Intrusion Detection Using a New Particle Swarm Method and Support Vector Machines

Essam Al Daoud

***Abstract*—**Intrusion detection is a mechanism used to protect a system and analyse and predict the behaviours of system users. An ideal intrusion detection system is hard to achieve due to nonlinearity, and irrelevant or redundant features. This study introduces a new anomaly-based intrusion detection model. The suggested model is based on particle swarm optimisation and nonlinear, multi-class and multi-kernel support vector machines. Particle swarm optimisation is used for feature selection by applying a new formula to update the position and the velocity of a particle; the support vector machine is used as a classifier. The proposed model is tested and compared with the other methods using the KDD CUP 1999 dataset. The results indicate that this new method achieves better accuracy rates than previous methods.

***Keywords*—**Feature selection, Intrusion detection, Support vector machine, Particle swarm.

I. Introduction

An intrusion is an action that attempts to compromise the availability, confidentiality or integrity of a resource. An attacker can gain illegal access to a system by exploiting bugs in a trusted program, a system configuration error, or by fooling an authorised user into providing information that can be used to control the system. Intrusion detection refers to a broad range of approaches to detect malicious attacks on computers and networks. Intrusion detection systems (IDS) are required as an additional 'wall', together with other prevention techniques, to protect systems. Intrusion detection models can be categorised into two main types: misuse-based and anomaly-based [1]. A misuse-based IDS is also known as signature-based or pattern-based, detecting known attacks based on information stored in a database. This type of intrusion detection, although efficient in detecting existing intrusions, is fooled by any small modification to the original model. Anomaly-based models can be used to detect both known and unknown intrusions, detecting deviations from normal connections. The main challenges in current anomaly intrusion detection systems are their low detection rates, which mean that they can potentially miss detecting serious attacks and the high 'false alarm' rates, which mean that a normal connection may be falsely classified as an attack. Therefore, many machine-learning methods have been used, such as artificial neural networks, Naïve Bays, random fields and multi-class support vector machines (SVM) [2, 3].

In general, attacks can be divided into four categories [4]:

- **Denial of Service (DoS):** Attacker tries to prevent legitimate users from using a service, computer or resource. Examples include SYN Flood, Ping of Death, Back, Smurf, Land, Apache2 and Teardrop.
- **Remote to User (R2L):** Attacker tries to gain access to the victim machine. Examples are Sendmail, Dictionary, Named, Guest, Imap, Ftp_write.
- **User to Root (U2R):** Attacker has local access to the victim machine and tries to gain super user privileges. Examples are Perl, Xterm, Loadmodule, Eject, Fdformat.
- **Probing (Probe):** Attacker tries to gain information about the target host. Examples are Saint, Nmap, Mscan, Satan, Ipsweep.

This study is based on the combination of a new particle swarm optimisation method and nonlinear, multi-class and multi-kernel support vector machines.

II. Particle Swarm Optimization

PSO is a relatively recent heuristic optimisation method; its mechanics are inspired by the natural flocking and swarming behaviour of birds and insects. A set of randomly generated solutions (initial swarm) is used to explore the space, and each particle (a bird, a fish or an insect) makes use of its individual memory, as well as the knowledge gained by the swarm as a whole, to find the best solution (a rich source of food and to avoid predators).

Let the position of the particle *i* at the step *t* be the vector x_i^t, its velocity be the vector v_i^t and the fitness function or the quality function be $f: X \to R$, where $X \subset R^d$. The best previous position of the particle i is denoted by p_i and the associated fitness value by $pbest_i = f(p_i)$. The best position for the swarm is denoted by *g* and the associated fitness value is $gbest_i = f(g\ )$. Algorithm 1, below, represents the basic PSO algorithm [5].

Algorithm 1: Basic PSO

```
Input: the fitness function, $f$, $\varphi_1$ and $\varphi_2$
Output: the best position, $x$
Step 1: Initialize the particles $x_i^0$ randomly
Step 2: Find the fitness $f(x_i^t)$ for each particle using the
    current position.
Step 3: Compare the fitness of each individual to its best
    fitness such that:
        If $f(x_i^t) < pbest_i$
            $pbest_i = f(x_i^t)$
            $p_i = x_i^t$
```

E. Al-Daoud is with faculty of Science and Information Technology, Computer Science Department, Zarka University, Jordan (Tel +962-796680005, e-mail:essamdz@zpu.edu.jo).

