Particle Swarm Optimization (PSO) and Artificial Bee Colony (ABC) to optimize parameters C and σ, two popular algorithms used for optimization problems. Both methods are swarm intelligence algorithms based on a population of individuals. These individuals are self-organized and even though, they are simple and there is no centralized control structure, the swarm reflects an "intelligent" global behavior. One key factor when implementing these algorithms is choosing the right fitness function.

C. Combining With Particle Swarm Optimization

Particle Swarm Optimization (PSO) is an evolutionary computation technique developed by Kennedy and Eberhart in 1995 [23]. The algorithm was inspired by the coordinate movement dynamics of groups of animals, such as bird flocking. This method optimizes a problem by trying to improve a candidate solution (the current location) with regard to a given measurement of quality(the fitness function). In our case it will search for the best parameters: C and σ based on the accuracy of the SVM algorithm. PSO performs searches using a population (or swarm) of agents (called particles). Each particle i has a current position loci = (loci,1, loci,2, ..., loci,d) t and a current flying velocity

veli = (veli,1, veli,2, ..., veli,d) t, where d is the problem dimension (d is two in our case). To discover the optimal solution, each particle moves in the direction of its previous best position (p_best) and its best global position (g_best), according to the following equations:

vel i,j t+1= w · vel i,j t+1 + c1 · r1 · (p_besti,j − loci,j t)+ c2 · r2 · (g_besti,j − loci,j t) (2)
loci,jt+1=loci,jt+veli,jt+1 (3)
In the above formula w is the inertia constant weight that isused to balance the global exploration and local exploration, c1 and c2 are personal and social learning factors (or acceleration constants), r1 and r2 are random numbers in the range [0, 1]. Furthermore, the value of the flying velocity (veli,j) is limited to the range [−vmax, vmax] and the value of the current position (loci,j) is limited to the range [−lmax, lmax]. A pseudocode of the Standard PSO is given in Algorithm 1. For the fitness function we used the accuracy obtained after training SVM with the user parameters C as Ploc,0 and σ as P loc,1 (Ploc is a two dimensional variable). PSO has three tunable parameters: the number of particles (SN), the maximum number of iterations (MAX_IT) and the maximum value of the fitness function (MAX_VAL).

Algorithm 1

Standard PSO
for i ← 1 SN do
Pvel ← RandomV el
Ploc ← RandomLoc
Pp

_best ← Ploc
Pg
_best ← Compare (F it (Pp_best), F it (Pg_best))
end for
while t < MAX_IT AND val < MAX_V AL do
for i ← 1 SN do
Pvel t +1 ← UpdateVel (Pvelt, Pp_best, Pg_best)
Ploct +1 ← UpdateLoc (Ploct, Pvelt +1)
Pp
_best ← Compare(F it(Pp_best), F it(Ploct +1))
Pg
_best ← Compare(F it(Pp_best), F it(Pg_best))
end for
t ← t + 1
val ← F it(Pg_best)
end while

D. Combining With Artificial Bee Colony

The Artificial Bee Colony (ABC)[1][16][6] algorithm is a swarm based algorithm introduced by Karaboga in 2005 [24] and it was inspired by the intelligent foraging behaviour of honey bees. The algorithm searches for an optimal solution (best food source) using a population (called swarm) of individuals (bees). In ABC, the swarm contains three groups of bees: employed bees that have a food source, scout bees that choose a food source after watching the dance of the employed bees and scout bees randomly searching for foodsource[15]. The solution for the optimization problem is given by the position of the food source. The nectar amount of the food source corresponds to the quality (fitness) of that solution [16]. First, the algorithm initializes the food sources (xi where i = 1, ..SN, SN is the population size) and sets the control parameters. Each food source xi has a d dimensional variable. This multidimensional variable(indicating the position of the food source) must be optimized so as to maximize the objective function (denoted as fi in equation (6)). In our case the objective function is the accuracy of the SVM classifier obtained with the food source solution as input parameters(C and σ). The position of food source xi is initialized as: $x_{i,j} = l_j + r \cdot (u_j - l_j)$ (4) where l j is the lower bound of xi,j and uj is the upper limit of x i,j. The employed bee modifies its position (solution) in her memory, depending on the local information. To determine the new food source, the employed bee uses the following formula: $x_{i,j} = x_{i,j} + r_{i,j} * (x_{i,j} - x_{k,j})$ (5) where xk is a randomly selected food source, j is a randomly selected index, r i,j is a random number within the range [−1, 1]. If the nectar amount (fitness value form equation (6)) of the new solution is better than that of the previous one, the bee keeps the new solution and forgets the old one. fiti = 11 + fi (6) After the employed bees share their information about their food source, the onlooker bees choose their food source. To do this, the onlooker bee will use the expression below: pi = fiti(xi) SN P i =1 fiti(xi) (7)

