An Efficient Parzen-Window Based Network Intrusion Detector Using a Pattern Synthesis Technique

P. Viswanath[1], M. Narasimha Murty[2], and Satish Kambala[3]

[1] Dept. of CSE, IIT-Guwahati, Guwahati-781039, India
viswanath@iitg.ernet.in
[2] Dept. of CSA, IISc, Bangalore-560012, India
mnm@csa.iisc.ernet.in
[3] Dept. of CSE, NIT, Tiruchirapalli-620015, India
satish_kambala@yahoo.co.in

Abstract. The problem of detecting anomalous network connections caused by intrusion activities is called Network intrusion detection. Conventional classification methods use data from both normal and intrusion classes to build the classifiers. However, intrusion data are usually scarce and difficult to collect. Novelty detection approach overcomes this problem which depends only on normal data. For this purpose, nonparametric density estimation approaches based on Parzen-window estimators are proposed earlier. Two fundamental problems faced are, (i) due to curse of dimensionality, for high dimensional data with a limited training set, the estimation can be biased and (ii) high computational requirements. We propose, (i) a novel pattern synthesis technique to synthesize artificial new training patterns to increase the training set size and thus to reduce the curse of dimensionality effect, and (ii) a compact data representation scheme to store the entire synthetic set to reduce the computational costs. The effectiveness of our methods are experimentally demonstrated.

1 Introduction

Intrusion detection is the process of monitoring events occurring in a computer system or network and analyzing them for signs of intrusions, defined as attempts to bypass the security mechanisms of a computer or network. From a high-level view, the goal is to find out whether or not a system is operating normally. Network-based intrusion detection systems monitor network behavior by examining the content as well as the format of network data packets, which typically are not specific to the exact operating systems used by individual computers as long as these computers can communicate with each other using the same network protocol. For these types of systems, one may take a data mining approach by "mining" through the data to detect possible attacks.

Typical *classification* problems can be formulated as follows. A discriminative classifier is built using training examples from all c (≥ 2) classes, so that it can classify the given pattern into one of c classes. Many network intrusion methods

S.K. Pal et al. (Eds.): PReMI 2005, LNCS 3776, pp. 799–804, 2005.

are based on this approach [1,2]. This means, they have to use normal as well as intrusion data for training the classifier. But the main drawback in the case of intrusion detection is that there may not exist any valid underlying model from which the intrusion data is generated. So, one needs to differentiate between normal and abnormal (novel) data by using only normal data. This is called novelty detection.

Recently, Yeung and Chow [3] have proposed a novelty detection method using Parzen-Window based density estimation approach. For a given test pattern, its class conditional density for the normal class is obtained by using Gaussian kernel based Parzen-Window method. If this density exceeds a threshold density then the pattern is classified as normal, otherwise as novel. But two main drawbacks are, (i) due to curse of dimensionality, the estimated density can be biased with a limited training set, and (ii) the computational requirements are very high.

In this paper we propose (i) a pattern synthesis technique called *partition based pattern synthesis* to increase the training set size which in turn can reduce the curse of dimensionality effect, and (ii) a compact representation called *partitioned pattern count tree (PPC-tree)* to store the synthetic set which can reduce the computational requirements. Experimentally it is shown that our approach can outperform some of the related methods.

The rest of this paper is organized as follows: Section 2 gives a brief description of the density estimation approaches to novelty detection. Section 3 describes pattern synthesis. A Compact representation for the training set will be presented in Section 4. In Section 5 we will describe our method. Experimental results will be presented in Section 6. Finally, some concluding remarks will be made in Section 7.

2 Parzen-Window Based Density Estimation

Parzen introduced a nonparametric method for estimating density functions [4]. Let $p(x)$ be the density function to be approximated. Given a set $D = \{x_1, x_2, ..., x_n\}$ of n i.i.d. samples drawn according to $p(x)$, the Parzen-window estimate of $p(x)$ based on the n examples in D is

$$\hat{p}(x) = \frac{1}{n} \sum_{i=1}^{n} \delta_n(x - x_i)$$

where $\delta_n(.)$ is a kernel function with localized support and its exact form depends on n.