Step 4: Compare the fitness of each particle to the global:

$$\text{If } f(x_i^t) < gbest_i$$
$$gbest_i = f(x_i^t)$$
$$g = x_i^t$$

Step 5: if the condition is satisfied, stop and return g

Step 6: Update the velocity vector for each particle as following:

$$v_i^{t+1} = v_i^t + \varphi_1 rand()(p_i - x_i^t) + \varphi_2 rand()(g - x_i^t)$$

Step 7: Update the positions as following:

$$x_i^{t+1} = x_i^t + v_i^{t+1}$$
$$t = t + 1$$

Step 8: Return to Step 2

φ_1 and φ_2 are two constants, where $\varphi_1 \in [1.5, 2]$ and $\varphi_2 \in [2, 2.5]$, and represent cognitive and social acceleration coefficients, respectively. The random values in the velocity equation can be any value in the range (0, 1).

III. Variants of Particle Swarm

Many variations have been developed to improve the quality of the solution or the speed of the convergence of the PSO. Clerc and Kennedy [6] suggested a new velocity equation to control the convergence properties of the particles using a constriction factor:

$$v_i^{t+1} = \chi(v_i^t + \varphi_1 rand()(p_i - x_i^t) + \varphi_2 rand()(g - x_i^t)) \tag{1}$$

with

$$\chi = 2k/(|\,2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\,|) \tag{2}$$

where $k \in [0,1]$, $\varphi = \varphi_1 + \varphi_2$ and $\varphi > 4$. Usually, φ_1 =2.05, φ_2 =2.05 and k=1. This variant is known the canonical POS. Velocity clamping is another equation to control the global exploration of the particle:

$$v_i^{t+1} = \begin{cases} v_i^{t+1} & if \quad v_i^{t+1} < v_i^{\max} \\ v_i^{\max} & Other \end{cases} \tag{3}$$

The value of $v_i^{\max}$ is proportional to the difference between the maximum and the minimum position of the particle i [7]. Mendes et al. [8] proposed a fully informed particle swarm (FIPS). This method uses information from all the neighbours of particle i; thus, the new velocity equation becomes:

$$v_i^{t+1} = \chi\left[v_i^t + \sum_{k \in N} \varphi_k rand()(p_k - x_i^t)\right] \tag{4}$$

where N is the set of the neighbourhoods of the particle i. The neighbourhoods can take on any topology such as ring, global, star, hybrid, tree network or even a randomly generated topology [9]. PSO can be modified to handle the binary data by using a sigmoid function as in the following formula [10]:

$$v = sign\left(\frac{1}{1 + e^{-v_{ij}^{t+1}}}\right) \tag{5}$$

$$x_{ij}^{t+1} = \begin{cases} 1 & rand() < v \\ 0 & Other \end{cases} \tag{6}$$

IV. New Particle Swarm method

The new velocity and position equations that are introduced in this section can be used for feature selection by combining them with any classification method; alternatively, they can also be used for binary data optimisation. The equations can be described as follows:

$$v_{ij}^{t+1} = \varphi(\mu(v_{ij}^t), \sigma(\pi(p_{ij}, x_{ij}^t)), \sigma(\pi(g, x_{ij}^t))) \tag{7}$$

$$\phi(y_1, y_2, y_3) = \begin{cases} 1 & if \quad \exists k \;\; .\ni. \quad y_k = 1 \\ 0 & Other \end{cases} \tag{8}$$

$$\pi(x, y) = \begin{cases} 1 & if \quad x \neq y \\ 0 & if \quad x = y \end{cases} \tag{9}$$

$$\sigma(\overline{x}) = \begin{cases} x_j = 0 & if \quad x_j = 0 \quad or \quad rand() < c \\ 1 & Other \end{cases} \tag{10}$$

$$x_i^{t+1} = \begin{cases} x_{ij}^t & if \quad v_{ij}^{t+1} = 0 \\ not(x_{ij}^t) & Other \end{cases} \tag{11}$$

The value of c is in the interval (0, 1). If c is close to 1, the optimisation process is fast but the result is not accurate. On the other hand, if c is close to 0, the optimisation process is slow but the result is more accurate. The function μ is used to reset some entries in the vector v_i^t randomly to 0. The suggested procedure can be used for feature selection by combining it with any suitable classifier. Let the particle i be represented as $p_i = (b_1, b_2, ..., b_n)$, where n is the number of the features, b_j is 0 if the feature j is not selected and 1 if the feature j is selected. Let η be an accuracy function $\eta : f \to r$, where f represents the selected features and r is the classification accuracy, and let λ be the feature function $\lambda : p_i \to f$. Thus, the quality function can be represented as follows:

$$\text{The quality} = \eta(\lambda(p_i)) \tag{12}$$

SVM is a classification method which tries to find the optimal hyper-planes between different classes. The decision function is:

