Pattern Recognition 43 (2010) 222 – 229

Contents lists available at ScienceDirect

Pattern Recognition

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

A triangle area based nearest neighbors approach to intrusion detection

Chih-Fong Tsai*, Chia-Ying Lin

Department of Information Management, National Central University, Taiwan

ARTICLE INFO

Article history:
Received 3 December 2008
Received in revised form 8 March 2009
Accepted 24 May 2009

Keywords:
Intrusion detection
Machine learning
Triangle area
k-means
k-nearest neighbors
Support vector machines

ABSTRACT

Intrusion detection is a necessary step to identify unusual access or attacks to secure internal networks. In general, intrusion detection can be approached by machine learning techniques. In literature, advanced techniques by hybrid learning or ensemble methods have been considered, and related work has shown that they are superior to the models using single machine learning techniques. This paper proposes a hybrid learning model based on the triangle area based nearest neighbors (TANN) in order to detect attacks more effectively. In TANN, the *k*-means clustering is firstly used to obtain cluster centers corresponding to the attack classes, respectively. Then, the triangle area by two cluster centers with one data from the given dataset is calculated and formed a new feature signature of the data. Finally, the *k*-NN classifier is used to classify similar attacks based on the new feature represented by triangle areas. By using KDD-Cup '99 as the simulation dataset, the experimental results show that TANN can effectively detect intrusion attacks and provide higher accuracy and detection rates, and the lower false alarm rate than three baseline models based on support vector machines, *k*-NN, and the hybrid centroid-based classification model by combining *k*-means and *k*-NN.

1. Introduction

The Internet has become a part of our daily life today. It is now an essential tool in the world and aids people in many areas such as business, biology, education, etc. Business in particular uses the Internet as an important component in their business models [1]. Not only businesses use the Internet in their daily operations and communications with their customers, but also customers use the Internet applications such as e-mail and websites to do some business activities. Therefore, information security needs to be concerned over the Internet environment.

It is common to find that the Internet environment has some risks of attacking. In order to prevent the attacks, many systems are designed to thwart Internet-based attacks. Intrusion detection is one of the major information security research problems in the Internet-based attacks.

The goal of an Intrusion Detection System (IDS) is to provide a layer of defense against malicious uses of computer systems by sensing a misuse or a breach of a security policy and alerting operators to an ongoing attack. An IDS is used to detect all types of malicious network traffic and computer usage that cannot be detected by a conventional firewall. Intrusion detection is based on the assumption that the behavior of the intruder differs from that of a legitimate user in ways that can be qualified [2].

In addition, an IDS is able to resist external attacks. Existing IDSs can be divided into two categories according to the detection approaches: anomaly detection and misuse detection or signature detection [3].

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. These systems seem to be effective and efficient. However, there are still many drawbacks to these Internet-based IDSs. The major problem is that they fail to generalize to detect new attacks without known signatures [4]. Besides, a potential drawback of these techniques is the rate of false alarms. This can happen primarily because the previously unseen system's behaviors may also be recognized as anomalies, and hence flagged as potential intrusions [5]. In addition, Shon et al. [6] indicate that these systems become a single point of failure. If the deployed IDS system is disabled for any reason, then it often gives the attacker the time to compromise the systems and possibly gain a foothold in the network.

In order to solve the problems mentioned above, numbers of anomaly detection systems are developed based on machine learning techniques. These systems use a "normal behavior" to detect those unexpected attacks. In particular, supervised and unsupervised

* Corresponding author. Tel.: +886 3 422 7151; fax: +886 3 4254604.
E-mail address: cftsai@mgt.ncu.edu.tw (C.-F. Tsai).

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.05.017

machine learning techniques have both been considered for anomaly detection (c.f. Section 2.2).

In this paper, we propose a novel method based on the idea of the triangle area based nearest neighbors (TANN) by combining unsupervised and supervised learning techniques to detect attacks. Given a dataset D containing m examples ($D = d_1, \ldots, d_m$), a triangle area by two cluster centers obtained by k-means and one data (d_i) from D is calculated to form a new feature for the data (d_i). For classification, the k-nearest neighbor (k-NN) classifier is used to measure the similarity of attacks based on the feature of triangle areas.

This paper is organized as follows. Section 2 briefly describes the concept of hybrid learning techniques and a number of related studies are compared in terms of their techniques developed, datasets used, evaluation methods considered, and baseline models compared. Section 3 introduces the proposed TANN approach. Section 4 presents the experimental setup and results. Conclusion and future work are provided in Section 5.

2. Literature review

2.1. Hybrid machine learning

In general, a hybrid machine learning model is based on combining the clustering and classification techniques. Related work developing the hybrid model is usually based on one clustering technique used as the first component for "pre-classification" and one classification technique as the second component for the final classification task [7–9]. In particular, the first clustering technique can be used to filter out unrepresentative data as performing the data reduction (or outlier detection) task. That is, the data which cannot be clustered accurately can be regarded as noisy data or outliers. Then, the representative data without the noisy data are used to train the classifier in order to improve the classification result.

On the other hand, the classification technique can be used as the first component and the clustering technique for the second one. This is because clustering is the unsupervised learning technique and it cannot distinguish data accurately like supervised one. Therefore, a classifier can be trained at first, and its output is subsequently used as the input for the cluster to improve the clustering result.

