Network Intrusion Detection Using Multiclass Support Vector Machine

Arvind Mewada
*Computer Science department Maulana Azad Institute of National Technology, MANIT Bhopal, India*, mewadatec@yahoo.co.in

Prafful Gedam
*Computer Science dept. Technocrat Institute of Technology, RGPV Bhopal, India*, Praffulit@gmail.com

Shamaila Khan
*Computer Science dept. Radharaman Institute of Science and Technology, RGPV Bhopal, India*, shamilak@gmail.com

M. Udayapal Reddy
*Computer Science department Maulana Azad Institute of National Technology, MANIT Bhopal, India*, udayapalreddy@gmail.com

Follow this and additional works at: https://www.interscience.in/ijcct

**Recommended Citation**

This Article is brought to you for free and open access by Interscience Research Network. It has been accepted for inclusion in International Journal of Computer and Communication Technology by an authorized editor of Interscience Research Network. For more information, please contact sritampatnaik@gmail.com.
Network Intrusion Detection Using Multiclass Support Vector Machine

Abstract—Intrusion detection is a topic of interest in current scenario. Statistical IDS overcomes many pitfalls present in signature based IDS. Statistical IDS uses models such as NB, C4.5 etc for classification to detect Intrusions. Multiclass Support Vector Machine is able to perform multiclass classification. This paper shows the performance of MSVM (1-versus-1, 1-versus-many and Error Correcting Output Coding (ECOC)) and it’s variants for statistical NBIDS. This paper explores the performance of MSVM for various categories of attacks.

Keywords-NBIDS; KDDCUP99; MSVM; Intrusion Detection; Kernel Function; RBF.

I. INTRODUCTION

This Intrusion detection (ID) is the processing procedure of identification and response to the action of malicious use computers and network resources [1]. Intrusion Detection Systems (IDS) are computer programs that tries to perform intrusion detection by comparing observable behavior against suspicious patterns, preferably in real-time. Intrusion is primarily network based activity [2]. The primary aim of Intrusion Detection Systems (IDS) is to protect the availability, confidentiality and integrity of critical networked information systems.

II. TECHNIQUES OF IDS

A. Host-Based IDS and Network Based IDS

IDS can be classified based on which events they monitor, how they collect information and how they deduce from the information that an intrusion has occurred. IDSs that operates on a single workstation are known as host intrusion detection system (HIDS), A HBIDS adds a targeted layer to security to particularly vulnerable or essential systems, it monitors audit trails and system logs for suspicious behaviors [3] while A network-based IDS monitors network traffic for particular network segments or devices and analyzes network, transport, and application protocols to identify suspicious activity.

B. Misuse and Anomaly detection Techniques

Misuse detection uses the “signatures” of known attacks to identify a matched activity as an attack instance. Misuse detection has low false positive rate, but unable to detect novel attacks. It is more accurate but it lacks the ability to identify the presence of intrusions that do not fit a pre-defined signature, resulting not adaptive [4]. Misuse detection discovers attacks based on patterns extracted from known intrusions [5]. Statistical based IDS: Statistical detection techniques assume that all intrusive activities are necessarily anomalous. This means that if we could establish a "normal activity profile" for a system, we could, in theory, flag all system states varying from the established profile by statistically significant amounts as intrusion attempts. Anomaly detection is based on modeling the normal activity of the computer system. Unfortunately, the acquisition of profiles of normal activity is not an easy task. The audit records used to produce the profiles of normal activity may contain traces of intrusions leading to misdetections, and also activities of legitimate users often deviate from their normal profile as modeled, leading to high false alarm rates [6].

C. Network Attack in IDS

- Denial of service[20] (DOS): In this type of attack an attacker makes some computing or memory resources too busy or too full to handle legitimate requests, or denies legitimate users access to a machine. Examples are Apache2, Back, Land, Mail bomb, SYN Flood, Ping of death, Process table, Smurf, Teardrop.
- Remote to user[7] (R2L): In this type of attack an attacker who does not have an account on a remote machine sends packets to that machine over a network and exploits some vulnerability to gain local access as a user of that machine. Examples are Dictionary, Ftp_write, Guest, Imap, Named, Phf, Send mail, Xlock.
- User to root (U2R): In this type of attack an attacker starts out with access to a normal user account on the system and is able to exploit system vulnerabilities to gain root access to the system. Examples are Eject, Loadmodule, Ps, Xterm, Perl, and Fdformat.
- Probing: In this type of attacks an attacker scans a network of computers to gather information or find known vulnerabilities. An attacker with a map of

International Journal of Computer and Communication Technology (IJCCT), ISSN: 2231-0371, Vol-1, Iss-4

Arvind Mewada
Computer Science department
Maulana Azad Institute of National Technology, MANIT
Bhopal, India
mewadatec@yahoo.co.in

