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

starts by randomly generating a large population of candidate programs. Some type of fitness measure to evaluate the performance of each individual in a population is used. A large number of iterations is then performed that low performing programs are replaced by genetic recombinations of high performing programs. That is, a program with a low fitness measure is deleted and does not survive for the next computer iteration.

2.2.8. Fuzzy logic

Fuzzy logic (or fuzzy set theory) is based on the concept of the fuzzy phenomenon to occur frequently in real world. Fuzzy set theory considers the set membership values for reasoning and the values range between 0 and 1. That is, in fuzzy logic the degree of truth of a statement can range between 0 and 1 and it is not constrained to the two truth values (i.e. true, false). For examples, "rain" is a commonly natural phenomenon, and it may have very fierce change. Raining may be able to convert the circumstances from slight to violent (Zimmermann, 2001).

2.3. Hybrid classifiers

In the development of an IDS, the ultimate goal is to achieve the best possible accuracy for the task at hand. This objective naturally leads to the design of hybrid approaches for the problem to be solved. The idea behind a hybrid classifier is to combine several machine learning techniques so that the system performance can be significantly improved. More specifically, a hybrid approach typically consists of two functional components. The first one takes raw data as input and generates intermediate results. The second one will then take the intermediate results as the input and produce the final results (Jang, Sun, & Mizutani, 1996).

In particular, hybrid classifiers can be based on cascading different classifiers, such as neuro-fuzzy techniques. On the other hand, hybrid classifiers can use some clustering-based approach to preprocess the input samples in order to eliminate unrepresentative training examples from each class. Then, the clustering results are used as training examples for classifier design. Therefore, the first level of hybrid classifiers can be based on either supervised or unsupervised learning techniques.

Finally, hybrid classifiers can also be based on the integration of two different techniques in which the first one aims at optimizing the learning performance (i.e. parameter tuning) of the second model for prediction.

2.4. Ensemble classifiers

Ensemble classifiers were proposed to improve the classification performance of a single classifier (Kittler, Hatef, Duin, & Matas, 1998). The term "ensemble" refers to the combination of multiple weak learning algorithms or weak learners. The weak learners are trained on different training samples so that the overall performance can be effectively improved.

Among the strategies for combining weak learners, the "majority vote" is arguably the most commonly used one in the literature. Other combination methods, such as boosting and bagging, are based on training data resampling and then taking a majority vote of the resulting weak learners.

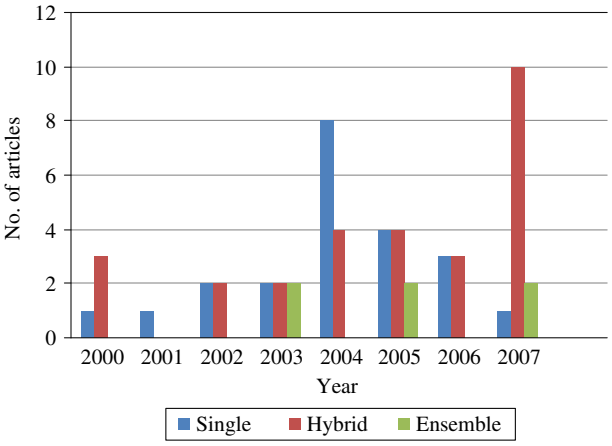

Fig. 1. Yearwise distribution of articles for the types of classifier design.

3. Comparisons of related work

3.1. Types of classifier design

The methods for intrusion detection can be generally divided into three categories, namely single, hybrid, and ensemble. To understand the types of classifier design, Table 1 shows the total numbers of the 55 articles using single, ensemble, and hybrid classifiers respectively. Fig. 1 presents yearwise distribution of these articles in terms of their classifier design.

Regarding Table 1, single classifiers have the largest number of literatures between 2000 and 2007. On the other hand, very few studies consider ensemble classifiers although they could outperform single classifiers in terms of classification accuracy.

Fig. 1 depicts the number of papers within each year. The number of papers using single methods yields a peak in 2004 and decreases gradually after that. Due to the recent developments in intrusion detection, it is now very difficult to design a single approach which outperforms the existing ones. On th other hand, the hybrid approaches have moved from marginalization to mainstream in the recent years. A strong supporting evidence is that there are 10 research publications based on hybrid approaches in

Table 1
Total numbers of articles for the types of classifier design.