We choose to use Gaussian kernel functions for two reasons. First, the Gaussian function is smooth and hence the estimated density function $\hat{p}(x)$ also varies smoothly. Second, if we assume a special form of the Gaussian family in which the function is radially symmetrical, the function can be completely specified by a variance parameter only. Thus $\hat{p}(x)$ can be expressed as a mixture of radially symmetrical Gaussian kernels with common variance σ^2:

$$\hat{p}(x) = \frac{1}{n(2\pi)^{d/2}\sigma^d} \sum_{i=1}^{n} \exp\left\{-\frac{\|x - x_i\|^2}{2\sigma^2}\right\} \quad (1)$$

where d is the dimensionality of the feature space.

3 Partition Based Pattern Synthesis

We present in this section an instance based pattern synthesis method called partition based pattern synthesis where the given training set and some of the properties of the data are used to generate new artificial patterns.

Let $F = \{f_1, \ldots, f_d\}$ be the set of features, $\mathcal{X}$ be a set of patterns. For a pattern X, let X_B be the projection of X onto features in B where $B \subseteq F$. Let $\pi_B(\mathcal{X}) = \{X_B \mid X \in \mathcal{X}\}$.

Let us define a product ($\times$) similar to cartesian product over the sets of projected patterns as follows. Let $Z = (X^1_{B_1}, X^2_{B_2}, \ldots, X^p_{B_p})$ be a pattern such that $Z_{B_1} = X^1_{B_1}, Z_{B_2} = X^2_{B_2}, \ldots, Z_{B_p} = X^p_{B_p}$ where $\{B_1, B_2, \ldots, B_p\}$ is a partition of F and $X^1, \ldots, X^p$ are p patterns (not necessarily distinct). Then

$$\pi_{B_1}(\mathcal{X}) \times \pi_{B_2}(\mathcal{X}) \times \ldots \times \pi_{B_p}(\mathcal{X}) = \{(X^1_{B_1}, X^2_{B_2}, \ldots, X^p_{B_p}) \mid X^1, \ldots, X^p \in \mathcal{X}\}.$$

The synthetic set of patterns for a class with label l is,

$$\mathrm{SS}_{Q_l}(\mathcal{X}^l) = \pi_{B_{l1}}(\mathcal{X}^l) \times \pi_{B_{l2}}(\mathcal{X}^l) \times \ldots \times \pi_{B_{lp}}(\mathcal{X}^l)$$

where $\mathcal{X}^l$ is the set of given pattens which belongs to the class with label l and $Q_l = \{B_{l1}, B_{l2}, \ldots, B_{lp}\}$ is a partition of F.

Example: This illustrates the concept of partition based pattern synthesis. Let $F = (f_1, f_2, f_3, f_4)$, $\mathcal{X}^l = \{(a, b, c, d), (p, q, r, s), (w, x, y, z)\}$ be the given training set for class with label l, and $\pi_l = \{B_{l1}, B_{l2}\}$ such that $B_{l1} = (f_1, f_2)$ and $B_{l2} = (f_3, f_4)$. Then, the synthetic set for the class is $SS_{Q_l}(\mathcal{X}^l) = \pi_{B_{l1}}(\mathcal{X}^l) \times \pi_{B_{l2}}(\mathcal{X}^l)$ $= \{(X^a_{B_{l1}}, X^b_{B_{l2}}) \mid X^a, X^b \in \mathcal{X}^i\} = \{(a, b, c, d),\ (a, b, r, s),\ (a, b, y, z),\ (p, q, c, d),$ $(p, q, r, s),\ (p, q, y, z),\ (w, x, c, d),\ (w, x, r, s),\ (w, x, y, z)\ \}$.

It is easy to see that the original set of patterns $\mathcal{X}^l$ and the synthetic set $SS_{Q_l}(\mathcal{X}^l)$ are from the same probability distribution if the subsets of F *viz.,* $B_{l1}, \ldots, B_{lp}$ are statistically independent for the given class. But unfortunately, such a partitioning of F may not exist and even if they exist, finding it would be a computationally demanding problem. So, an approximate partitioning method is used based on correlation coefficient between pairs of features. The set of features is partitioned such that on average features in different blocks are least correlated with each other.