Prafful Gedam
Computer Science dept.
Technocrate Institute of Technology,
RGPV Bhopal, India
Praffulit@gmail.com

Shamila Khan
Computer Science dept.
Radharaman Institute of Science and Technology,
RGPV Bhopal, India
shamilak@gmail.com

M. Udayapal Reddy
Computer Science department
Maulana Azad Institute of National Technology, MANIT
Bhopal, India
udayapalreddy@gmail.com

Prafful Gedam
Computer Science Dept.
Technocrat Institute of Technology,
RGPV Bhopal, India
Praffulit@gmail.com

Shamila Khan
Computer Science dept.
Radharaman Institute of Science and Technology,
RGPV Bhopal, India
shamilak@gmail.com

M. Udayapal Reddy
Computer Science department
Maulana Azad Institute of National Technology, MANIT
Bhopal, India
udayapalreddy@gmail.com
machines and services that are available on a network can use this information to look for exploits. Examples are Ipsweep, Mscan, Saint, Satan, and Nmap.

III. SUPPORT VECTOR MACHINE

The theory of Support Vector Machine (SVM) is from statistics which is proposed by Vapnik. The basic principle of SVM is finding the optimal linear hyperplane in the feature space that maximally separates the two target classes. For linearly separable and non-separable data, it can be translated into quadratic programming (QP) and can get an only limit point. In the case of non-linear, SVM can map the input to a high-dimensional feature space by using non-linear mapping and then the linear hyperplane can be found [8].

A. SVM classification model

The basic principle of SVM is finding the optimal linear hyperplane in the feature space that maximally separates the two target classes. The hyperplane which separates the two classes can be defined as:

\[
\omega \cdot x + b = 0
\]

Here \( x_k \) is a group of samples:

\[
\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \subseteq \mathbb{R}^n, y_i \in \{-1, 1\}, \quad k \text{ is the number of styles}; \quad n \text{ is the input dimension}; \quad w \text{ and } b \text{ are nonzero constants} \ [9] \ [10].
\]

B. Linearly separable model

Assume a training set:

\[
\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \subseteq \mathbb{R}^n, y_i \in \{-1, 1\}, \quad k \text{ is the number of samples}. \text{ Thus, the problem can be described as:}
\]

Minimize \[
\frac{1}{2} ||w||^2
\]

Subject to \[
y_i (\omega \cdot x_i + b) \geq 1, \quad i = 1, 2, \ldots, k.
\]

This is a quadratic programming (QP) problem. To solve it, we have to introduce Lagrangian:

\[
L(w, b, \alpha) = \frac{1}{2} (w \cdot w) - \sum_{i=1}^{k} \alpha_i \{y_i (\omega \cdot x_i) + b\} - 1
\]

According to the Kuhn-Tucher conditions, we obtain

\[
\sum_{i=1}^{k} \alpha_i = 0, \quad \alpha_i \geq 0 \quad i = 1, 2, \ldots, k
\]

We can obtain the value of vector \( \omega \) from (3).

C. Non-linear separable model

In the non-linear problem, it can be solved by extending the original set of variables \( x \) in a high dimensional feature space with the map \( \Phi \). Suppose that input vector \( x \in \mathbb{R}^d \) is transformed to feature vector \( \Phi(x) \) by a map \( \Phi: \mathbb{R}^d \rightarrow \mathbb{H} \), then we can find a function \( K(\mathbb{R}^d, \mathbb{R}^d) \rightarrow \mathbb{R} \) that satisfies condition \( K(x, x) = \Phi(x), \Phi(x) \), so we can replace the inner-product between two vectors \( (x_i, x_j) \) by \( K(x_i, x_j) \) and the QP problem expressed by (4) becomes:

\[
\maximize \sum_{i=1}^{k} \alpha_i - \frac{1}{2} \sum_{i=1}^{k} \sum_{j=1}^{k} \alpha_i \alpha_j y_i y_j K(x_i, x_j)
\]

\[
\text{subject to } \sum_{i=1}^{k} \gamma_i \alpha_i = 0, \quad \alpha_i \geq 0 \quad (i = 1, 2, \ldots, k)
\]

The optimal separating hyperplane (5) can be rewritten as:

\[
f(x) = \sum_{\text{sup vector}} \alpha_i y_i \Phi(x_i) \Phi(x) + b
\]
D. Multiclass support vector machine on non-linear separable model

Support vector machines are formulated for two class problems. But because support vector machines employ direct decision functions, an extension to multiclass problems is not straightforward. There are roughly four types of support vector machines that handle multiclass problems. But we use here only three for our research work (1-vs-many), pair wise coupling (1-vs-1), and error-correcting output coding (ECOC) [11].

