An Enhanced Support Vector Machine Model for Intrusion Detection

JingTao Yao, Songlun Zhao, and Lisa Fan

Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada S4S 0A2
[jtyao,zhao200s,fan]@cs.uregina.ca

Abstract. Design and implementation of intrusion detection systems remain an important research issue in order to maintain proper network security. Support Vector Machines (SVM) as a classical pattern recognition tool have been widely used for intrusion detection. However, conventional SVM methods do not concern different characteristics of features in building an intrusion detection system. We propose an enhanced SVM model with a weighted kernel function based on features of the training data for intrusion detection. Rough set theory is adopted to perform a feature ranking and selection task of the new model. We evaluate the new model with the KDD dataset and the UNM dataset. It is suggested that the proposed model outperformed the conventional SVM in precision, computation time, and false negative rate.

Keywords: Intrusion detection, support vector machine, feature selection, rough sets.

1 Introduction

Various intrusion detection systems are studied and proposed to meet the challenges of a vulnerable internet environment [1, 3]. It is not an exaggerated statement that an intrusion detection system is a must for a modern computer system. Intrusion detection technologies can be classified into two groups: misuse detection and anomaly detection [1]. A misuse detection system detects intrusion events that follow known patterns. These patterns describe a suspect set of sequences of actions or tasks that may be harmful. The main limitation of this approach is that it cannot detect possible novel intrusions, i.e., events that have never happened and captured previously. An anomaly detection based system analyzes event data and recognizes patterns of activities that appear to be normal. If an event lies outside of the patterns, it is reported as a possible intrusion. It is considered as a self-learning approach. We focus on anomaly intrusion detection in this study.

Many artificial intelligence techniques have been used for anomaly intrusion detection. Qiao *et al.* [12] presented an anomaly detection method by using a hidden Markov model to analyze the UNM dataset. Lee *et al.* [9] established an anomaly detection model that integrates the association rules and frequency

episodes with fuzzy logic to produce patterns for intrusion detection. Mohajeran *et al.* [10] developed an anomaly intrusion detection system that combines neural networks and fuzzy logic to analyze the KDD dataset. Wang *et al.* [14] applied genetic algorithms to optimize the membership function for mining fuzzy association rules.

Support Vector Machines (SVM) have become one of the popular techniques for anomaly intrusion detection due to their good generalization nature and the ability to overcome the curse of dimensionality [2, 13]. Although there are some improvements, the number of dimensions still affects the performance of SVM-based classifiers [2]. Another issue is that an SVM treats every feature of data equally. In real intrusion detection datasets, many features are redundant or less important [8]. It would be better if we consider feature weights during SVM training. Rough set theory has proved its advantages on feature analysis and feature selection [5, 6, 16]. This paper presents a study that incorporates rough set theory to SVM for intrusion detection. We propose a new SVM algorithm for considering weighting levels of different features and the dimensionality of intrusion data. Experiments and comparisons are conducted through two intrusion datasets: the KDD Cup 1999 dataset[1] and the UMN dataset that was recorded from the trace of systems calls coming from a UNIX system[2].

2 A Brief Overview of Support Vector Machines

An SVM model is a machine learning method that is based on statistical learning theories [13]. It classifies data by a set of support vectors that represent data patterns.

A general two-class classification problem is to find a discriminant function $f(\boldsymbol{x})$, such that $y_i = f(\boldsymbol{x_i})$ given N data samples $(\boldsymbol{x_1}, y_1) \ldots (\boldsymbol{x_i}, y_i) \ldots (\boldsymbol{x_N}, y_N)$. A possible linear discriminant function can be presented as $f(\boldsymbol{x}) = sgn(\boldsymbol{w} \cdot \boldsymbol{x} - b)$ where $\boldsymbol{w} \cdot \boldsymbol{x} - b = 0$ can be viewed as a separating hyperplane in the data space. Therefore, choosing a discriminant function is to find a hyperplane having the maximum separating margin with respect to the two classes. The final linear discriminant is formulated as $f(\boldsymbol{x}) = sgn(\sum_{i=1}^{l} \alpha_i y_i (\boldsymbol{x_i} \cdot \boldsymbol{x} - b)$, where l is the number of training records, $y_i \in \{-1, +1\}$ is the label associated with the training data, $0 \leq \alpha_i \leq C$ (constant $C > 0$), and $\boldsymbol{x_i}$ is the support vectors.