Finally, if employed bees can no longer improve their solution (it has performed all the predefined number of trials) then their solution is abandoned and they become scout bees. Scout bees will randomly select a new food source, based on equation (4). A pseudocode of ABC is shown in Algorithm 2. ABC has three tunable parameters: the number of food sources (SN from Algorithm 2), the maximum number of trails for improving a food source and the maximum number of loops (MaxCycle from Algorithm 2). Furthermore, in order to improve the execution time we add a maximum value for the objective function (Max_f)

Algorithm 2 ABC

Initialization P hase
xi ← Random cf (4)

fiti ← CalcF it cf (6)
repeat
for all i ← EmpBees do
xi ∗ ← newSol(r, xi, xk) cf (5)
fiti(xi) ← CalcF it cf (6)
Apply GREEDY to select best solution
end for
Compute pi cf (7) for solutions xi
for all i ← OnlookerBees do
select xi depending on pi
xi
← newSol(r, xi ∗ , xk) cf (5)
fiti(xi 00) ← CalcF it cf (6)
Apply GREEDY to select best solution
end for
if Abandoned solutions for ScoutBees then
new solution cf (4)
end if
Memorize best solution
cycle ← cycle + 1
until cycle = MaxCycle OR f(best sol) ≥ Max_f

III.PROPOSED MODEL

Our proposed IDS model has three different phases. First the data set is reprocessed by transforming the symbolic valued attributes to numeric and applying the discretion algorithm. Then, IG is used for feature selection and SVM is used for classification. For SVM, the parameters are selected by the swarm intelligence algorithm (ABC or PSO).

A. Data Set And Pre-processing

For the evaluation we will use the NSL-KDD data set [12]. This data set is an advanced version of KDD-Cup that does not suffer from issues such as redundancy and complexity level of data. Each NSL-KDD connection vector contains 41 features and is marked as as either normal or an attack. These features can be classified into three groups: connection based (9 features), content based (13 features) and time based (19 features). The attacks fall into four categories : Denial of service (DoS), Remote-to-Local(R2L), User-to-Root(U2R) and Probing. From this data set we will randomly select 9,566 records and divide them in order to create one file for training (a balanced data set) and one file for testing. We map the symbolic valued attributes (protocol_type, service, flag, class) to numeric.

B. Feature Selection

The technological evolution has brought many changes. With the huge data volume and the increasing speed of the network traffic, IDS must offer a viable solution to detect intrusions. The feature selection process can reduce the model complexity by removing irrelevant features [4]. In paper [9] feature selection methods are divided into three categories: filter, wrapper and hybrid (which is a combination between filter and wrapper). In this paper we used Information Gain (IG) for feature selection. This method is simple and belongs to the filter category. The IG of a given attribute A with respect to the class attribute C, denoted as I(C| A), is the reduction in uncertainty about the value of C when we know the value of A [2]. Let C and A be discreet variables that take the values C = (c 1 , ..., ck) and A = (a 1 , .., an). H(C) is the entropy, that measures the uncertainty about the

value of C. H(C| A) is the conditional entropy of C given A, that measures the uncertainty about the value of C after observing values of A. Thus

$I(C| A) = H(C) - H(C| A)$

$H(C) = -k \sum_{i=1} P(ci)\log 2(P(ci))$

$H(C| A) = - n \sum_{j=1} P(aj)\ k$

$\sum_{i=1} P(ci| aj)\log 2(P(ci| aj))$ (8)

where, P(ci| aj) is the posterior probabilities of C given the values of A. In order to apply the IG method, the numeric attributes are first discretized using the method of Fayyad and Irani [10].For this we used the InfoGainAttributeEval with Ranker as the search method from Weka. From the ranked list the top 26 features were selected .