- One per class (OPC) also known as “one against others.” OPC trains K binary classifiers, each of which separates one class from the other (K - 1) classes. Given a point X to classify, the binary classifier with the largest output determines the class of X.
- The Pair wise coupling (PWC) constructs K (K-1)/2 pair wise binary classifiers. The classifying decision is made by aggregating the outputs of the pairwise classifiers.
- Error-correcting output coding (ECOC) [19] [12] used to reduce classification error by exploiting the redundancy of the coding scheme. ECOC employs a set of binary classifiers assigned with codeword’s such that the Hamming distance between each pair is far enough apart to enable good error correction.

IV. DATASET AND EXPERIMENTS

The KDD Cup 1999 uses a version of the data on which the 1998 DARPA Intrusion Detection Evaluation Program was performed. Each instance in the KDD Cup 1999 datasets contains 41 features that describe a connection. Features 1-9 stands for the basic features of a packet, 10-22 for content features, 23-31 for traffic features and 32-41 for host based features. There are 38 different types attack in training and test data together and these types of attack fall into four main categories: probe, denial of service (DoS), remote to local (R2L) and user to root (U2R) [14]. We have taken 26 total no of classes to classification.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Train Records</th>
<th>Test Records</th>
</tr>
</thead>
<tbody>
<tr>
<td>KDD CUP99</td>
<td>48,984,31(4.9 million)</td>
<td>3,11,029(0.3 million)</td>
</tr>
</tbody>
</table>

The environment used for the experiment is Pentium (IV 3GH) processor, 512 MB RAM, running window XP (SP2) based multiclass SVMlight [15]. We have implemented Cauchy[22] and ANOVA[21] kernel functions. For Cauchy and ANOVA kernels, the accuracy for all the three MSVM methods was very low. We exclude the results for these. The experiment using RBF[16][17][18] kernel function for intrusion detection (multiclass classification) with parameters as g=0.001, c=1, q=50, n=40 gave the intrusion detection rate 92.05%, 90.65% & 92.00% for one-vs.-one, one-vs.-many & ECOC MSVM methods respectively.

V. EVALUATION MATRICS

The Evaluation Metrics mainly used following steps to evaluate the performance of classifier:

- **True positives:** The true positives (TP) and true negatives (TN) are correct classifications.
- **False positive:** A false positive (FP) occurs when the outcome is incorrectly predicted as yes (or positive) when it is actually no (negative).
- **False negative:** A false negative (FN) occurs when the outcome is incorrectly predicted as negative when it is actually positive.
- **Recall:** The percentage of the total relevant documents in a database retrieved by search. If user knew that there were 1000 relevant documents in a database and his search retrieved 100 of these relevant documents, his recall would be 10%.

\[
\text{Recall} = \frac{TP}{TP + FN}
\]

- **Precision:** The percentage of relevant documents in relation to the number of documents retrieved. If search retrieves 100 documents and 20 of these are relevant, then precision is 20%.

\[
\text{Precision} = \frac{TP}{TP + FP}
\]

- **F-measure:** The harmonic mean of precision and recall

\[
F = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}
\]

The true positive rate is TP divided by the total number of positives, which are TP + FN. The false positive rate is FP divided by the total number of negatives, FP + TN. ROC area in ROC analysis we plot true positive ratio (tpr) against, false positive ratio (fpr). The overall success rate is the number of correct classifications divided by the total number of classifications:

\[
\frac{TP + TN}{TP + TN + FP + FN}
\]

In a multiclass prediction, the result on a test set is often displayed as a two dimensional confusion matrix with a row and column for each class. Each metrics element shows the number of test examples for which the actual class is the row and the predicted class is the column. Good results correspond to large numbers down the main diagonal and small, ideally zero, off-diagonal elements

<table>
<thead>
<tr>
<th>Predicted Class</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual Class</td>
<td>True Positive</td>
<td>False Negative</td>
</tr>
<tr>
<td>False Negative</td>
<td>False Positive</td>
<td>True Negative</td>
</tr>
</tbody>
</table>

TABLE II. CONFUSION MATRICS

International Journal of Computer and Communication Technology (IJCCT), ISSN: 2231-0371, Vol-1, Iss-4
VI. CONCLUSION AND FUTURE WORK

There are many kernel functions which can be used in MSVM for anomaly detection in the IDS. Among those we have implemented Cauchy and ANOVA kernel functions. We performed experiment using Cauchy, ANOVA and RBF kernel function over three types of MSVM and found that the RBF kernel function gives better performance in the MSVM for anomaly detection. The intrusion detection rate is 92.05%, 90.65% & 92.00% for one-vs.-one, one-vs.-many & ECOC methods respectively using RBF kernel function. This result can further be improved by using composite (combine two kernel function) of two or more kernel functions.

REFERENCES