Centroid-based classification is one specific hybrid learning approach [10]. Centroid-based models find a representative data point for a particular class, which is called the centroid. Given a set of n data vectors $D = \{\vec{d_1}, \ldots, \vec{d_n}\}$, classified along a set C of m classes, $C = \{C_1, \ldots, C_m\}$, we use $D_{C_j} (1 \leq j \leq m)$ to represent the set of data vectors belonging to class C_j. The centroid of a particular class C_j is represented by a vector $\vec{C_j}$, which is a combination of the data vector $\vec{d_j}$ belonging to that class.

Therefore, centroid-based models are very efficient during the training and classification stages as they are based on the number of centroids, rather than the number of the original training data. That is, there is little computation is involved.

2.2. Related work

Table 1 compares a number of recent related work in terms of their detection techniques developed, datasets used, evaluation methods considered, baseline classifiers for comparisons, etc.

Regarding Table 1, there are several issues which can be discussed. For the problem domain, anomaly detection is the mostly considered research problem in literature. Second, much related work for intrusion detection uses the datasets of DARPA1998 and KDD-Cup '99 for experiments. In particular, the KDD-Cup '99 dataset is the mostly used. For the evaluation methods, detection rate (DR), false positive (FP), false negative (FN), true positive (TP), false alarm (FA), and accuracy are examined mostly. Finally, support vector

Table 1
Comparisons of related work.

Work	Technique	Dataset	Problem domain	Evaluation method	Baseline
Abadeh et al. [11]	GA[a]+FL[b]	DARPA 1998	Anomaly detection	DR[c], FA[d]	FL
Chen et al. [12]	GA+ANN[e]	DARPA 1998	Anomaly detection	FP[f], FN[g]	ANN, 2 layer ANN
Kayacik et al. [13]	SOM	KDD-Cup '99	Anomaly detection	FP, DR	SVM[h], k-NN
Khan et al. [7]	SOM[i]+SVM	DARPA 1998	Anomaly detection	FP, FN, accuracy	SVM
Li and Guo [14]	TCM k-NN	KDD-Cup '99	Anomaly detection	TP[j], FP	SVM, ANN, k-NN
Liu et al. [15]	SOM+ANN	DARPA 1998	Anomaly and misuse detection	DR, FA, FP	SVM, DT[k], SOM
Ozyer et al. [16]	Genetic fuzzy classifier	KDD-Cup '99	Anomaly and misuse detection	DR	GA
Peddabachigari et al. [17]	DT+SVM	KDD-Cup '99	Anomaly and misuse detection	Accuracy	SVM, DT
Shon and Moon [1]	GA+SVM	DARPA 1999	Anomaly detection	DR, FP, FN	SVM
Shon et al. [6]	GA+ANN/k-NN/SVM	DARPA 1998	Anomaly detection	DR, FP, FN	ANN, k-NN, SVM
Wang et al. [18]	Bayesian latent class	KDD-Cup '99	Anomaly detection	DR, FP	LR[l]
Chen et al. [19]	SVM, ANN	DARPA 1998	Anomaly detection	DR, FP	SVM, ANN
Mukkamala et al. [20]	Ensemble of SVM/ANN	KDD-Cup '99	Anomaly detection	Accuracy	SVM, ANN
Zhang and Shen [21]	Robust SVM, one-class SVM	DARPA 1998	Anomaly detection	DR, FA	SVM, k-NN
Zhang et al. [8]	C-means clustering+ANN	KDD-Cup '99	Anomaly and misuse detection	DR, FP	ANN
Liu et al. [9]	Nearest neighbor clustering+GA	KDD-Cup '99	Anomaly detection	DR, FP	GA
Peddabachigari et al. [22]	SVM, DT	KDD-Cup '99	Anomaly detection	Accuracy	SVM, DT, ANN

[a]GA: genetic algorithm.
[b]FL: fuzzy logic.
[c]DR: detection rate.
[d]FA: false alarm.
[e]ANN: artificial neural networks.
[f]FP: false positive.
[g]FN: false negative.
[h]SVM: support vector machines.
[i]SOM: self-organizing maps.
[j]TP: true positive.
[k]DT: decision trees.
[l]LR: logistic regression.

machines and *k*-NN are two of the most popular baseline classifiers for comparisons.

In related work, it is common to combine two techniques, especially for combining some clustering technique as the first component and classification for the second one (e.g. [7,8,15]), which belongs to one of the hybrid approaches described above. No matter what kind of approaches proposed in related work (including hybrid approaches), they always focus on solving the classification/detection problem over the original feature space. That is, if the chosen dataset contains *n*-dimensional features for each data, the classifiers they train and test are based on these features. However, our proposed approach is to transform the original multi-dimensional feature space into triangle areas, which can be obtained by the distance between cluster centers and data points. Then, the triangle area based features are regarded as the new feature space and they are used for the final classification decision. (See Section 3 for the detailed description of this approach.)

3. The triangle area based nearest neighbors approach

The proposed approach (TANN) is composed of three stages, which are clustering centers extraction, the new data formation by the triangle area, and *k*-NN training and testing based on the new data.

The idea behind TANN is to extend the centroid-based and nearest neighbor classification approaches. Specifically, we assume that all the centroids over a given dataset have their discrimination capabilities for distinguishing both similar and dissimilar classes. That is, the distance between an unknown data and its nearest centroid and other distances between this unknown data and other centroids can be all considered for classification. Therefore, in the feature space, an unknown data with any two centroids can result in a triangle area, thus TANN is proposed and intended to be able to improve classification performances over the centroid-based and nearest neighbor approaches. In particular, using the triangle area as the feature space for classification is the novelty of this paper.