When the surface separating two classes is not linear, we can transform the data points to another higher dimensional space such that the data points will be linear separable. The nonlinear discriminant function of SVM is:

$$f(\boldsymbol{x}) = sgn(\sum_{i=1}^{l} \alpha_i y_i K(\boldsymbol{x_i}, \boldsymbol{x}) + b), \tag{1}$$

where $K(\boldsymbol{x_i}, \boldsymbol{x})$ is the kernel function that is used to transform data points.

[1] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
[2] http://www.cs.unm.edu/~immsec/systemcalls.htm

Algorithm 1: Feature Weights Calculation

Input : Dataset D.
Output: A weight vector W.
Find out all the reducts of D using rough sets;
$N_{feature} \leftarrow$ *number of features in D*;
$N_{reduct} \leftarrow$ *number of reducts of D*;
//Initialize the weight of each feature.
for $(i \leftarrow 0$ **to** $N_{feature})$ **do**
  $w_i \leftarrow 0$;
end
// Calculate the weight of each feature.
for $(i \leftarrow 0$ **to** $N_{feature})$ **do**
  for $(j \leftarrow 0$ **to** $N_{reduct})$ **do**
    if $(feature\ i\ in\ the\ j^{th}\ reduct\ R_j)$ **then**
      $m \leftarrow$ *number of features in* R_j;
      $w_i \leftarrow w_i + \frac{1}{m}$;
    end
  end
end
Scale the values of feature weights into the interval $[0, 100]$;

3 Enhancing SVM Learning with Weighted Features

Various SVM kernel functions are proposed for users to choose from for different applications [2, 7]. The most common kernel functions are the linear function, polynomial function, sigmoid function, and radial basis function. These kernel functions do not consider the differences between features of data. From the general SVM kernel function format $K(\boldsymbol{x_i}, \boldsymbol{x})$, we can see that all features of the training or test datasets are treated equally. Treating all features equally may not be efficient and it may affect the accuracy of SVM. A possible solution to consider the importance of different features is to add weights to a kernel function. The weights are used to measure the importance of each feature. A generic form of the new kernel function is formulated as $K(\boldsymbol{wx_i}, \boldsymbol{wx})$, where $\boldsymbol{w}$ is a vector consisting of weights of features of data set. A nonlinear discriminant function with feature weights is formulated as,

$$f(\boldsymbol{x}) = sgn(\sum_{i=1}^{l} \alpha_i y_i K(\boldsymbol{wx_i}, \boldsymbol{wx}) + b). \tag{2}$$

This enhanced kernel is independent to particular kernel functions. For different applications, one may choose the most suitable kernel function to apply the feature weights on. We use rough set theory to calculate and generate these weights from training data in this study. The basic principles of weight calculation are: 1) if a feature is not in any reducts then the weight of this feature is 0; 2) the more times a feature appears in the reducts, the more important this feature is; 3) the fewer the number of features in a reduct, the more important

these features appearing in this reduct are. If a reduct has only one feature, the feature belonging to this reduct is the most important.

Based on the above principles, we propose an algorithm as depicted in Algorithm 1 that adopts rough set theory to rank features and calculate feature weights. After the feature ranking process, we consider those features with 0 weights as the least important features and delete them. In Algorithm 1, feature ranking and feature selection are conducted in the same process.

4 Experiments and Results Analysis

Two datasets, KDD and UNM, are used in experiments to evaluate the performance of the proposed new model. The KDD dataset consists of network connection records generated by a TCP/IP dump. It contains $4,940,000$ connection records. There are 41 features in each record. 10% of the original data are training data with a label which identifies which category the record belongs. We only discuss binary classification.