$$F(x) = \sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b \tag{13}$$

where K is a suitable kernel function. In this research, the used kernel is a combination of a polynomial and radial basis function:

$$K(x,y) = \exp(-\alpha \|x_1 - y_1\|^2) + (x_2.y_2 + 1)^{\beta} \tag{14}$$

where $\alpha \in \{0.07, 0.7, 0.4, 0.7, 5, 10, 100\}$, $\beta \in \{2, 3, 4, 5, 6, 7, 8\}$, $x=[x_1\ x_2]$ and $y= [y_1\ y_2]$. x and y are the features extracted by $\lambda(p_i)$. Thus, each particle consists of two parts: the first part is the set of features that will be used with the radial basis kernel, and the second part will be used with the polynomial kernel.

V. Intrusion Detection Dataset

In this study, we will use the KDD CUP 1999 intrusion detection contest data [15]. This data was prepared by the 1998 DARPA Intrusion Detection Evaluation program by MIT Lincoln Labs [MIT]. The program acquired 9 weeks' of raw transmission control protocol (TCP) dump data. The raw data was processed into approximately 5 million connection records. The data set contains 24 attack types. All of these attacks fall into 4 main categories as described in the Introduction of this paper. Table I summarizes the recorded attacks.

Every record in the dataset has 41 features that are shown in Table II. Features 2, 3 and 4 are converted into numbers; for example, the second feature 'Protocol Type' is replaced by 1, 2 or 3 instead of the values TCP, UDP (user datagram protocol) or ICMP (internet control message protocol), respectively.

VI. Experimental Results

The performance of the intrusion detection system is calculated using a detection rate and a false alarm rate as follows:

$$\text{Detection Rate} = \frac{TP}{TP + FN} \tag{15}$$

$$\text{False Alarm Rate} = \frac{FP}{TN + FP} \tag{16}$$

TABLE I
KDD Dataset Categories

Attack	Category	# Samples
normal	Normal	97277
smurf	Dos	280790
neptune		107201
back		2203
teardrop		979
pod		264
land		21
satan	Probe	1589
ipsweep		1247
portsweep		1040
nmap		231
warezclient	R2L	1020
guess_passwd		53
warezmaster		20
imap		12
ftp_write		8
multihop		7
phf		4
spy		2
buffer_overflow	U2R	30
rootkit		10
loadmodule		9
perl		3

TABLE II
KDD CUP'99 Features

No.	Features	No.	Features
1	duration	22	is_guest_login
2	protocol_type	23	count
3	service	24	srv_count
4	flag	25	serror_rate
5	src_bytes	26	srv_serror_rate
6	dst_bytes	27	rerror_rate
7	land	28	srv_rerror_rate
8	wrong_fragt.	29	same_srv_rate
9	urgent	30	diff_srv_rate
10	hot	31	srv_diff_h_rate
11	num_fail_login	32	host_count
12	logged_in	33	host_srv_count
13	nu_comprom	34	h_same_sr_rate
14	root_shell	35	h_diff_srv_rate
15	su_attempted	36	h_src_port_rate
16	num_root	37	h_srv_d_h_rate
17	nu_file_creat	38	h_serror_rate
18	nu_shells	39	h_sr_serro_rate
19	nu_access_files	40	h_rerror_rate
20	nu_out_cmd	41	h_sr_rerro_rate
21	is_host_login		

where FN is the number of false negatives , FP is the number of false positives, TN is the number of true negatives and TP is the number of true positives. Particle swarm and multi-class SVM are applied on 70% of the KDD dataset; the remaining 30% of the dataset is used for testing. The experiment is repeated several times to find the detection rate or false alarm rate for one class of the KDD dataset (R), as follows:

$$R = \alpha s \times \beta s \times \text{Classes} \tag{17}$$

In this study, the length of set αs is 7 and of set βs is 7. The number of classes is 5, and these are: Normal, DoS, Probe, R2L and U2R. Thus, the experiment is repeated 245 times for each class and 245*10=2450 times for all classes and both performance measurements. Moreover, 10 particles with 10 epochs are applied, therefore, the total number of executed experiments is 2450*10*10=245000; for training purposes, however, individual records can be tested very quickly. Tables III and IV compare the method proposed by this study with previous methods using the detection rate and false alarm rate, respectively.