3.1. Extraction of cluster centers

First of all, the *k*-means algorithm is used as the clustering method to find out five cluster centers of each category as the representative data points over the dataset. Note that as the KDD-Cup '99 dataset is used in this paper (c.f. Section 4.1.1), which includes one type of normal access and four types of Internet attacks, five cluster centers are extracted. Therefore, the value of *k* is set by 5. Fig. 1 shows the process of this stage.

3.2. New data formation by triangle areas

To calculate a triangle area in the feature space, three data points need to be provided. In this stage, two cluster centers obtained by *k*-means and one data point from the dataset are used to form a triangle area. Fig. 2 shows an example of the five cluster centers (*A*, *B*, *C*, *D*, and *E*) and one data point (*Xi*). Subsequently, ten triangle areas are obtained to form a new feature vector for the data point (*Xi*). That is, *ΔXiAB*, *ΔXiAC*, *ΔXiAD*, *ΔXiAE*, *ΔXiBC*, *ΔXiBD*, *ΔXiBE*, *ΔXiCD*, *ΔXiCE*, and *ΔXiDE*.

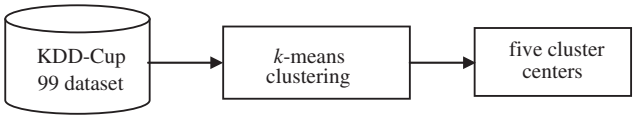

Fig. 1. Cluster centers extraction.

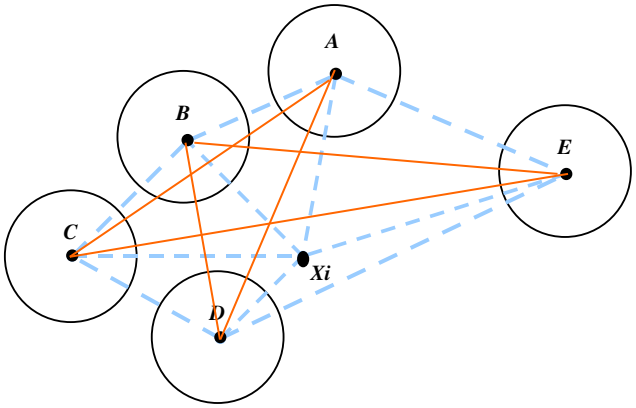

Fig. 2. An example of forming triangle areas.

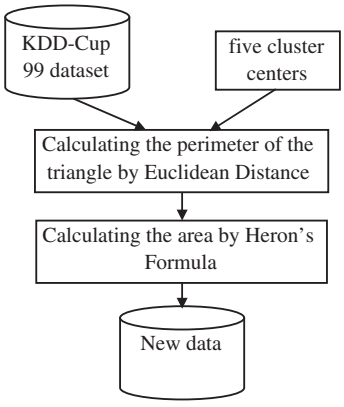

Fig. 3. New training data formation.

Fig. 3 shows the steps of forming the new data, and the followings describe these steps, respectively.

3.2.1. Perimeter of the triangle—Euclidean distance

First of all, three points are selected in order to form a triangle area. The first one is based on one of the original data points in the training data (i.e. *Xi*). The other two are from the five cluster centers generated by *k*-means in the first stage.

We define the data point *Xi* ($i = 1, \ldots, m$, where m is the total number of the data samples). Then, the Euclidean distance formula is used to measure the distance between two points. The Euclidean distance between points $A=(a_1, a_2, a_3, \ldots, a_n)$ and $B=(b_1, b_2, b_3, \ldots, b_n)$ in the *n*-feature space (see Fig. 4), can be defined as

$$dis\,AB = \sqrt{(a_1-b_1)^2+(a_2-b_2)^2+\cdots+(a_n-b_n)^2} = \sqrt{\Sigma(a_i-b_i)^2} \quad (1)$$

The perimeter of the triangle is defined as $G = a+b+c$, where $a=\overline{AB}$, $b=\overline{BX_i}$, and $c=\overline{AX_i}$ (i.e. the distances between *A* and *B*, *B* and *Xi*, and *A* and *Xi*, respectively).

3.2.2. Forming the triangle area—Heron's formula

After obtaining the perimeter of the triangle for each data point corresponding to two cluster centers, the triangle area by Heron's formula can be calculated (see Fig. 5). Heron's formula states that the area, *T*, of a triangle whose sides have lengths *a*, *b*, and *c* is

$$T = \sqrt{S(S-a)(S-b)(S-c)} \quad (2)$$

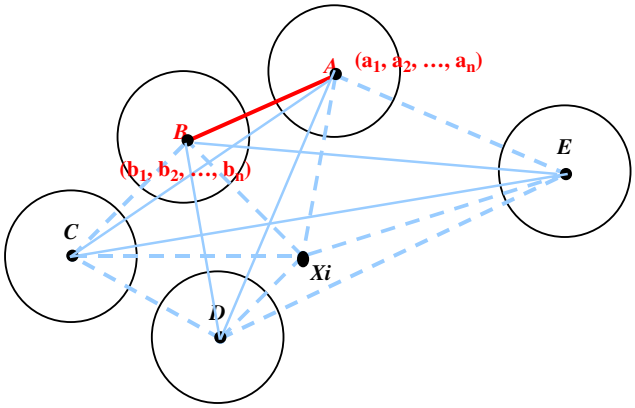

Fig. 4. The Euclidean distance between *A* and *B*.