IV. EVALUATION RESULTS

The experiments are made on a 1.80GHz Core (TM) 2 CPU personal computer with 2GB memory under Ubuntu 10.04.4. We implemented in Java the two algorithms (PSO and ABC) used for optimizing the SVM classifier. For the PSO algorithm we used 20 particles, c1 and c2 were set to 2.3 and 1.8, the inertia weight w was reduced from 0.9 to 0.5 and the problem dimension was set to 2. For parameter C we set the range between 1 and 108 and for σ we set the range between 0.001 and 25. The fitness function is the accuracy of the SVM classifier : Accuracy = number of correctly classified total number of examples (9) For the ABC method we used 20 bees (employed and onlooker bees), the limit was set to 40 and the problem dimension was set to 2. The lower and the upper bound for the two parameters (C and σ) are the same as in the PSO case. Weka version 3.6.10 [13] was used for the SVM classifier and feature selection.

A. Performance Measures

In machine learning and data mining algorithms, many different measures are used to evaluate the classification models. We use three performance measures: attack detection rate, false alarm rate and accuracy.

• Attack Detection Rate (ADR) - is the ratio between total numbers of attacks detected by the system to the total number of attacks present in the data set. ADR will show if our proposed model is capable of detecting attacks (thus raising alarms).

• False Alarm Rate (FAR) - is the ratio between total number of misclassified instances to the total number of normal instances. The FAR value will show if our model generates many false alarms. For IDS, this value should be as low as possible, otherwise too many false alarms may confuse the administrator.

• Accuracy - is the ratio between total number of correctly classified instances to the total number of samples from the data set. The accuracy will show if our model is capable of raising proper alarms, when it detects attacks and not generating false alarms when the network traffic is normal. These measures are calculated by using the confusion matrix.Where :

• TP (true positives) the number of attack records that are correctly classified, thus properly raising alarms.

• FP (false positive) the number of normal records that are incorrectly classified, thus generating false alarms.

•TN (true negatives) the number of normal records that are correctly classified as normal.
• FN (false negative) the number of attack records that are incorrectly classified as normal.

B. Results and Analysis

We use Weka to implement the SVM classifier and we compare our model with the standard SVM (with default options). Results show that our proposed model (IG with SVM and Swarm Intelligence algorithm) has the highest detection rate and the lowest false alarm rate. Nonetheless, the IG-SVM model offers poorer results than SVM, but its training process takes less time to execute. Our test revealed that the best performances are obtained when ABC algorithm is applied for optimizing the SVM classifier (best parameters obtained by ABC are C = 9.341707 and σ = 9.764002). The parameter selection process is faster when using PSO; but, the classifier obtained form SVM with PSO has a lower detection rate and a higher false alarm rate. On the other hand, the optimization process is executed only once, during the training of the NIDS. Thus, we can say that ABC is a more suitable solution.

IV. CONCLUSIONS

In this paper we come up with survey on proposed an IDS model that combines IG feature selection with the SVM classifier. For SVM parameters C and σ are elected by a swarm intelligence algorithm (PSO or ABC). The NSL-KDD network intrusion benchmark was used for conducting several tests in order to evaluate our proposed model. We showed that Swarm Intelligence algorithms can be used for optimizing the parameters of the SVM classifier, thus obtaining an improved detection system, more accurate than the standard SVM(with default parameter values). Tests showed that our model can obtain better results in terms of attack detection rate, false alarm rate and accuracy. Moreover, the IG-ABC-SVM approach obtained the highest detection rate (98.53%) and the lowest false alarm rate (0.0374). The future work will focus on applying swarm intelligence data mining algorithms for the feature selection process, hoping to obtain better results. Also, the experiments have been conducted using only two swarm intelligence algorithms. In the future, we are looking forward to using other types of swarm intelligence algorithms.

REFERENCES

[1] S. M. Metev and V. P. Veiko, *Laser Assisted Microtechnology*, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.

[2] A. Chatterjee, S. P. Ghoshal and V. Mukherjee , "Artificial Bee Colony Algorithm for Transient Performance Augmentation of Grid Connected Distributed Generation" in Swarm, Evolutionary, and Memetic Computing, Springer Berlin Heidelberg, Vol. 6466, 2010, pp. 559–566.