	Single	Hybrid	Ensemble
No. of articles	26 Balajinath and Raghavan (2000), Bouzida et al. (2004), Chen et al. (2005), Chimphlee et al. (2006), Depren et al. (2005), Eskin et al. (2002), Fan et al. (2004), Heller et al. (2003), Li and Guo (2007), Liao and Vemuri (2002), Mukkamala et al. (2004), Peddabachigari et al. (2004), Ramos and Abraham (2005), Schultz et al. (2001), Scott (2004), Shyu et al. (2003), Tian et al. (2004), Wang and Stolfo (2004), Wang and Battiti (2006), Wang et al. (2004), Wang et al. (2006), Zhang and Shen (2005)	23 Abadeh et al. (2007), Bridges and Vaughn (2000), Chavan et al. (2004), Chen et al. (2007), Depren et al. (2005), Eskin et al. (2002), Florez et al. (2002), Giacinto and Roli (2003), Jiang et al. (2006), Joo et al. (2003), Kayacik et al. (2007), Khan et al. (2007), Lee and Stolfo (1998, 2000), Liu and Yi (2006); Liu et al. (2007, 2004), Luo and Bridgest (2000), Moradi and Zulkernine (2004), Ozyer et al. (2007), Peddabachigari et al. (2007), Shon et al. (2006), Shon and Moon (2007); Stein et al. (2005), Toosi and Kahani (2007), Tsang et al. (2007), Xiang and Lim (2005), Zhang et al. (2005), Zhang et al. (2004)	6 Abadeh et al. (2007), Giacinto et al. (2006, 2008), Giacinto and Roli (2003), Han and Cho (2003), Kang et al. (2005), Mukkamala et al. (2005); Peddabachigari et al. (2007)

Table 2
Total numbers of articles for single classifiers.

	Fuzzy logic	*K*-NN	SVM	NB	MLP	DT	SOM	GA
No. of articles	1 Chimphlee et al. (2006)	6 Bouzida et al. (2004), Eskin et al. (2002), Li and Guo (2007), Liao and Vemuri (2002), Wang and Stolfo (2004), Wang et al. (2004)	7 Chen et al. (2005), Eskin et al. (2002), Heller et al. (2003), Peddabachigari et al. (2004), Tian et al. (2004), Wang and Battiti (2006), Zhang and Shen (2005)	3 Schultz et al. (2001), Scott (2004), Wang et al. (2006)	2 Chen et al. (2005), Shyu et al. (2003)	4 Bouzida et al. (2004), Depren et al. (2005), Fan et al. (2004), Peddabachigari et al. (2004)	1 Ramos and Abraham (2005)	2 Balajinath and Raghavan (2000), Mukkamala et al. (2004)

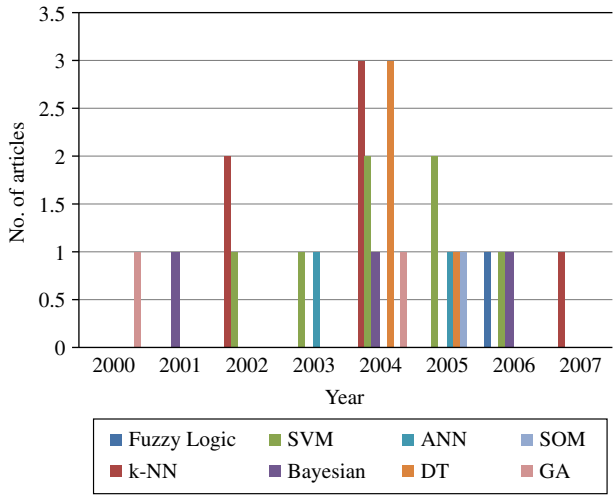

Fig. 2. Yearwise distribution of articles for single classifiers.

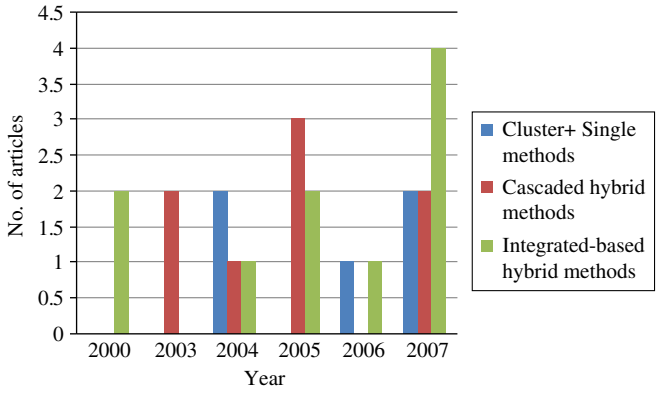

Fig. 3. Yearwise distribution of articles for hybrid classifiers.