4 A Compact Representation

Partition based pattern synthesis can generate synthetic set of size $O(n^p)$, where n is the original set size and p is the number of blocks in the partition. Hence

explicitly storing the synthetic set is very space consuming. In this section we present a compact representation of the original set which is suitable for the synthesis. For large data sets, this representation requires less storage space than that for the original set. This is called *partitioned pattern count tree (PPC-tree).*

PPC-tree is a generalization of a data structure called *pattern count tree (PC-tree)* [5]. PC-tree, as well as PPC-tree are suitable when each feature can take discrete values (can be categorical values also). For continuous valued features, an appropriate discretization needs to be done first. PPC-tree structure is described with an example below.

Example: Let $\{(a,b,c,x,y,z),\ (a,b,d,x,y,z),\ (a,e,c,x,y,u),\ (f,b,c,x,y,v)\}$ be the original training set for a class with label i. Let the partition chosen is $Q_i = \{B_{i1}, B_{i2}\}$ such that $B_{i1} = \{f_1, f_2, f_3\}$ and $B_{i2} = \{f_4, f_5, f_6\}$, respectively. Then the PPC-tree is shown in Fig. 1. Each node of the tree (except root) is of the format (*feature : count*).

The PPC-tree is a set of tree structures (called PC-trees) $\{\mathcal{T}_{i1}, \mathcal{T}_{i2}\}$, where $\mathcal{T}_{i1}$ represents the set of projected patterns $\mathcal{X}^i_{B_{i1}} = \{(a,b,c),\ (a,b,d),\ (a,e,c),\ (f,b,c)\}$. Similarly $\mathcal{T}_{i2}$ represents the set of projected patterns $\mathcal{X}^i_{B_2}$.

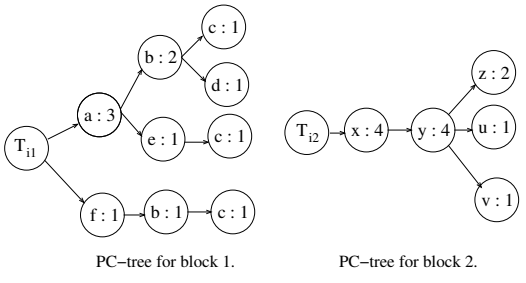

Fig. 1. PPC-tree $\mathcal{T}_i = \{\mathcal{T}_{i1}, \mathcal{T}_{i2}\}$

A path from root to leaf of $\mathcal{T}_{i1}$ represents a projected pattern onto features in B_{i1} and that of $\mathcal{T}_{i2}$ represents a projected pattern onto B_{i2}. Merging the two projected patterns gives a synthetic pattern according to the partition.

The space and time (to build) requirements for the representation is $O(n)$ where n is the number of original patterns.

5 Novelty Detection Using Synthetic Patterns

For the normal class, partition the set of features into p blocks (p is chosen based on a three-fold cross-validation) using the approximate partitioning method. Let the partition be $\{B_1, \ldots, B_p\}$. Let the PPC-tree representation for the synthetic set be $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_p\}$. Let T be the given test pattern. For each tree structure $\mathcal{T}_i \in \mathcal{T}$, we maintain two lists $dist_i$ and $dist\_count_i$ each of length equal to the number of paths (same as the number of leaf nodes i.e., l_i) in the tree structure

$\mathcal{T}_i$. The j^{th} entry in the list $dist_i$ is $dist_{ij}$ and this gives squared Euclidean distance between T_{B_i} and a projected pattern corresponding to a path in the tree, whereas the corresponding entry in the $dist\_count_i$ list is $dist\_count_{ij}$ which gives the number of times the projected pattern (given by the count field of the leaf node of the corresponding path) is repeated. Find the set of pairs $\mathcal{S} = \{(\text{dist}_{1j_1} + \ldots + \text{dist}_{pj_p},\ \text{dist\_count}_{1j_1} \times \ldots \times \text{dist\_count}_{pj_p}) \mid 1 \leq j_i \leq l_i,\ 1 \leq i \leq p\}$, where first member of each pair gives squared Euclidean distance between T and a synthetic pattern and second member gives number of times the synthetic pattern is repeated. These distances are used to estimate the density at the given test pattern T by using the Equation 1 whose modified version is given below.

