Expert Systems with Applications 36 (2009) 11994–12000

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Review

Intrusion detection by machine learning: A review

Chih-Fong Tsai [a], Yu-Feng Hsu [b], Chia-Ying Lin [c], Wei-Yang Lin [d,*]

[a] *Department of Information Management, National Central University, Taiwan*
[b] *Department of Information Management, National Sun Yat-Sen University, Taiwan*
[c] *Department of Accounting and Information Technology, National Chung Cheng University, Taiwan*
[d] *Department of Computer Science and Information Engineering, National Chung Cheng University, Taiwan*

ARTICLE INFO

Keywords:
Intrusion detection
Machine learning
Hybrid classifiers
Ensemble classifiers

ABSTRACT

The popularity of using Internet contains some risks of network attacks. Intrusion detection is one major research problem in network security, whose aim is to identify unusual access or attacks to secure internal networks. In literature, intrusion detection systems have been approached by various machine learning techniques. However, there is no a review paper to examine and understand the current status of using machine learning techniques to solve the intrusion detection problems. This chapter reviews 55 related studies in the period between 2000 and 2007 focusing on developing single, hybrid, and ensemble classifiers. Related studies are compared by their classifier design, datasets used, and other experimental setups. Current achievements and limitations in developing intrusion detection systems by machine learning are present and discussed. A number of future research directions are also provided.

1. Introduction

The Internet has become a part of daily life and an essential tool today. It aids people in many areas, such as business, entertainment and education, etc. In particular, Internet has been used as an important component of business models (Shon & Moon, 2007). For the business operation, both business and customers apply the Internet application such as website and e-mail on business activities. Therefore, information security of using Internet as the media needs to be carefully concerned. Intrusion detection is one major research problem for business and personal networks.

As there are many risks of network attacks under the Internet environment, there are various systems designed to block the Internet-based attacks. Particularly, intrusion detection systems (IDSs) aid the network to resist external attacks. That is, the goal of IDSs is to provide a wall of defense to confront the attacks of computer systems on Internet. IDSs can be used on detect difference types of malicious network communications and computer systems usage, whereas the conventional firewall can not perform this task. Intrusion detection is based on the assumption that the behavior of intruders different from a legal user (Stallings, 2006).

In general, IDSs can be divided into two categories: anomaly and misuse (signature) detection based on their detection approaches (Anderson, 1995; Rhodes, Mahaffey, & Cannady, 2000). Anomaly detection tries to determine whether deviation from the established normal usage patterns can be flagged as intrusions. On the other hand, misuse detection uses patterns of well-known attacks or weak spots of the system to identify intrusions.

In literature, numbers of anomaly detection systems are developed based on many different machine learning techniques (c.f. Section 3). For example, some studies apply single learning techniques, such as neural networks, genetic algorithms, support vector machines, etc. On the other hand, some systems are based on combining different learning techniques, such as hybrid or ensemble techniques. In particular, these techniques are developed as classifiers, which are used to classify or recognize whether the incoming Internet access is the normal access or an attack. However, there is no a review of these different machine learning techniques over the intrusion detection domain.

Therefore, the goal of this paper is to review 55 related studies/systems published from 2000 to 2007 by examining what techniques have been used, what experiments have been conducted, and what should be considered for future work based on the machine learning's perspective.

This paper is organized as follows. Section 2 provides an overview of machine learning techniques and briefly describes a number of related techniques for intrusion detection. Section 3 compares related work based on the types of classifier design, the chosen baselines, datasets used for experiments, etc. Conclusion and discussion for future research are given in Section 4.

* Corresponding author. Tel.: +886 5 2720411; fax: +886 5 2720859.
E-mail address: wylin@cs.ccu.edu.tw (W.-Y. Lin).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.05.029

2. Machine learning techniques

2.1. Pattern classification

Pattern recognition is the action to take raw data and activity on data category (Michalski, Bratko, & Kubat, 1998). The methods of supervised and unsupervised learning can be used to solve different pattern recognition problems (Theodoridis & Koutroumbas, 2006, 2006). In supervised learning, it is based on using the training data to create a function, in which each of the training data contains a pair of the input vector and output (i.e. the class label).

The learning (training) task is to compute the approximate distance between the input–output examples to create a classifier (model). When the model is created, it can classify unknown examples into a learned class labels.

2.2. Single classifiers

The intrusion detection problem can be approached by using one single machine learning algorithm. In literature, machine learning techniques (e.g. *k*-nearest neighbor, support vector machines, artificial neural network, decision trees, self-organizing maps, etc.) have been used to solve these problems.