TABLE III
THE DETECTION RATE OF KDD DATASET

	Dos	Probe	U2R	R2L	Normal
Proposed	**97.9**	**98.6**	68.9	19.5	**99.8**
KDD Winner [11]	97.1	83.3	13.2	8.4	99.5
Multi-SVM [12]	96.8	75	5.3	4.2	99.6
PNrule [13]	96.9	73.2	6.6	10.7	99.5
Random Field [14]	97.4	**98.6**	**86.3**	**29.6**	--

TABLE IV
THE FALSE ALARM RATE OF KDD DATASET

	Dos	Probe	U2R	R2L	Normal
Proposed	0.07	3.1	**0.05**	**0.35**	**15.5**
KDD Winner[11]	0.1	35.2	28.6	1.2	27.0
Multi- SVM[12]	0.1	11.7	47.8	35.4	27.8
PNrule[13]	**0.05**	7.5	89.5	12.0	27.0
Random Field [14]	0.07	**0.91**	**0.05**	**0.35**	--

VII. CONCLUSION

Intrusion detection systems are used to improve enterprise network security, acting as a ‘second line of defence’. The perfect intrusion detection system would have a 100% detection rate together with a 0% false positive rate which is very difficult to achieve. This study investigates and evaluates the performance of new techniques for intrusion detection using the KDD CUP 1999 dataset. The suggested techniques are based on particle swarm optimisation, and nonlinear, multi-class and multi-kernel support vector machines. The results show that our model outperforms other state-of-the-art methods. Further work needs to be done to apply the suggested techniques to real-world commercial intrusion detection systems and to enhance the computation speed, detection and false alarm rates.

REFERENCES

[1] H. Zhang, X. Wang, Y Wang, "Network Connection Based Intrusion detection Using Rough Set Classification," Proceedings 2006 International Conference on Communications, Circuits and Systems 3: 2128 – 2132, 2006.

[2] A.O. Adetunmbi, B.K. Alese, O.S. Ogundele, S.O. Falaki, A "Data Mining Approach to Network Intrusion Detection," Journal of Computer Science & Its Applications, 14 (2): 24 -37, 2007.

[3] D. M. Farid, N. Harbi, M. Z. Rahman, “Combining Naïve Bayes and Decision Tree for Adaptive Intrusion Detection,” International Journal of Network Security & Its Applications, 2(2):12-25, 2010.

[4] D. M. Farid, N. Harbi, M. Z. Rahman, “Anomaly Network Intrusion Detection Based on Improved Self Adaptive Bayesian Algorithm,” Journal of Computers, Academy Publisher, 5(1):, 23- 31, 2010.

[5] A.W. Mohemmed, N.C. Sahoo, T.K. Geok, “A new particle swarm optimization based algorithm for solving shortest-paths tree problems, *in IEEE CEC 2007*, pp. 3221–3225, 2007.

[6] M. Clerc, J. Kennedy, "The particle swarm explosion, stability, and convergence in a multidimensional complex space". IEEE Transactions on Evolutionary Computation 6(1): 58-73, 2002.

[7] D. P. Rini, S. M. Shamsuddin, S. S. Yuhaniz, " Particle Swarm Optimization: Technique, System and Challenges," international Journal of Computer Applications, 14(1): 0975 – 8887, 2011.

[8] R.Mendes, J.Kennedy, J. Neves," The fully informed particle swarm: Simpler, maybe better," IEEE Transactions on Evolutionary Computation 8(3) 204 – 210, 2004.

[9] R. Parimala, R. Nallaswamy, "Feature selection using a novel particle swarm optimization and its variants," *I.J. Information Technology and Computer Science,* 5: 16-24, 2012.

[10] L. Y. Chuang, S. W. Tsai, C.H. Yang, "Catfish Binary Particle Swarm Optimization for Feature Selection" *International Conference on Machine Learning and Computing*, *IPCSIT, IACSIT Press, Singapore, 3: 40-44, 2011.*

[11] B. Pfahringer, "Winning the KDD99 Classification Cup: Bagged Boosting," *SIGKDD Explorations,* 1: 65–66, 2000.

[12] T. Ambwani, "Multi class support vector machine implementation to intrusion detection," in *Proc. of IJCNN*: 2300-2305. 2003.

[13] R. Agarwal, M. V. Joshi, "PNrule: A New Framework for Learning Classifier Models in Data Mining," In *Proceedings of First SIAM Conference on Data Mining*, Chicago, April 2001. Expanded version available as IBM Research Division Report, RC 21719, April 2000.

[14] K. K. Gupta, B. Nath, and R. Kotagiri, "Layered Approach using Conditional Random Fields for Intrusion Detection," *IEEE Transactions on Dependable and Secure Computing,* vol. 5, 2008.

[15] KDDCUP99: http://kdd.ics.uci.edu/databases/kddcup99/