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 attacks, 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 $f : X \rightarrow R$, 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 $gbest = f(g)$. 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:

\[
\text{If } f(x_i^t) < pbest_i \\
pbest_i = f(x_i^t) \\
p_i = x_i^t
\]
Step 4: Compare the fitness of each particle to the global:
If \( f(x_j^t) < g_{best_j} \)
\[ g_{best_j} = f(x_j^t) \]
Step 5: if the condition is satisfied, stop and return \( g \)
Step 6: Update the velocity vector for each particle as following:
\[ v_{i}^{t+1} = v_i^t + \phi_1 \text{rand}((p_i - x_i^t) + \phi_2 \text{rand})(g - x_i^t) \]
Step 7: Update the positions as following:
\[ x_i^{t+1} = x_i^t + v_{i}^{t+1} \]

Step 8: Return to Step 2

\( \phi_1 \) and \( \phi_2 \) are two constants, where \( \phi_1 \in [1.5, 2] \) and \( \phi_2 \in [2, 2.5] \), 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:

\[ v_{i}^{t+1} = \chi(v_i^t + \phi_1 \text{rand}((p_i - x_i^t) + \phi_2 \text{rand})(g - x_i^t)) \]

with
\[ \chi = 2k / (1 - \phi - \sqrt{\phi^2 - 4\phi}) \]

where \( k \in [0,1] \), \( \phi = \phi_1 + \phi_2 \), and \( \phi > 4 \). Usually, \( \phi_1 = 2.05 \), \( \phi_2 = 2.05 \) and \( k = 1 \). This variant is known the canonical POS.

Velocity clamping is another equation to control the global exploration of the particle swarm (FIPS). This method uses information from all the neighbours of particle \( i \); thus, the new velocity equation becomes:

\[ v_{i}^{t+1} = \chi \left[ v_i^t + \sum_{k \in N} \phi_1 \text{rand}((p_k - x_i^t) \right] \]

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 \([10]\):

\[ v = \text{sign} \left( \frac{1}{1 + e^{-v_{ij}^t}} \right) \]

### 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:

\[ v_{ij}^{t+1} = \phi(\mu(v_{ij}^t), \sigma(\pi(p_j, x_i^t)), \sigma(\pi(g, x_i^t))) \]

\[ \phi(\mu, x_j, y_j) = \begin{cases} 1 & \text{if } \exists k \neq j \quad y_k = 1 \\ 0 & \text{Other} \end{cases} \]

\[ \pi(x, y) = \begin{cases} 1 & \text{if } x \neq y \\ 0 & \text{if } x = y \end{cases} \]

\[ \sigma(x) = \begin{cases} 1 & x_j = 0 \text{ or } \text{rand}() < c \\ 0 & \text{Other} \end{cases} \]

\[ x_{ij}^{t+1} = \begin{cases} x_j^t & \text{if } v_{ij}^{t+1} = 0 \\ \text{not}(x_j^t) & \text{Other} \end{cases} \]

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 \( \mu \) 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 \( p_i = (b_1, b_2, \ldots, b_n) \), 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 \( \eta \) be an accuracy function \( \eta : f \rightarrow r \), where \( r \) represents the selected features and \( r \) is the classification accuracy, and let \( \lambda \) be the feature function \( \lambda : p_i \rightarrow f \). Thus, the quality function can be represented as follows:

\[ \text{The quality} = \eta(\lambda(p_i)) \]
SVM is a classification method which tries to find the optimal hyper-planes between different classes. The decision function is:

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

(13)

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

\[ K(x, y) = \exp(-\alpha \|x_i - y_i\|^p) + (x_2, y_2 + 1)^\beta \]  

(14)

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:

\[ \text{Detection Rate} = \frac{TP}{TP + FN} \]  

(15)

\[ \text{False Alarm Rate} = \frac{FP}{TN + FP} \]  

(16)

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:

\[ R = \alpha s \times \beta s \times \text{Classes} \]  

(17)

In this study, the length of set \( \alpha s \) is 7 and of set \( \beta s \) is 7. The number of classes is 5, and these are: Normal, DoS, Probe, R2L and U2R. Thus, the experiment is repeated 245 times.
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=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>
<thead>
<tr>
<th>TABLE III</th>
<th>THE DETECTION RATE OF KDD DATASET</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dos</td>
</tr>
<tr>
<td>Proposed</td>
<td>97.9</td>
</tr>
<tr>
<td>Multi-SVM [12]</td>
<td>96.8</td>
</tr>
<tr>
<td>PNrule [13]</td>
<td>96.9</td>
</tr>
<tr>
<td>Random Field [14]</td>
<td>97.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE IV</th>
<th>THE FALSE ALARM RATE OF KDD DATASET</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dos</td>
</tr>
<tr>
<td>Proposed</td>
<td>0.07</td>
</tr>
<tr>
<td>KDD Winner [11]</td>
<td>0.1</td>
</tr>
<tr>
<td>Multi-SVM [12]</td>
<td>0.1</td>
</tr>
<tr>
<td>PNrule [13]</td>
<td>0.05</td>
</tr>
<tr>
<td>Random Field [14]</td>
<td>0.07</td>
</tr>
</tbody>
</table>

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