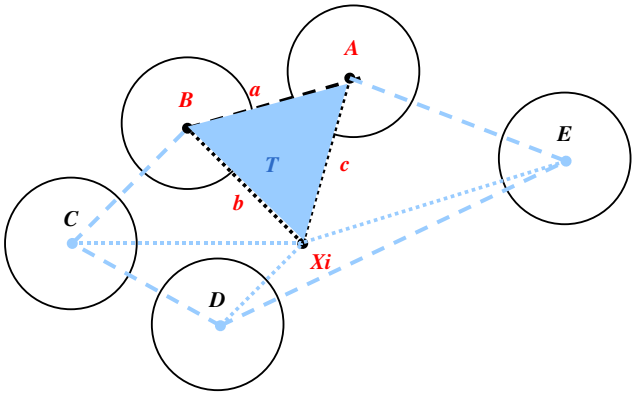

Fig. 5. Calculation of a triangle area.

where S is the semiperimeter of the triangle

$$S = (a + b + c)/2 \tag{3}$$

3.2.3. Formation of new data

At first, *Xi* with two out of the five centers (C_1, C_2, C_3, C_4, and C_5) generated by *k*-means ($k = 5$) form a triangle. Then, 10 triangles, T_1–T_{10} for each *Xi* are used to form the new data. In this process, we define the sum total as

$$\Delta X_i, C_1, C_2, \ldots, C_5 = \sum_{n=1}^{4} \Delta X_i C_n C_{n+1} + \sum_{n=1}^{3} \Delta X_i C_n C_{n+2} + \sum_{n=1}^{2} \Delta X_i C_n C_{n+3} + \Delta X_i C_1 C_5 \tag{4}$$

That is, the original data point *Xi* which is represented by $f_1, f_2, \ldots, f_n$ as the *n*-dimensional features and o_i as the output label of *Xi* are then transformed to $\Delta X_i C_1 C_2$, $\Delta X_i C_1 C_3$, $\Delta X_i C_1 C_4$, $\Delta X_i C_1 C_5$, $\Delta X_i C_2 C_3$, $\Delta X_i C_2 C_4$, $\Delta X_i C_2 C_5$, $\Delta X_i C_3 C_4$, $\Delta X_i C_3 C_5$, and $\Delta X_i C_4 C_5$ as the 10 triangle areas to represent the new feature of *Xi*.

3.3. Training and testing k-NN

Finally, the new data are used to train and test the *k*-NN classifier. That is, the new data composed of triangle area based features are divided into training and testing sets based on *n*-fold cross-validation (c.f. Section 4.1.1), in which the training and testing sets are used to train and test the *k*-NN classifier, respectively. Fig. 6 shows the process of training and testing *k*-NN.

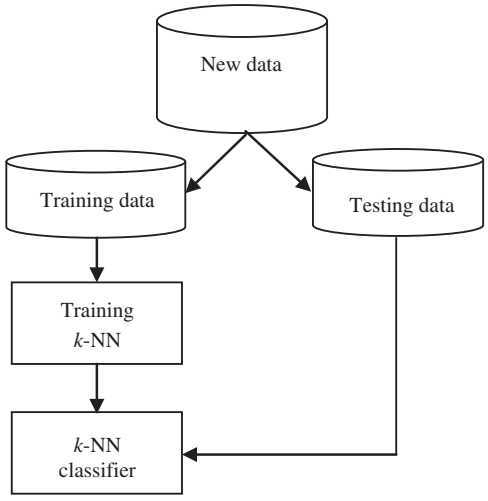

Fig. 6. Training and testing *k*-NN.

4. Experiments

4.1. Experimental setup

4.1.1. The dataset

Since there is no standard dataset for intrusion detection, the dataset used in this paper is based on the KDD-Cup '99 dataset[1] which has been considered mostly in related work (c.f. Section 2.2). In order to reasonably assess the proposed approach, "10% of KDD-Cup '99", and "Corrected (Test)" of the KDD-Cup '99 dataset are used as the training/testing and validation sets, respectively. It should be noted that much related work described in Section 2.2 only uses one specific KDD-Cup '99 dataset for classifier training and testing where the testing result is regarded as the final classification performance. Therefore, they do not further consider another dataset for validation.

In these two datasets, each pattern represents a network connection represented by a 41-dimensional feature vector, in which nine features are of the intrinsic types, 13 features are of the content type, and the remaining 19 features are of the traffic type. Each pattern of the dataset is labeled as belonging to one out of five classes, which are *normal* traffic and four different classes of attacks as follows:

- *Probing*: an attacker scans a network of computers to find known vulnerabilities, e.g. port scanning;
- *Denial of Service* (DoS): unauthorized access from a remote machine, e.g. guessing password;
- *Remote to Local* (R2L): unauthorized access to local super-user (root) privileges, e.g., various "buffer overflow" attacks;
- *User to Root* (U2R): surveillance and other probing, e.g., port scanning.

For the "10% of KDD-Cup '99" dataset, the distribution of normal and attack types of the connection records for classifier training and testing is summarized in Table 2.

During the training and testing stages, 10-fold cross-validation is used to avoid the variability of the samples that affects the performance of model training and testing. In 10-fold cross-validation, the whole dataset is divided into 10 unduplicated subsets. Nine of the 10 subsets are used for training and the remainder is served as the testing subset. Therefore, there are 10 classification results by 10-fold

[1] http://www.sigkdd.org/kddcup/index.php?section = 1999&method = data

Table 2
Sample distributions of the dataset.

