Expert Systems with Applications 38 (2011) 306–313

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A novel intrusion detection system based on hierarchical clustering and support vector machines

Shi-Jinn Horng [a,b,*], Ming-Yang Su [c], Yuan-Hsin Chen [b], Tzong-Wann Kao [d], Rong-Jian Chen [b], Jui-Lin Lai [b], Citra Dwi Perkasa [a]

[a] *Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, 43, Sec/4. Kee-Lung Road, 106 Taipei, Taiwan*
[b] *Department of Electronic Engineering, National United University, Miaoli, Taiwan*
[c] *Department of Computer Science and Information Engineering, Ming Chuan University, Taoyuan, Taiwan*
[d] *Department of Electronic Engineering, Northern Taiwan Institute of Science and Technology, Taipei, Taiwan*

ARTICLE INFO

Keywords:
Network intrusion detection system (NIDS)
Support vector machines (SVMs)
Hierarchical clustering algorithm
KDD Cup 1999
Network security
Data mining

ABSTRACT

This study proposed an SVM-based intrusion detection system, which combines a hierarchical clustering algorithm, a simple feature selection procedure, and the SVM technique. The hierarchical clustering algorithm provided the SVM with fewer, abstracted, and higher-qualified training instances that are derived from the KDD Cup 1999 training set. It was able to greatly shorten the training time, but also improve the performance of resultant SVM. The simple feature selection procedure was applied to eliminate unimportant features from the training set so the obtained SVM model could classify the network traffic data more accurately. The famous KDD Cup 1999 dataset was used to evaluate the proposed system. Compared with other intrusion detection systems that are based on the same dataset, this system showed better performance in the detection of DoS and Probe attacks, and the beset performance in overall accuracy.

1. Introduction

Due to the advancement of the Internet technology, numerous e-commerce transactions are completed online. However, given the weakness of servers (Vigna, Robertson, Kher, & Kemmerer, 2003; Yu, Tsai, & Weigert, 2007), hackers often intrude upon these Internet transactions, and use DoS/DDoS attacks to prevent these servers from providing services. Network intrusion detection system (NIDS), as an important link in the network security infrastructures, aims to detect malicious activities, such as denial of service attacks, port scans, or even attempts to crack into computers by monitoring network traffic. In addition to inspecting incoming network traffic, NIDS can also obtain valuable information on an ongoing intrusion from outgoing or local traffic. NIDS is often not a standalone system, but works with other systems as a firewall. If necessary, it can update the blacklist of the firewall in real-time to block the suspicious connections.

A common problem of NIDS is that it specifically detects known service or network attacks only, which is called misuse detection, by using pattern matching approaches. On the other hand, an anomaly detection system detects attacks by building profiles of normal behaviors first, and then identifies potential attacks when their behaviors are significantly deviated from the normal profiles. Many methods have been proposed in the past few years on the design of effective NIDSs, among which, decision tree has been proven to have good performance. Bagged boosting (Pfahringer, 2000), based on C5 decision trees is a method used by the KDD Cup 1999 winner. In the KDD Cup (1999), the approaches of the top three winners were based on the tree methods with some variations. The runner-up entry used a variant of decision tree proposed by Kernel Miner (Levin, 2000), which is a data mining tool for building the optimal decision forest.

Some NIDSs are based on the fuzzy set theory. For example, fuzzy rough C-means (FRCM), proposed by Chimphlee, Abdullah, Md Sap, Srinoy, and Chimphlee (2006), utilized the advantage of fuzzy set theory and rough set theory for network intrusion detection. Another fuzzy approach, proposed by Toosi and Kahani (2007), combined the neuro-fuzzy network, fuzzy inference approach, and genetic algorithms to design their NIDS, and was evaluated by the KDD Cup 1999 dataset. Novikov, Yampolskiy, and Reznik (2006), the combination of a radial basis function (RBF) neural network and a multi layer perception (MLP) neural network was used to design an intrusion detection system. Sabhnani and Serpen (2003) analyzed the performances of a comprehensive set of pattern recognitions and machine learning algorithms. Their system

* Corresponding author at: Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, 43, Sec/4. Kee-Lung Road, 106 Taipei, Taiwan. Tel.: +886 2 27376700; fax: +886 2 27301081.

E-mail addresses: horngsj@yahoo.com.tw (S.-J. Horng), minysu@mail.mcu.edu.tw (M.-Y. Su), yschen@nuu.edu.tw (Y.-H. Chen), tkao@tsint.edu.tw (T.-W. Kao), rjchen@nuu.edu.tw (R.-J. Chen), jllai@nuu.edu.tw (J.-L. Lai), liemcici@gmail.com (C.D. Perkasa).

0957-4174/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2010.06.066

in Sabhnani and Serpen (2003), which outperformed the KDD Cup 1999 winner's system, combined several classifiers, one designated for one type of attacks in the KDD Cup 1999 dataset. Xuren, Famei, and Rongsheng (2006), an association rule discovering system was applied in a rough set theory framework to design an NIDS.

Many researches have applied data mining techniques in the design of NIDS. One of the promising techniques is support vector machine (SVM), which solid mathematical foundations (Khan, Awad, & Thuraisingham, 2007; Yu, Yang, Han, & Li, 2003) have provided satisfying results. SVM separates data into multiple classes (at least two) by a hyperplane, and simultaneously minimizes the empirical classification error and maximizes the geometric margin. Thus, it is also known as maximum margin classifiers.