2007. Note that these is only one paper using single method in the same year. Without a doubt, the hybrid approaches provide better flexibility and thus are gaining more popularity in the recent year.

3.2. Single classifiers

For the work based on designing single classifiers, Table 2 shows the total numbers of articles using different classification techniques, e.g. SVM, MLP, etc. Similar to Figs. 1 and 2 presents yearwise distribution of these articles in terms of their developed classifiers.

K-NN and SVM are the most commonly used techniques of single approach on intrusion detection. This result implies (although the number of comparative samples is not large) that SVM is getting more considered for single classifier design. On the other hand, fuzzy logic and SOM do not largely consider in intrusion detection.

3.3. Hybrid classifiers

As there are three strategies to design hybrid classifiers, Table 3 shows the total numbers of articles based on the three types of hybrid classifiers respectively. Fig. 3 presents yearwise distribution of these articles in terms of their hybrid classifier design.

Table 4
Yearwise distribution of articles for baseline classifiers.

	'07	'06	'05	'04	'03	'02	'01	'00	Total
k-Means		2	1						3
SVM	7	3	5	4		2			21
MLP	2	2	3	4	2				13
K-NN	2	1	2		1	3			9
LR[a]		1							1
SOM	1		2						3
DT	4		5	1				2	12
GA	1			1				1	3
Bayesian	1		1						2

[a] LR: Logistic Regression.

Table 5
Yearwise distribution of datasets used.

	'07	'06	'05	'04	'03	'02	'01	'00	Total
KDD99	7	4	5	7	3	2	1	1	30
DARPA1998	5	3	4	2		2	1	1	18
DARPA1999	1			1		1			3
UNM			1						1
SSCNNJU				1					1
CUCS				1					1
RWND					1				1
PACCT					1				1
Windows system					1				1
network tcpdump data								1	1

Table 3
Total numbers of articles for hybrid classifiers.

	Cluster + Single methods	Cascaded hybrid methods	Integrated-based hybrid methods
No. of articles	5 Chavan et al. (2004), Khan et al. (2007), Liu and Yi (2006), Liu et al. (2007), Liu et al. (2004)	8 Chen et al. (2007), Depren et al. (2005), Giacinto and Roli (2003), Joo et al. (2003), Kayacik et al. (2007), Moradi and Zulkernine (2004), Stein et al. (2005), Zhang et al. (2005)	10 Abadeh et al. (2007), Bridges and Vaughn (2000), Depren et al. (2005), Luo and Bridgest (2000), Ozyer et al. (2007), Peddabachigari et al. (2007), Shon et al. (2006), Toosi and Kahani (2007), Xiang and Lim (2005)

Table 6
Yearwise distribution of feature selection considered.

	'07	'06	'05	'04	'03	'02	'01	'00	Total
Yes Abadeh et al. (2007), Bouzida et al. (2004), Bridges and Vaughn (2000), Chavan et al. (2004), Chen et al. (2007), Florez et al. (2002), Kayacik et al. (2007), Khan et al. (2007), Lee and Stolfo (2000), Leon et al. (2004), Liu and Yi (2006), Liu et al. (2007), Moradi and Zulkernine (2004), Ozyer et al. (2007), Ramos and Abraham (2005), Schultz et al. (2001), Shon and Moon (2007), Shyu et al. (2003), Stein et al. (2005), Tian et al. (2004), Toosi and Kahani (2007), Tsang et al. (2007), Wang and Battiti (2006), Wang et al. (2004), Xiang and Lim (2005), Zhang et al. (2005)	9	2	4	6	1	1	1	2	26
No Agarwal and Joshi (2000), Balajinath and Raghavan (2000), Chen et al. (2005), Chimphlee et al. (2006), Depren et al. (2005), Ertoz et al. (2003), Eskin et al. (2002), Fan et al. (2001), Giacinto and Roli (2003), Heller et al. (2003), Jiang et al. (2006), Joo et al. (2003), Li and Guo (2007), Liao and Vemuri (2002), Liu et al. (2004), Luo and Bridgest (2000), Mukkamala et al. (2004), Peddabachigari et al. (2007), Peddabachigari et al. (2004), Portnoy et al. (2001), Scott (2004), Shon et al. (2006), Wang and Stolfo (2004), Wang et al. (2006), Zhang et al. (2005), Zhang et al. (2004), Zhang and Shen (2005)	2	4	4	8	4	3	2	3	30

The result shows that integrated-based hybrid classifiers are the most considered hybrid classifier design approach for intrusion detection. In particular, they are mostly developed in 2007. On the other hand, cascaded hybrid classifiers are also largely considered in literature.