The system call dataset is from the University of New Mexico (UNM). It consists of $4,298$ normal traces and 1,001 intrusion traces. Each trace is the list of system calls issued by an lpr process from the beginning of its execution to the end. There are 182 different system calls in the dataset.

Four measures adapted from information retrieval [4] are used to evaluate the performance of an SVM model: precision = $\frac{A}{A+B}$, recall= $\frac{A}{A+C}$, false negative rate=$\frac{C}{A+C}$, and false positive rate=$\frac{B}{B+D}$. A, B, C, and D represent the number of detected intrusions, not intrusions but detected as intrusions, not detected intrusions, and not detected non-intrusions respectively.

A false negative occurs when an intrusion action has occurred but the system considers it as a non-intrusive behavior. A false positive occurs when the system classifies an action as an intrusion while it is a legitimate action. A good intrusion detection system should perform with a high precision and a high recall, as well as a lower false positive rate and a lower false negative rate. To consider both the precision and false negative rate is very important as the normal data usually significantly outnumbers the intrusion data in practice. To only measure the precision of a system is misleading in such a situation. A poor intrusion detection system may have a high precision but a high false negative rate.

There are four steps in our experiments. The first step is to remove redundant intrusion records. Both KDD and UNM datasets have more intrusion data than normal data. We filter the redundant intrusion records until the two resulting datasets consisting of 1.5% intrusions and 98.5% normal records. There are no obvious feature-value pairs in the dataset. We use a mapping method to convert the dataset to feature-value format. The second step is to use rough set feature ranking and selection to calculate weights of each feature and delete unimportant features. After processing, the number of features of the KDD dataset is narrowed down from 41 to 16 and the UNM dataset is narrowed down from 467 to 9. The third step is to train the SVM. We generate one training set and three test sets for each of the datasets. For the KDD dataset, each set has 50,000 randomly

selected records. Each set has 2,000 records for the UNM dataset. Based on previous research, we choose $\gamma = 10^{-6}$ for RBF kernel $e^{-||\boldsymbol{x_i}-\boldsymbol{x}||^2 \cdot \gamma}$ [17]. The last step is to build a decision function to classify the test data. Experimental results for the two datasets are presented in Table 1 and 2.

Table 1. Comparisons of the experimental results on the KDD dataset

	N_{record}	$N_{feature}$	Precision (%)	False Negative (%)	CPU-second
test set 1					
Conventional SVM	5×10^4	41	99.82	7.69	222.28
Enhanced SVM	5×10^4	16	99.86	6.39	75.63
Improvement		**60.0%**	**0.4%**	**16.9%**	**66.0%**
test set 2					
Conventional SVM	5×10^4	41	99.80	8.25	227.03
Enhanced SVM	5×10^4	16	99.85	6.91	78.93
Improvement		**60.0%**	**0.5%**	**16.2%**	**65.0%**
test set 3					
Conventional SVM	5×10^4	41	99.88	7.45	230.27
Enhanced SVM	5×10^4	16	99.91	5.49	77.85
Improvement		**60.0%**	**0.3%**	**26.3%**	**66.0%**

Table 2. Comparisons of the experimental results on the UNM dataset

	N_{record}	$N_{feature}$	Precision (%)	False Negative (%)	CPU-second
test set 1					
Conventional SVM	2×10^3	467	100	0	1.62
Enhanced SVM	2×10^3	9	100	0	0.28
Improvement		**98%**			**83%**
test set 2					
Conventional SVM	2×10^3	467	100	0	1.71
Enhanced SVM	2×10^3	9	100	0	0.29
Improvement		**98%**			**83%**
test set 3					
Conventional SVM	2×10^3	467	100	0	1.59
Enhanced SVM	2×10^3	9	100	0	0.25
Improvement		**98%**			**84%**

Here are some observations from the experiments. The improvements of performance are consistent for all of the six test sets. This suggests that the new model has a good generalization ability. The new model outperforms the conventional SVM in all three measures, namely, precision, false negative rate and CPU time for the KDD dataset. Although the improvement for precision is only 0.4% on average, the improvement for the other two are significant. The improvements for false negative rate are between 16.2% and 26.8%. The time used for the new model is only one third of the conventional SVM model. For the UNM dataset, the precision and false negative rate of conventional SVM are perfect with no room for improvement. These results are similar to the results from other researchers with other methods on this dataset [9, 15]. However, the CPU time is significantly reduced with the new model.