1. Let $\mathcal{S} = \{(d_1, c_1), (d_2, c_2), \ldots, (d_m, c_m)\}$ where $m = \mid \mathcal{S} \mid = (l_1 \times l_2 \times \ldots \times l_p)$.
2. Density at T is found by using

$$\hat{p}(x) = \frac{1}{n^p (2\pi)^{d/2} \sigma^d} \sum_{i=1}^{m} c_i \exp\left\{-\frac{d_i}{2\sigma^2}\right\} \tag{2}$$

{Hence, a total of $m = l_1 \times l_2 \times \ldots \times l_p$ computations are required. But according to the experimental studies, the values of each of l_i is much smaller than n. So m is much smaller than n^p, the total number of synthetic patterns.} If the calculated density, $p_m(x)$ exceeds a threshold value (chosen based on a three-fold cross-validation) then the test pattern T is classified as normal, otherwise as novel (abnormal).

6 Experiments

In our experiments, we use KDD Cup 1999 dataset (for details refer [3]) which was obtained from a real-world military network environment with simulated intrusions.

The dataset has four intrusion categories: *probing*, *denial-of-service* (DoS), *user-to-root* (U2R), and *remote-to-local* (R2L). We use the performance measures *true acceptance rate* (TAR) and *true detection rate* (TDR) as given in [3]. TAR measures the percentage of normal connections in the test set that are classified as normal, whereas TDR measures the percentage of intrusive connections in the test set that are detected as intrusions.

We use the training dataset consisting of 97276 patterns, all from normal class to estimate the density model. The test dataset consists of 311029 patterns. Each test pattern is classified as either normal or abnormal based on a threshold value which is fixed based on a three-fold cross-validation using training dataset.

We have compared our results with those of Yeung and Chow [3], KDD Cup winner and KDD Cup Runner. Table 1 gives the TAR and TDR values for all the methods. The results of Yeung and Chow are taken from [3]. Since the KDD Cup is concerned with multi-class classification but we are interested only in normal/intrusion discrimination, we have converted the results of KDD Cup winner and Runner into our format (as done by Yeung and Chow [3]). The best results in each category are highlighted.

Table 1. Comparison of our model with others (showing (%))

Method	TAR	TDR				Cost
	Normal	Probing	DoS	U2R	R2L	
Ours	96.20	**99.76**	97.59	**96.49**	**37.12**	**0.1796**
Yeung and Chow	97.38	99.17	96.71	93.57	31.17	0.2024
KDD Winner	**99.45**	87.73	**97.69**	26.32	10.27	0.2260
KDD Runner	99.42	89.01	97.57	22.37	7.38	0.2339

The proposed method is also compared with others based on a scoring scheme. KDD Cup scoring scheme uses a cost matrix which is modified (as done by Yeung and Chow [3]) to suit the given approach.

Although our method is not always better because its TAR is lower, it is fair to say that our method can achieve performance comparable to the best methods, with the favorable characteristics that it requires no intrusion data at all and effectively uses very large number of synthesized training patterns.

7 Concluding Remarks

The paper presented a novelty detection approach to detect abnormal patterns. This is useful when the normal class is well defined and abnormal class does not have an explicit definition apart from saying that it is not normal. Parzen-Window based novelty detection is presented. To reduce the curse of dimensionality effect we used partition based synthetic patterns. To reduce the computational requirements we used PPC-tree representations. The results are compared with other related approaches and found to be encouraging.

References

1. Lippmann, R., Cunningham, R.: Improving intrusion detection performance using keyword selection and neural networks. Computer Networks **34** (2000) 579–603
2. Lee, W., Xiang, D.: Information-theoretic measures for anomaly detection. In: Proceedings of the IEEE Symposium on Security and Privacy. (2001) 130–143
3. Yeung, D.Y., Chow, C.: Parzen-window network intrusion detectors. In: Proceedings of the 16th International Conference on Pattern Recognition. Volume 4. (2002) 385–388
4. Parzen, E.: On estimation of a probability density function and mode. Annals of Mathematical Statistics **33** (1962) 1065–1076
5. Ananthanarayana, V., Murty, M., Subramanian, D.: An incremental data mining algorithm for compact realization of prototypes. Pattern Recognition **34** (2001) 2249–2251