Class	No. of samples	Samples percentage (%)
Normal	97,277	19.69
Probe	4107	0.83
DoS	391,458	79.24
U2R	52	0.01
R2L	1126	0.23
	494,020	100

Table 3
Sample distributions of the validation dataset.

Class	No. of samples	Samples percentage (%)
Normal	60,593	19.4
Probe	4166	1.33
DoS	231,455	74.4
U2R	88	0.028
R2L	14,727	4.73
	311,029	100

cross-validation. Then, classification accuracy can be averaged by the 10 classification results.

In addition, a validation dataset, i.e. "Corrected (Test)", is further used in order to validate the developed classifiers. Table 3 shows the dataset information.

4.1.2. Dimensionality reduction

In general, the dataset for intrusion detection contains a very large and high dimensional data. In order to reduce the problem that high dimensional data may affect (or degrade) detection performances, principal component analysis (PCA) is used to extract representative features. PCA has been widely used in the domain of intrusion detection [23], such as [15,16].

In particular, we set the factor loadings equal to or greater than 0.5 to extract important features from the chosen dataset. There are six features extracted, which are "land", "urgent", "num_failed_logins", "num_shells", "is_host_login", and "num_outbound_cmds".

Then, these selected features for each data point of the dataset are used to create the baseline and TANN classifiers.

4.1.3. Baseline classifiers

Regarding Table 1 (c.f. Section 2.2), much related work uses SVM as the baseline classifier, i.e. 10 studies out of 17. In addition, k-NN is also considered for comparisons since it can be conveniently used as a benchmark for all the other classifiers [24].

Therefore, to evaluate the proposed method, the k-NN and SVM classifiers are compared. Particularly, for k-NN, a number of different k values (i.e. $k = 1,3,5,\ldots,25$) is examined in order to find out the optimal or best k-NN over the dataset. On the other hand, for SVM, the polynomial kernel function is used where the polynomial degree is set from 1 to 5 to obtain the SVM classifier providing the best performance for comparisons.

In addition, the centroid-based classifier is also considered which is based on combining k-means and k-NN since the idea of TANN is derived from these two classification techniques. The k-means algorithm is used to obtain the cluster centers (i.e. five centers in this case). Then, the five cluster centers are used as the training data for k-NN ($k = 1$) classification rather than using the whole original training dataset to construct a k-NN classifier.

Note that it is difficult to fairly compare different approaches proposed recently as shown in Table 1. This is because although much related work uses the same dataset, they consider different input variables (i.e. features), which may result in different classification results. In other words, these approaches may perform well over their individual feature settings. As there is no answer to which features of the KDD-Cup '99 dataset are more representative and no further funding for the good baseline(s) for comparisons, we consider the most widely used baseline, SVM, i.e. 11 studies out of 18 use SVM as the baseline (c.f. Table 1). In addition, two other classifiers, k-NN and the centroid-based classifier, which have similar characteristics to TANN, are also compared.

Table 4
Confusion matrix.

Actual	Predicted	
	Normal	Intrusions (attacks)
Normal	TN	FP
Intrusions (attacks)	FN	TP

4.1.4. Evaluation methods

The performance measures for intrusion detection can be calculated by a confusion matrix as shown in Table 4.

- True positives, the number of malicious executables correctly classified as malicious;
- True negatives (TN), the number of benign programs correctly classified as benign;
- False positives, the number of benign programs falsely classified as malicious;
- False negative, the number of malicious executables falsely classified as benign.

As a result, the rate of accuracy, detection and false alarm which are also examined in related work (c.f. Section 2.2) can be obtained by

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$Detection\ rate = \frac{TP}{TP + FP} \tag{6}$$

$$False\ alarm = \frac{FP}{FP + TN} \tag{7}$$

Besides the three evaluation measures, this paper further uses receiver operating characteristics (ROC) curves to compare the performances of the created classifiers, which is less considered in related work. The ROC curve is a way of visualizing the trade-offs detection and false positive rates. In the ROC curve, the diagonal line $y = x$ represents the strategy of randomly guessing a class. Any classifier that appears in the upper left triangle performs better than random guessing [25].

4.2. Experimental results

4.2.1. k-NN

After testing a number of different k values, we find out the best value for k is 21. Tables 5 and 6 show the classification results for each of the five classes and the binary classes (normal and attacks), respectively.

In short, the rates of average accuracy, detection, and false alarm of k-NN ($k = 21$) are 93.87%, 93.39%, and 28.69%, respectively.

4.2.2. SVM

Similar to k-NN, Tables 7 and 8 show the classification results of SVM for each of the five classes and the binary classes (normal and attacks), respectively. SVM (degree = 2) performs the best. It provides 94.98% average accuracy, the 98.97% detection rate, and the 4.02% false alarm rate.

Table 5
Classification results of k-NN ($k = 21$) for the five classes.

Actual	Predicted					Accuracy (%)
	Normal	Probe	DoS	U2R	R2L	
Normal	6936	2378	393	20	0	71.31
Probe	23	384	4	0	0	93.43
DoS	216	1324	37,528	69	9	95.87
U2R	0	0	2	2	1	40.00
R2L	0	9	1	6	97	85.84
FP (%)	96.67	9.38	98.95	2.06	90.65	

Table 6
Classification results of k-NN ($k = 21$) for the normal and attack classes.

Actual	Predicted	
	Normal	Intrusions (attacks)
Normal	6936	2791
Intrusions (attacks)	239	39,436

Table 7
Classification results of SVM for the five classes.