5 Conclusion

We propose an enhanced SVM model for intrusion detection. The new model adopts rough sets to rank the features of intrusion detection data. Only the

important features will be counted when training an SVM. It is suggested that the proposed new model is effective for the KDD dataset. Although the precision levels of both the conventional SVM and the new model are about the same, the false negative rates of the new model are lower than the conventional SVM model. In addition, the time used to detect an intrusion of the new model is much less than the conventional SVM. An additional set of experiments was conducted with the UNM dataset. Both conventional SVM and the new model performed perfectly in terms of accuracy. However, the new model still has an advantage, i.e., the running time is much less as fewer number of features are used for classification.

References

1. Bace, R.G.: *Intrusion Detection.* Macmillan Technical Publishing. (2000).
2. Burge, C.: A Tutorial on Support Vector Machines for Pattern Recognition. *Data mining and knowledge discovery journal.* **2**(2) (1998) 121–167.
3. Dasarathy, B.V.: Intrusion detection, *Information Fusion.* **4**(4) (2003) 243-245.
4. Frakes, W.B., Baeza-Yates, R., Ricardo, B.Y.: *Information Retrieval: Data Structures and Algorithms*, Prentice-Hall, 1992.
5. Han, J.C., Sanchez, R., Hu, X.H.,: Feature Selection Based on Relative Attribute Dependency: An Experimental Study. RSFDGrC'05, I, LNAI. **3641** (2005) 214-223.
6. Hu, K., Lu, Y., Shi, C.: Feature Ranking in Rough Sets. *AI Communications.* **16** (2003) 41-50.
7. Joachims, T.: *Making large-Scale SVM Learning Practical, Advances in Kernel Methods - Support Vector Learning*, MIT-Press, (1999).
8. John, G.H., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. Proc. of the 11th Int. Conf. on Machine Learning. (1994) 121-129.
9. Lee, W., Stolfo, S.J.: Data Mining Approaches for Intrusion Detection. The 7th USENIX Security Symposium. (1998) 79–94.
10. Mohajerani, M., Moeini, A., Kianie, M.: NFIDS: A Neuro-fuzzy Intrusion Detection System. Proc. of the 10th IEEE Int. Conf. on Electronics, Circuits and Systems. (2003) 348–351.
11. Pawlak, Z., Grzymala-Busse, J., Slowinski, R., Ziarko, W.: Rough Set. *Communications of the ACM.* **38**(11) (1995) 89–95
12. Qiao, Y., Xin, X.W., Bin, Y., Ge, S.: Anomaly Intrusion Detection Method Based on HMM. *Electronics Letters.* **38**(13) (2002) 663–664.
13. Vapnik, V.N.: *The Nature of Statistical Learning Theory.* Springer (1995).
14. Wang, W.D., Bridges, S.: Genetic Algorithm Optimization of Membership Functions for Mining Fuzzy Association Rules. Proc. of the 7th Int. Conf. on Fuzzy Theory & Technology. (2000) 131–134.
15. Warrender, C., Forrest, S., Pearlmutter, B.: Detecting Intrusions Using System Calls: Alternative Data Models. Proc. of the IEEE Symposium on Security and Privacy. (1999) 133–145.
16. Yao, J.T., Zhang, M.: Feature Selection with Adjustable Criteria. RSFDGrC'05, I, LNAI. **3641** (2005) 204–213.
17. Yao, J.T., Zhao, S.L., Saxton, L.V.: A study on Fuzzy Intrusion Detection. Proc. of Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security, SPIE. **5812** (2005) 23–30.