Note here that as the number of studies focusing on ensemble classifiers for intrusion detection is very few, they are not compared in this paper.

3.4. Baselines

There are many baseline approaches, against which more involved methods are compared, in the literature. That is, each work generally chooses different baseline classifiers to validate their intrusion detection systems. Table 4 shows yearwise distribution of articles using different techniques as their baselines.

Apparently, SVM is the most widely used baseline technique. In addition, it has also been considered recently for model comparisons. For related wok focusing on hybrid or ensemble classifiers, many of their baseline classifiers are only based on some of the above single classifiers.

3.5. Datasets

Table 5 shows yearwise distribution of datasets used for experiments. Note that some work divide one dataset into two datasets for two experiments.

Currently, as there are only few public datasets like KDD'99, DARPA 1998 and DARPA 1999, much related work considers these datasets for their experiments. Very few studies use non-public or their own datasets. This result shows that these public datasets are recognized as standard datasets in intrusion detection.

3.6. Feature selection

Before training, the step of feature (or variable) selection may be considered. The process of feature selection identifies which features are more discriminative than the others. This has the benefit of generally improving system performance by eliminating irrelevant and redundant features. Table 6 shows yearwise distribution of feature selection considered in related work.

This result reveals that not all studies perform feature selection before classifier training. In particular, 26 experiments considered feature selection. On the other hand, 30 experiments do not perform feature selection. In total, feature selection is not very popular procedure in intrusion detection. However, nine studies in 2007 use different feature selection methods for their experiments. This implies that feature selection could improve some certain level of classification accuracy in intrusion detection.

4. Discussion and conclusion

We have reviewed current studies of intrusion detection by machine learning techniques. In particular, this paper reviews recent papers which are between 2000 and 2007. In addition, we consider a large number of machine learning techniques used in the intrusion detection domain for the review including single, hybrid, and ensemble classifiers.

Regarding the comparative results of related work, developing intrusion detection systems using machine learning techniques still needs to be researched. The following issues could be useful for future research.

- Baseline classifiers. The chosen one single classifier for the model comparison and evaluation may be no longer a good candidate as the baseline classifier. It would be valuable if different ensemble classifiers and hybrid classifiers are compared in terms of prediction accuracy.
- The architecture of multiple classifiers. Designing more sophisticated classifiers via combining ensemble and hybrid classifiers can be examined. Since the idea of combining multiple classifiers is to collaborate each other instead of competition, it may be worth combining ensemble and hybrid classifiers for intrusion detection.
- Feature selection. As there are numbers of feature selection approaches, the reviewed studies which consider feature selection only choose one specific method, it is not known which method perform the best especially under what classification techniques for intrusion detection.

References

Abadeh, M. S., Habibi, J., Barzegar, Z., & Sergi, M. (2007). A parallel genetic local search algorithm for intrusion detection in computer networks. *Engineering Applications of Artificial Intelligence, 20*, 1058–1069.

Agarwal, R., & Joshi, M. V. (2000). *A new framework for learning classifier models in data mining*. Department of Computer Science, University of Minnesota.

Anderson, J. (1995). *An introduction to neural networks*. Cambridge: MIT Press.

Balajinath, B., & Raghavan, S. V. (2000). Intrusion detection through behavior model. *Computer Communication, 24*, 1202–1212.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. England: Oxford University.

Bouzida, Y., Cuppens, F., Cuppens-Boulahia, N., & Gombault, S. (2004). Efficient intrusion detection using principal component analysis. In *Paper presented at the proceedings of the 3eme conference surla securite et architectures reseaux (SAR)*. Orlando, FL, USA.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, P. J. (1984). *Classification and regressing trees*. California: Wadsworth International Group.