Actual	Predicted					Accuracy (%)
	Normal	Probe	DoS	U2R	R2L	
Normal	9336	378	13	0	0	95.98
Probe	12	397	2	0	0	96.59
DoS	2076	4464	32,433	139	34	82.85
U2R	0	0	2	3	0	60.00
R2L	0	7	1	16	89	78.76
FP (%)	81.72	7.57	99.94	1.90	72.36	

Table 8
Classification results of SVM for the normal and attack classes.

Actual	Predicted	
	Normal	Intrusions (attacks)
Normal	9336	391
Intrusions (attacks)	2088	37,587

Table 9
Classification results of k-means+k-NN for the five classes.

Actual	Predicted					Accuracy (%)
	Normal	Probe	DoS	U2R	R2L	
Normal	9350	366	11	0	0	96.12
Probe	11	397	3	0	0	96.59
DoS	2076	4360	32,538	143	29	83.12
U2R	0	0	2	3	0	60.00
R2L	0	7	2	16	90	78.26
FP (%)	81.75	7.74	99.94	1.85	75.63	

4.2.3. Combining k-means and k-NN

Table 9 and 10 show the classification results of the centroid-based classifier for each of the five classes and the binary classes (normal and attacks), respectively. The rates of average accuracy, detection, and false alarm of this classifier are 95.01%, 99.01%, and 3.88%, respectively.

4.2.4. TANN

For the proposed approach, different k values of k-NN were also examined and we found out that when $k = 17$, TANN performs the best. Tables 11 and 12 show the classification results of TANN for each of the five classes and the binary classes (normal and attacks), respectively.

Table 10
Classification results of k-means+k-NN for the normal and attack classes.

Actual	Predicted	
	Normal	Intrusions (attacks)
Normal	9350	377
Intrusions (attacks)	2087	37,590

Table 11
Classification results of TANN for the five classes.

Actual	Predicted					Accuracy (%)
	Normal	Probe	DoS	U2R	R2L	
Normal	9436	278	13	0	0	97.01
Probe	15	390	6	0	0	94.89
DoS	185	3196	35,601	127	37	90.94
U2R	0	0	2	3	0	60.00
R2L	0	6	1	15	91	80.53
FP (%)	97.92	10.08	99.94	2.07	71.09	

Table 12
Classification results of TANN for the normal and attack classes.

Actual	Predicted	
	Normal	Intrusions (attacks)
Normal	9436	291
Intrusions (attacks)	200	39,475

The accuracy rate of TANN is 99.01% which outperforms the baselines of k-NN (93.87%), SVM (94.98%), the centroid-based classifier (95.01%). In addition, TANN also provides the higher detection rate (99.27%) than k-NN (93.39%), SVM (98.97%), and the centroid-based classifier (99.01%). Finally, for the false alarm rate, TANN also performs the best (2.99%) over k-NN (28.69%), SVM (4.02%) and the centroid-based classifier (3.88%).

4.2.5. Further comparisons

This section compares these classifiers in terms of their TP, TN, FP, FN, DR, FA, and accuracy performances over the testing and validation datasets as well as the accuracy of detecting four different types of attacks. The comparative results allow us to see the performance of each classifier and identify the best one for intrusion detection.

Tables 13 and 14 show the performances of these classifiers over the testing and validation datasets, respectively. Based on the testing and validation datasets, TANN performs the best. That is, it provides the highest accuracy and detection rates and the lowest false alarm rate.

Fig. 7 further examines the average accuracy of each class including one normal class and four different types of attacks.

It should be noted here that although TANN cannot effectively detect each of the four types of attacks compared with the three baseline classifiers, the major task of the intrusion detection system is to filter out potential attacks and allow normal connection to access. That is, the rate of detecting whether the connection is the normal access or attacks should be as high as possible at the first line of security. In this case, TANN performs the best over the three baseline classifiers.

Besides, Fig. 8 shows an ROC curve depicting the relationship between false positive and detection rates. The result indicates that TANN outperforms k-NN, SVM, and the centroid-based classifier.

Finally, the t test is used to examine the level of significance of these classifiers in terms of the accuracy, detection, and false alarm rates which are shown in Tables 15, 16 and 17, respectively. The result can be used to examine whether the classification performances of these classifiers are significantly different. In other words, a high

Table 13
Comparisons of these models by the testing dataset.

	True positive (TP)	True negatives (TN)	False positives (FP)	False negatives (FN)	Detection rate (DR) (%)	False alarm (FA) (%)	Accuracy (%)
k-NN	39,436	6936	2791	239	93.39	28.69	93.87
SVM	37,587	9336	391	2008	98.97	4.02	95.14
k-means+*k*-NN	37,590	9350	377	2087	99.01	3.88	95.01
TANN	39,475	9436	291	200	99.27	2.99	99.01

Table 14
Comparisons of these models by the validation dataset.