2.2.1. K-nearest neighbor

K-nearest neighbor (*k*-NN) is one of the most simple and traditional nonparametric technique to classify samples (Bishop, 1995; Manocha & Girolami, 2007). It computes the approximate distances between different points on the input vectors, and then assigns the unlabeled point to the class of its *K*-nearest neighbors. In the process of create *k*-NN classifier, *k* is an important parameter and different *k* values will cause different performances. If *k* is considerably huge, the neighbors which used for prediction will make large classification time and influence the accuracy of prediction.

k-NN is called instance based learning, and it is different from the inductive learning approach (Mitchell, 1997). Thus, it does not contain the model training stage, but only searches the examples of input vectors and classifies new instances. Therefore, *k*-NN "on-line" trains the examples and finds out *k*-nearest neighbor of the new instance.

2.2.2. Support vector machines

Support vector machines (SVM) is proposed by Vapnik (1998). SVM first maps the input vector into a higher dimensional feature space and then obtain the optimal separating hyper-plane in the higher dimensional feature space. Moreover, a decision boundary, i.e. the separating hyper-plane, is determined by support vectors rather than the whole training samples and thus is extremely robust to outliers.

In particular, an SVM classifier is designed for binary classification. That is, to separate a set of training vectors which belong to two different classes. Note that the support vectors are the training samples close to a decision boundary. The SVM also provides a user specified parameter called penalty factor. It allows users to make a tradeoff between the number of misclassified samples and the width of a decision boundary.

2.2.3. Artificial neural networks

The neural network is information processing units which to mimic the neurons of human brain (Haykin, 1999). Multilayer perceptron (MLP) is the widely used neural network architecture in many pattern recognition problems. A MLP network consists of an input layer including a set of sensory nodes as input nodes, one or more hidden layers of computation nodes, and an output layer of computation nodes. Each interconnection has associated with it a scalar weight which is adjusted during the training phase.

In addition, the backpropagation learning algorithm is usually used to train a MLP, which are also called as backpropagation neural networks. First of all, random weights are given at the beginning of training. Then, the algorithm performs weights tuning to define whatever hidden unit representation is most effective at minimizing the error of misclassification.

2.2.4. Self-organizing maps

Self-organizing map (SOM) (Kohonen, 1982) is trained by an unsupervised competitive learning algorithm, a process of self organization. The aim of SOM is to reduce the dimension of data visualization. That is, SOM projects and clusters high-dimensional input vectors onto a low-dimensional visualized map, usually 2 for visualization. It usually consists of an input layer and the Kohonen layer which is designed as two-dimensional arrangement of neurons that maps *n* dimensional input to two dimensions. Kohonen's SOM associates each of the input vectors to a representative output. The network finds the node closest to each training case and moves the winning node, which is the closest neuron (i.e. the neuron with minimum distance) to the training case.

That is, SOM maps similar input vectors onto the same or similar output units on such a two-dimensional map. Therefore, output units will self-organize to an ordered map and those output units with similar weights are also placed nearby after training.

2.2.5. Decision trees

A decision tree classifies a sample through a sequence of decisions, in which the current decision helps to make the subsequent decision. Such a sequence of decisions is represented in a tree structure. The classification of a sample proceeds from the root node to a suitable end leaf node, where each end leaf node represents a classification category. The attributes of the samples are assigned to each node, and the value of each branch is corresponding to the attributes (Mitchell, 1997).

A well-known program for constructing decision trees is CART (Classification and Regressing Tree) (Breiman, Friedman, Olshen, & Stone, 1984). A decision tree with a range of discrete (symbolic) class labels is called a classification tree, whereas a decision tree with a range of continuous (numeric) values is called a regression tree.

2.2.6. Naïve bayes networks

There are many cases where we know the statistical dependencies or the causal relationships between system variables. However, it might be difficult to precisely express the probabilistic relationships among these variables. In other words, the prior knowledge about the system is simply that some variable might influence others. To exploit this structural relationship or casual dependencies between the random variables of a problem, one can use a probabilistic graph model called Naïve Baysian Networks (NB).

The model provides an answer to questions like "What is the probability that it is a certain type of attack, given some observed system events?" by using conditional probability formula. The structure of a NB is typically represented by a directed acyclic graph (DAG), where each node represents one of system variables and each link encodes the influence of one node upon another (Pearl, 1988). Thus, if there is a link from node *A* to node *B*, *A* directly influences *B*.

2.2.7. Genetic algorithms

Genetic algorithms (GA) use the computer to implement the natural selection and evolution (Koza, 1992). This concept comes from the "adaptive survival in natural organisms". The algorithm