Bridges, S. M., & Vaughn, R. B. (2000). Intrusion detection via fuzzy data mining. In *Paper presented at the twelfth annual Canadian information technology security symposium*. Ottawa, USA.

Chavan, S., Shah, K. D. N., & Mukherjee, S. (2004). Adaptive neuro-fuzzy intrusion detection systems. In *Paper presented at the in proceedings of the international conference on information technology: Coding and computing (ITCC'04)*.

Chen, Y., Abraham, A., & Yang, B. (2007). Hybrid flexible neural-tree-based intrusion detection systems. *International Journal of Intelligent Systems, 22*, 337–352.

Chen, W.-H., Hsu, S.-H., & Shen, H.-P. (2005). Application of SVM and ANN for intrusion detection. *Computer and Operations Research, 32*, 2617–2634.

Chimphlee, W., Addullah, A. H., Sap, M. N. M., Srinoy, S., & Chimphlee, S. (2006). Anomaly-based intrusion detection using fuzzy rough clustering. In *Paper presented at the international conference on hybrid information technology (ICHIT'06)*.

Depren, O., Topallar, M., Anarim, E., & Ciliz, M. K. (2005). An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. *Expert Systems with Applications, 29*, 713–722.

Ertoz, L., Eilertson, E., Lazarevic, A., Tan, P.-N., Dokas, P., Kumar, V., et al. (2003). *Detection and Summarization of Novel Network Attacks Using Data Mining*.

Eskin, E., Arnold, A., Prerau, M., Portnoy, L., & Stolfo, S. (2002). *A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data*. Kluwer.

Fan, W., Lee, W., Miller, M., Stolfo, S. J., & Chan, P. K. (2001). Using artificial anomalies to detect unknown and known network intrusions. *Knowledge and Information Systems*, 507–527.

Fan, W., Lee, W., Miller, M., Stolfo, S. J., & Chan, P. K. (2004). Using artificial anomalies to detect unknown and known network intrusions. *Knowledge and Information Systems*, 507–527.

Florez, G., Bridges, S. M., & Vaughn, R. B. (2002). An improved algorithm for fuzzy data mining for intrusion detection. In *Paper presented at the proceedings of the North American fuzzy information processing society conference (NAFIPS 2002)*. New Orleans, LA.

Giacinto, G., & Roli, F. (2003). Intrusion detection in computer networks by multiple classifier systems. In *Paper presented at the proceeding of ICPR 2002, 16th international conference on pattern recognition*. Quebec City, Canada.

Giacinto, G., Perdisci, R., Rio, M. D., & Roli, F. (2006). Intrusion detection in computer networks by a modular ensemble of one-class classifiers. *Information Fusion, 9*, 69–82.

Giacinto, G., Perdisci, R., Rio, M. D., & Roli, F. (2008). Intrusion detection in computer networks by a modular ensemble of one-class classifiers. *Information Fusion, 9*, 69–82.

Han, S.-J., & Cho, S.-B. (2003). Detecting intrusion with ruled-based integration of multiple models. *Computers and Security, 22*(7), 613–623.

Haykin, S. (1999). *Neural networks: A comprehensive foundation* (2nd ed.). New Jersey: Prentice Hall.

Heller, K. A., Svore, K. M., Keromytis, A. D., & Stolfo, S. J. (2003). One class support vector machines for detecting anomalous window registry accesses. In *Paper presented at the 3rd IEEE conference data mining workshop on data mining for computer security*. Florida.

Jang, J.-S., Sun, C.-T., & Mizutani, E. (1996). *Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence*. New Jersey: Prentice Hall.

Jiang, S. Y., Song, X., Wang, H., Han, J.-J., & Li, Q.-H. (2006). A clustering-based method for unsupervised intrusion detections. *Pattern Recognition Letters, 27*, 802–810.

Joo, D., Hong, T., & Han, I. (2003). The neural network models for IDS based on the asymmetric costs of false negative errors and false positive errors. *Expert System with Applications, 25*, 69–75.

Kang, D. K., Fuller, D., & Honavar, V. (2005). Learning classifiers for misuse and anomaly detection using a bag of system calls representation. In *Paper presented at the proceeding of the 2005 IEEE*.

Kayacik, H. G., Nur, Z.-H., & Heywood, M. I. (2007). A hierarchical SOM-based intrusion detection system. *Engineering Applications of Artificial Intelligence, 20*, 439–451.