	True positive (TP)	True negatives (TN)	False positives (FP)	False negatives (FN)	Detection rate (DR) (%)	False alarm (FA) (%)	Accuracy (%)
k-NN	35,780	8143	4995	484	87.75	38.02	88.91
SVM	35,426	10,508	693	2775	98.08	6.19	92.98
k-means+*k*-NN	36,516	9702	488	2696	98.68	4.79	93.55
TANN	37,785	10,092	402	1123	98.95	3.83	96.91

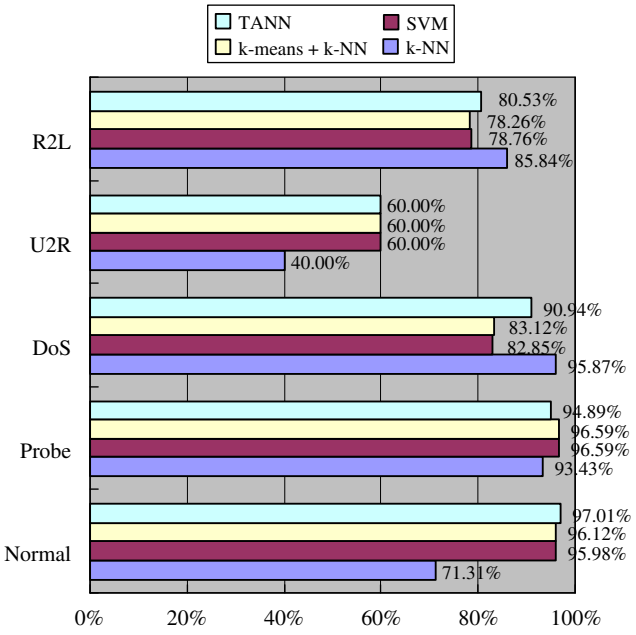

Fig. 7. Accuracy of each attack class.

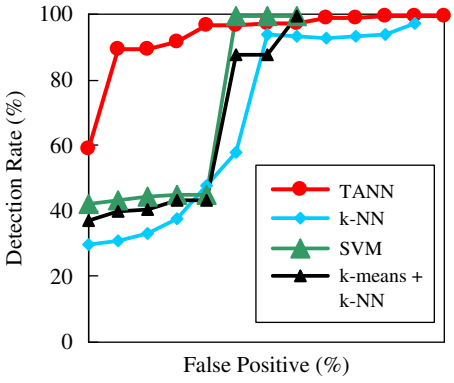

Fig. 8. ROC curves of these models.

level of significant difference (i.e. $p < 0.05$) between two classifiers implies that when using other datasets, the performances of these classifiers would provide very similar results to this paper.

The result indicates that TANN has a significant level of difference with *k*-NN, SVM, and the centroid-based classifier, respectively

Table 15
Result of *t* test (accuracy).

	TANN	*k*-means+*k*-NN	SVM	*k*-NN
TANN		0.272 ($t = 1.163$)	0.094 ($t = 1.853$)	0.000 ($t = 14.987$)
k-means+*k*-NN			0.813 ($t = 0.243$)	0.000 ($t = 10.491$)
SVM				0.000 ($t = 17.094$)

Table 16
Result of *t* test (detection rate).

	TANN	*k*-means+*k*-NN	SVM	*k*-NN
TANN		0.000 ($t = 6.988$)	0.000 ($t = 8.550$)	0.000 ($t = 8.937$)
k-means+*k*-NN			0.929 ($t = -0.092$)	0.095 ($t = 1.847$)
SVM				0.010 ($t = 3.171$)

Table 17
Result of *t* test (false alarm).

	TANN	*k*-means+*k*-NN	SVM	*k*-NN
TANN		0.001 ($t = -4.475$)	0.000 ($t = -5.779$)	0.000 ($t = -31.366$)
k-means+*k*-NN			0.200 ($t = -1.372$)	0.000 ($t = -30.769$)
SVM				0.000 ($t = -34.918$)

($p < 0.05$) over accuracy, the detection rate, and false alarm. Therefore, a reliable conclusion can be made that TANN significantly perform better than *k*-NN, SVM, and the centroid-based classifier over these three evaluation measures.

5. Conclusion

A modern computer network should acquire mechanisms to ensure the security policy of data and equipments inside the network. Intrusion Detection Systems (IDSs) are an integral package in any well configured and managed computer system or network. Two main approaches to intrusion detection are currently used, which are anomaly detection and misuse detection. In literature, machine learning techniques (supervised and/or unsupervised learning) have been considered for intrusion detection, especially for anomaly detection.

In this paper, we propose a novel hybrid method based on a triangle area based nearest neighbors approach, namely TANN, to detect attacks. In particular, *k*-means is firstly used to extract a number of cluster centers where each cluster center represents one particular type of attacks. Then, the triangle area can be calculated by two out of the cluster centers and one data point in the dataset. As a result,

these triangle areas represent a new feature for measuring similar attacks. Finally, for classification, the *k*-NN classifier is used based on the feature of triangle areas to detect intrusions.

By using the KDD-Cup '99 dataset with 10-fold cross-validation, TANN performs better than *k*-NN, SVM, and the centroid-based classifier in terms of average accuracy, the detection rate, false alarm, and the ROC curve. The *t* test also shows that the experimental results of these models have a high level of significant difference.

For future work, some issues could be considered. First, different clustering and classification techniques can be applied during the cluster center extraction and triangle area classification stages. For example, cluster and/or classifier ensemble can be used in these two stages, respectively. Secondly, it would be interesting to examine the performance of TANN over the datasets which contain different numbers of classes. That is, for the 5-class classification problem in this paper, there are 10 triangle areas as the new features for each data. If the problem contains a larger number of classes, say 20, then, the dimensionality of the new features as the number of triangle areas for each data will become very large. This may cause the "curse of the dimensionality" problem. Therefore, the stability and scalability of TANN for different numbers of classes for classification need to be examined in the future.