[3] A. Elngar, D. El A. Mohamed and F. M. Ghaleb, "A Real-Time Anomaly Network Intrusion Detection System with High Accuracy ", Information Sciences Letters, Vol. 2, No. 2, pp.49–56,May 2013.

[4] J. Pu, Y. Li, L. Xiao and X. Dong, "A Detection Method of Network Intrusion Based on SVM and Ant Colony Algorithm " in Proc. National Conference on Information Technology and Computer Science(CITCS 2012), Lanzhou, China, 2012, pp.153–156.

[5] H.F Eid, A. T. Azar and A. E. Hassanien, "Improved Real-Time Discretize Network Intrusion Detection Model", in Proc. Seventh International Conference on Bio-Inspired Computing: Theories and Application (BIC-TA 2012) Advances in Intelligent Systems and Computing, Gwalior, India, 2013, pp.99–109

[6] R. A. Sadek, M. S. Soliman and H. S. Elsayed, "Effective Anomaly Intrusion Detection System based on Neural Network with Indicator Variable and Rough set Reduction", International Journal of Computer Science Issues (IJCSI), Vol. 10, p227-233, Nov. 2013.

[7] D. Wu and J. Zheng, "A Dynamic Multistage Hybrid Swarm Intelligence Optimization Algorithm for Function Optimization", Discrete Dynamics in Nature and Society, vol. 2012, Article ID 578064, 22 pages, 2012.

[8] H. G. Jung, P. J. Yoon and J. Kim, "Genetic algorithm-based optimization of SVM-based pedestrian classifier", In The 22nd international technical conference on circuits/systems, computers and communications(ITC-CSCC2007), Busan, Korea, July 2007, pp. 783–784.

[9] J. Wang ,X. Hong and R. Ren, T. Li, "A real-time intrusion detection system based on PSO-SVM", in Proc. of the International Workshop on Information Security and Application 2009 (IWISA 2009), Qingdao, China, 2009,pp. 319–321

[10] Y. H. Chen, A. Abraham and B. Yang, "Hybrid flexible neural-treebased intrusion detection systems", International Journal of Intelligent Systems, Vol. 22(4), pp.337–352, April 2007.

[11] T. Xiao, G. Qu, S. Hariri, and M. Yousif, "An Efficient Network Intrusion Detection Method Based on Information Theory and Genetic Algorithm ", in Proc. of the 24th IEEE International Performance Computing and Communications Conference (IPCCC 2005), Phoenix, AZ, USA, 2005,pp.11–17.

[12] X. R. Yang, J. Y. Shen and R. Wang, "Artificial immune theory based network intrusion detection system and the algorithms design", in Proc. of 2002 International Conference on Machine Learning and Cybernetics, Beijing, China, 2002, pp.73–77.

[13] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set", In Proc. of the 2009 IEEE symposium on computational Intelligence in security and defense application (CISDA), Ottawa, ON, Canada, 2009, pp.1–6.

[14] H. Witten and E. Frank. Data Mining: Practical Machine Learning Tools and Techniques. 2nd edition, San Francisco: Morgan Kaufmann, 2005.

[15] M. A. Hall and G. Holmes. "Benchmarking Attribute Selection Techniques for Discrete Class Data Mining ", IEEE Transactions on knowledge and data engineering, vol. 15, no. 6, 2003, pp.1437–1447.

[16] D. Karaboga(2010), "Artificial bee colony algorithm", Scholarpedia [On-line], Vol. 5(3), pp. 6915. Available: http://www.scholarpedia.org/article/Artificial_bee_colony_algorithm [Dec. 2013].

[17] D. Karaboga and C. Ozturk, "A novel clustering approach: Artificial Bee Colony (ABC) algorithm", Applied Soft Computing, Elsevier Science Publishers B. V. Amsterdam, The Netherlands, Vol. 11(1), pp. 652–657, Jan. 2011.

[18] C. Koliasa,G. Kambourakisa and M. Maragoudakisa, "Swarm intelligence in intrusion detection: A survey", Computers and Security, Vol. 30(8), pp. 625–642, Nov. 2011.

[19] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition", Data Mining and Knowledge Discovery, vol. 2(2), pp.121–167, June 1998.