Khan, L., Awad, M., & Thuraisingham, B. (2007). A new intrusion detection system using support vector machines and hierarchical clustering. *The VLDB Journal, 16*, 507–521.

Kittler, J., Hatef, M., Duin, R. P. W., & Matas, J. (1998). On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20*(3), 226–239.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics, 43*, 59–69.

Koza, J. R. (1992). *Genetic programming: On the programming of computers by means of natural selection*. Massachusetts: MIT.

Lee, W., & Stolfo, S. (1998). Data mining approaches for intrusion detection. In *Paper presented at the proceedings of the seventh USENIX security symposium (SECURITY'98)*. San Antonio, TX.

Lee, W., & Stolfo, S. (2000). A framework for constructing features and models for intrusion detection systems. *ACM Transactions on Information and System Security (TISSEC), 3*(4), 227–261.

Leon, E., Nasraoui, O., & Gomez, J. (2004). Anomaly detection based on unsupervised niche clustering with application to network intrusion detection. *Evolutionary Computation, 1*, 502–508.

Liao, Y., & Vemuri, V. R. (2002). Use of *K*-nearest neighbor classifier for intrusion detection. *Computer and Security, 21*(5), 439–448.

Li, Y., & Guo, L. (2007). An active learning based TCM-KNN algorithm for supervised network intrusion detection. *Computer and Security, 26*, 459–467.

Liu, G., & Yi, Z. (2006). Intrusion detection using PCASOM neural networks. In *Paper presented at the proceeding of ISNN2006. Lecture notes in computer science*. Berlin, Heidelberg.

Liu, Y., Chen, K., Liao, X., & Zhang, W. (2004). A genetic clustering method for intrusion detection. *Pattern Recognition, 37*, 927–942.

Liu, G., Yi, Z., & Yang, S. (2007). A hierarchical intrusion detection model based on the PCA neural networks. *Neurocomputing, 70*, 1561–1568.

Luo, J., & Bridgest, S. M. (2000). Mining fuzzy association rules and fuzzy frequency episodes for intrusion detection. *International Journal of Intelligent Systems, 15*, 687–703.

Manocha, S., & Girolami, M. A. (2007). An empirical analysis of the probabilistic *K*-nearest neighbour classifier. *Pattern Recognition Letters, 28*, 1818–1824.

Michalski, R. S., Bratko, I., & Kubat, M. (1998). *Machine learning and data mining methods and applications*. Chichester, New York, Weinheim, Brisbane, Toronto, Singapore: Wiley.

Mitchell, T. (1997). *Machine learning*. New york: McGraw Hill.

Moradi, M., & Zulkernine, M. (2004). A neural network based system for intrusion detection and classification of attacks. In *Paper presented at the proceeding of the 2004 IEEE international conference on advances in intelligent systems – Theory and applications*. Luxembourg.

Mukkamala, S., Sung, A. H., & Abraham, A. (2004). Modeling intrusion detection systems using linear genetic programming approach. In *Paper presented at the proceedings of innovations in applied artificial intelligence, 17th international conference on industrial and engineering applications of artificial intelligence and expert systems (IEA/AIE). Lecture notes in computer science* (Vol. 3029). Springer.

Mukkamala, S., Sung, A. H., & Abraham, A. (2005). Intrusion detection using an ensemble of intelligent paradigms. *Network and Computer Applications, 28*, 167–182.

Ozyer, T., Alhajj, R., & Barker, K. (2007). Intrusion detection by integrating boosting genetic fuzzy classifier and data mining criteria for rule pre-screening. *Journal of Network and Computer Applications, 30*, 99–113.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems*. Morgan Kaufmann.

Peddabachigari, S., Abraham, A., Grosan, C., & Thomas, J. (2007). Modeling intrusion detection system using hybrid intelligent systems. *Journal of Network and Computer Applications, 30*, 114–132.

Peddabachigari, S., Abraham, A., & Thomas, J. (2004). Intrusion detection systems using decision trees and support vector machines. *International Journal of Applied Science and Computations*.

Portnoy, L., Eskin, E., & Stolfo, S. J. (2001). Intrusion detection with unlabeled data using clustering. In *Paper presented at the proceedings of ACM CSS workshop on data mining applied to security (DMSA-2001)*. Philadelphia, PA.