References

[1] T. Shon, J. Moon, A hybrid machine learning approach to network anomaly detection, Information Sciences 177 (2007) 3799–3821.
[2] W. Stallings, Cryptography and Network Security Principles and Practices, Prentice-Hall, USA, 2006.
[3] S. Northcutt, J. Novak, Network Intrusion Detection: An Analyst's Handbook, second ed, New Riders Publishing, 2000.
[4] W. Lee, S.J. Stolfo, P.K. Chan, E. Eskin, W. Fan, M. Miller, S. Hershkop, J. Zhang, Real time data mining-based intrusion detection, in: Proceedings of the DARPA Information Survivability Conference & Exposition II, Anaheim, USA, June 12–14, 2001, pp. 89–100.
[5] P. Dokas, L. Ertoz, V. Kumar, A. Lazarevic, J. Srivastava, P.-N. Tan, Data mining for network intrusion detection, in: Proceedings of NSF Workshop on Next Generation Data Mining, 2002.
[6] T. Shon, X. Kovah, J. Moon, Applying genetic algorithm for classifying anomalous TCP/IP packets, Neurocomputing 69 (2006) 2429–2433.
[7] L. Khan, M. Awad, B. Thuraisingham, A new intrusion detection system using support vector machines and hierarchical clustering, The VLDB Journal 16 (2007) 507–521.
[8] C. Zhang, J. Jiang, M. Kamel, Intrusion detection using hierarchical neural network, Pattern Recognition Letters 26 (2005) 779–791.
[9] Y. Liu, K. Chen, X. Liao, W. Zhang, A genetic clustering method for intrusion detection, Pattern Recognition 37 (2004) 927–942.
[10] A. Cardoso-Cachopo, A. Oliveira, Semi-supervised single-label text categorization using centroid-based classifiers, in: Proceedings of the ACM Symposium on Applied Computing, Seoul, Korea, March 11–15, 2007, pp. 844–851.
[11] M.S. Abadeh, J. Habibi, Z. Barzegar, M. Sergi, A parallel genetic local search algorithm for intrusion detection in computer networks, Engineering Applications of Artificial Intelligence 20 (8) (2007) 1058–1069.
[12] Y. Chen, A. Abraham, B. Yang, Hybrid flexible neural-tree-based intrusion detection systems, International Journal of Intelligent Systems 22 (2007) 337–352.
[13] H.G. Kayacik, Z.-H. Nur, M.I. Heywood, A hierarchical SOM-based intrusion detection system, Engineering Applications of Artificial Intelligence 20 (2007) 439–451.
[14] Y. Li, L. Guo, An active learning based TCM-KNN algorithm for supervised network intrusion detection, Computer and Security 26 (2007) 459–467.
[15] G. Liu, Z. Yi, S. Yang, A hierarchical intrusion detection model based on the PCA neural networks, Neurocomputing 70 (2007) 1561–1568.
[16] T. Ozyer, R. Alhajj, K. Barker, Intrusion detection by integrating boosting genetic fuzzy classifier and data mining criteria for rule pre-screening, Journal of Network and Computer Applications 30 (2007) 99–113.
[17] S. Peddabachigari, A. Abraham, C. Grosan, J. Thomas, Modeling intrusion detection system using hybrid intelligent systems, Journal of Network and Computer Applications 30 (2007) 114–132.
[18] Y. Wang, I. Kim, G. Mbateng, S.-Y. Ho, A latent class modeling approach to detect network intrusion, Computer Communications 30 (2006) 93–100.
[19] W.-H. Chen, S.-H. Hsu, H.-P. Shen, Application of SVM and ANN for intrusion detection, Computer and Operations Research 32 (2005) 2617–2634.
[20] S. Mukkamala, A.H. Sung, A. Abraham, Intrusion detection using an ensemble of intelligent paradigms, Network and Computer Applications 28 (2005) 167–182.
[21] Z. Zhang, H. Shen, Application of online-training SVMs for real-time intrusion detection with different considerations, Computer Communications 28 (2005) 1428–1442.
[22] S. Peddabachigari, A. Abraham, J. Thomas, Intrusion detection systems using decision trees and support vector machines, International Journal of Applied Science and Computations 11 (3) (2004) 118–134.
[23] A. Patcha, J.-M. Park, An overview of anomaly detection techniques: existing solution and latest technological trends, Computer Networks 51 (2007) 3448–3470.
[24] A.K. Jain, R.P.W. Duin, J. Mao, Statistical pattern recognition: a review, IEEE Transitions on Pattern Analysis and Machine Intelligence 22 (1) (2000) 4–37.
[25] T. Fawcett, An introduction to ROC analysis, Pattern Recognition Letters 27 (8) (2006) 861–874.

About the Author—CHIH-FONG TSAI obtained a Ph.D. at School of Computing and Technology from the University of Sunderland, UK in 2005 for the thesis entitled "Automatically Annotating Images with Keywords". He is now an Assistant Professor at the Department of Information Management, National Central University, Taiwan. He has published over 20 refereed journal papers including ACM Transactions on Information Systems, Information Processing & Management, Knowledge-Based Systems, Expert Systems with Applications, Expert Systems, Online Information Review, International Journal on Artificial Intelligence Tools, Neurocomputing, Journal of Systems and Software, etc. His current research focuses on multimedia information retrieval and data mining applications.

About the Author—Miss CHIA-YING LIN received the master degree in the Accounting and Information Technology Department of National Chung Cheng University, Taiwan, in 2008. Her current research interest focuses on data mining and intrusion detection.