Although SVMs have shown good results in data classification, they are not favorable for large-scale dataset because the training complexity is very dependent on the amount of data in the training set. Larger amount of data would lead to higher training complexity. However, many data mining applications involve millions or even billions of pieces of data records. For example, in the KDD Cup 1999 dataset, there are more than 4 million and 3 million instances in the training set and test set, respectively. The SVM technique is unable to operate at such a large dataset due to system failures caused by insufficient memory, or may take too long to finish the training. Since this study used the KDD Cup 1999 dataset, to reduce the amount of data, a hierarchical clustering method was applied to preprocess the dataset before SVM training. The clustering method could produce high quality dataset with far less instances that sufficiently represent all of the instances in the original dataset.

This study proposed an SVM-based intrusion detection system based on a hierarchical clustering algorithm to preprocess the KDD Cup 1999 dataset before SVM training. The hierarchical clustering algorithm was used to provide a high quality, abstracted, and reduced dataset for the SVM training, instead of the originally enormous dataset. Thus, the system could greatly shorten the training time, and also achieve better detection performance in the resultant SVM classifier. The remainder of this paper is organized as follows. Section 2 provides an overview of the hierarchical clustering algorithm proposed by Zhang, Ramakrishnan, and Livny (1996) and the SVMs. Section 3 describes the proposed system. Section 4 presents experimental results to demonstrate that the performance of the proposed system is better than others. Finally concluding remarks are given in Section 5.

2. Background

This section will give a detailed description about the balanced iterative reducing and clustering using hierarchies (BIRCH) hierarchical clustering algorithm that is used to produce fewer significant instances from a very large dataset. With fewer significant instances, the support vector machines (SVMs) can achieve shorter training time and better classification performance. A brief introduction of the SVM is also given in this section.

2.1. BIRCH hierarchical clustering algorithm

The BIRCH hierarchical clustering algorithm applied in this system was originally proposed by Zhang et al. (1996). The concept of BIRCH is different from other clustering algorithms, such as CURE (Guha, Rastogi, & Shim, 1998), ROCK (Guha, Rastogi, & Shim, 1999), and Chameleon (Karypis, Han, & Kumar, 1999), because it stores fewer abstracted data points than the whole dataset. Each abstracted point represents the centroid of a cluster of data points. Compared to CURE, ROCK, and Chameleon, the BIRCH clustering algorithm can achieve high quality clustering with lower processing cost. The advantages of BIRCH are as follows:

- Constructs a tree, called a clustering feature (CF) tree, by only one scan of dataset using an incremental clustering technique.
- Able to handle noise effectively.
- Memory-efficient because BIRCH only stores a few abstracted data points instead of the whole dataset.

2.1.1. Clustering feature (CF)

The concept of a clustering feature (CF) tree is at the core of BIRCH's incremental clustering algorithm. Nodes in the CF tree are composed of clustering features. A CF is a triplet, which summarizes the information of a cluster.

Definition 1 (*Clustering feature (Zhang et al., 1996)*). Given n d-dimensional data points in a cluster $\{x_i\}$, where $i = 1, 2, \ldots, n$, the clustering feature (CF) of the cluster is a 3-tuple, denoted as CF = (n, LS, SS), where n is the number of data points in the cluster, LS is the linear sum of the data points, i.e., $\sum_{i=1}^{n} x_i$, and SS is the square sum of the data points, i.e., $\sum_{i=1}^{n} x_i^2$.

Theorem 1 (CF addition theorem (Zhang et al., 1996)). *Assume that $CF_1 = (n_1, LS_1, SS_1)$ and $CF_2 = (n_2, LS_2, SS_2)$ are the CFs of two disjoint clusters. Then the CF of the new cluster, as formed by merging the two disjoint clusters is*

$$\mathrm{CF}_1 + \mathrm{CF}_2 = (n_1 + n_2, LS_1 + LS_2, SS_1 + SS_2) \tag{3}$$

For example, suppose there are three points (2, 3), (4, 5), (5, 6) in cluster C1, then the CF of C1 is

$$\begin{aligned}\mathrm{CF}_1 &= \left\{3, (2+4+5, 3+5+6), \left(2^2+4^2+5^2, 3^2+5^2+6^2\right)\right\}\\ &= \{3, (11, 14), (45, 70)\}.\end{aligned}$$

Suppose that there is another cluster C2 with CF_2 = {4, (40, 42), (100, 101)}. Then the CF of the new cluster formed by merging cluster C1 and C2 is

$$\begin{aligned}\mathrm{CF}_3 &= \{3+4, (11+40, 14+42), (45+100, 70+101)\}\\ &= \{7, (51, 56), (145, 171)\}.\end{aligned}$$

By Definition 1 and Theorem 1, the CFs of clusters can be stored and calculated incrementally and accurately as clusters are merged. Based on the information stored in CF, the centroid C and radius R of a cluster can be easily computed. The definitions of C and R of a cluster are given as follows. Given n d-dimensional data points, say $\{x_i\}$ and i = 1, 2, ..., n, in a cluster:

$$\text{the centroid}\quad C = \frac{\sum_{i=1}^{n} x_i}{n}, \quad \text{and} \tag{1}$$

$$\text{the radius}\quad R = \frac{\sum_{i=1}^{n} \|x_i - C\|^2}{n}. \tag{2}$$

where, R denotes the average distance of all member points to the centroid. As mentioned earlier, CF stores only the abstracted data point, i.e., statistically summary of data points that belong to the same cluster. After a data point is added into a cluster, the detail information of the data point itself is missing. Therefore, this approach can save space significantly for densely packed data points, especially when the size of the dataset is large.

2.1.2. CF tree

A CF tree is a height-balanced tree with two parameters, branching factor B and radius threshold T. Each non-leaf node in a CF tree contains the most B entries of the form (CF_i, $child_i$), where $1 \leqslant i \leqslant B$ and $child_i$ is a pointer to its ith child node, and CF_i is the CF