Ramos, V., & Abraham, A. (2005). ANTIDS: Self organized ant based clustering model for intrusion detection system. In *Paper presented at the proceedings of the fourth IEEE international workshop on soft computing as transdisciplinary science and technology (WSTST'05)*. Berlin: Springer-Verlag.

Rhodes, B., Mahaffey, J., & Cannady, J. (2000). Multiple self-organizing maps for intrusion detection. In *Paper presented at the proceedings of the 23rd national information systems security conference*. Baltimore, MD.

Schultz, M. G., Eskin, E., Zadok, E., & Stolfo, S. J. (2001). Data mining methods for detection of new malicious executables. In *Paper presented at the proceedings of the 2001 IEEE symposium on security and privacy (SP'01)*.

Scott, S. L. (2004). A Bayesian paradigm for designing intrusion detection systems. *Computational Statistics and Data Analysis, 45*, 69–83.

Shon, T., Kovah, X., & Moon, J. (2006). Applying genetic algorithm for classifying anomalous TCP/IP packets. *Neurocomputing, 69*, 2429–2433.

Shon, T., & Moon, J. (2007). A hybrid machine learning approach to network anomaly detection. *Information Sciences, 177*, 3799–3821.

Shyu, M., Chen, S., Sarinnapakorn, K., & Chang, L. (2003). A novel anomaly detection scheme based on principal component classifier. In *Paper presented at the proceedings of ICDM'03*.

Stallings, W. (2006). *Cryptography and network security principles and practices*. USA: Prentice Hall.

Stein, G., Chen, B., Wu, A. S., & Hua, K. A. (2005). Decision tree classifier for network intrusion detection with GA-based feature selection. In *Paper presented at the proceedings of the 43rd annual Southeast regional conference*. Kennesaw, Georgia.

Theodoridis, S., & Koutroumbas, K. (2006). *Pattern recognition*. Amsterdam, Boston, Heidelberg, London, New York, Oxford, Paris, San Diego, San Francisco, Singapore, Sydney, Tokyo: Academic Press.

Theodoridis, S., & Koutroumbas, K. (2006). *Pattern recognition* (3rd ed.). USA: Academic Press.

Tian, M., Chen, S. -C., Zhuang, Y., & Liu, J. (2004). Using statistical analysis and support vector machine classification to detect complicated attacks. In *Paper presented at the proceedings of the third international conference on machine learning and cybernetics*. Shanghai.

Toosi, A. N., & Kahani, M. (2007). A new approach to intrusion detection based on an evolutionary soft computing model using neuro-fuzzy classifiers. *Computer Communication, 30*, 2201–2212.

Tsang, C.-H., Kwong, S., & Wang, H. (2007). Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection. *Pattern Recognition, 40*, 2373–2391.

Vapnik, V. (1998). *Statistical learning theory*. New York: John Wiley.

Wang, W., & Battiti, R. (2006). Identifying intrusions in computer networks with principal component analysis. In *Paper presented at the proceedings of the first international conference on availability, reliability and security (ARES'06)*.

Wang, K., & Stolfo, S. J. (2004). Anomalous Payload-based network intrusion detection. In *Paper presented at the proceedings of recent advance in intrusion detection (RAID)*. Sophia Antipolis, France.

Wang, W., Guan, X., & Zhang, X. (2004). A novel intrusion detection method based on principle component analysis in computer security. In *Paper presented at the proceedings of the international symposium on neural networks*. Dalian, China.

Wang, Y., Kim, I., Mbateng, G., & Ho, S.-Y. (2006). A latent class modeling approach to detect network intrusion. *Computer Communications, 30*, 93–100.

Xiang, C., & Lim, S. M. (2005). Design of multiple-level hybrid classifier for intrusion detection system. In *Paper presented at the proceeding of the IEEE workshop machine learning for signal processing*.

Zhang, C., Jiang, J., & Kamel, M. (2005). Intrusion detection using hierarchical neural network. *Pattern Recognition Letters, 26*, 779–791.

Zhang, Z., & Shen, H. (2005). Application of online-training SVMs for real-time intrusion detection with different considerations. *Computer Communications, 28*, 1428–1442.

Zhang, L.-H., Zhang, G.-H., Yu, L., Zhang, J., & Bai, Y.-C. (2004). Intrusion detection using rough set classification. *Journal of Zhejiang University Science, 5*(9), 1076–1086.

Zimmermann, H. (2001). *Fuzzy set theory and its applications*. Kluwer Academic Publishers.