An Innovative Two-Stage Fuzzy kNN-DST Classifier for Unknown Intrusion Detection

Xueyan Jing[1], Yingtao Bi[2], and Hai Deng[1]
[1]Department of Electrical and Computer Engineering, Florida International University, USA
[2]Feinberg School of Medicine, Northwestern University, USA

Abstract: *Intrusion detection is the essential part of network security in combating against illegal network access or malicious attacks. Due to constantly evolving nature of network attacks, it has been a technical challenge for an Intrusion Detection System (IDS) to recognize unknown attacks or known attacks with inadequate training data. In this work, an innovative fuzzy classifier is proposed for effectively detecting both unknown attacks and known attacks with insufficient or inaccurate training information. A Fuzzy C-Means (FCM) algorithm is firstly employed to softly compute and optimise clustering centers of the training datasets with some degree of fuzziness counting for inaccuracy and ambiguity in the training data. Subsequently, a distance-weighted k-Nearest Neighbors (k-NN) classifier, combined with the Dempster Shafer Theory (DST) is introduced to assess the belief functions and pignistic probabilities of the incoming data associated with each of known classes. Finally, a two-stage intrusion detection scheme is implemented based on the obtained pignistic probabilities and their entropy function to determine if the input data are normal, one of the known attacks or an unknown attack. The proposed intrusion detection algorithm is evaluated through the application of the KDD'99 datasets and their variants containing known and unknown attacks. The experimental results show that the new algorithm outperforms other intrusion detection algorithms and is especially effective in detecting unknown attacks.*

Keywords: *Network security, intrusion detection, FCM, classifiers, DST.*

Received June 10, 2014; accepted February 10, 2015; published online August 22, 2015

1. Introduction

With the explosive growth of internet and network usage recently, the computer network security has become one of the most important issues for networking systems and information infrastructure due to increasingly widespread cyber theft, fraud and abuse [7]. Intrusion Detection Systems (IDS) have been a critical part of the cyber security systems because they are capable to identify all types of malicious network connections in real-time and to trigger cyber-countermeasures to minimize or even eliminate any potential damage or security breaches to networking and local data systems. Therefore, IDS has drawn significant research attentions recently [30]. Basically, there are mainly two types of IDS: Anomaly intrusion detection and misuse or signature-based intrusion detection [10, 12, 24]. Anomaly intrusion detection tries to identify if the data traffic pattern is abnormal by comparing it with previously obtained normal traffic profiles; while the misuse methods detect intrusions by matching the data signature or feature vector to that of one of the known attacks. Although, theoretically capable to detect unknown attacks, anomaly IDS are generally inefficient, time-consuming and very difficult to implement with poor performance due to the lack of training data. On the other hand, misuse IDS are very efficient in data classification, but they can only detect known attacks and suffer from high error detection rate especially when the attacks are unknown or the classifier is not properly or sufficiently sufficiently trained. Most of current IDS algorithms are based on modern classification methods. The current classifying algorithms used in IDS are typically designed based on the naïve Bayesian method [15, 19], support vector machine [2, 4, 18, 25, 32], particle swarm optimization [6, 11], generic algorithm [29], neural networks [9, 13, 20, 22], k-Nearest Neighbor (kNN) methods [14, 17], Fuzzy C-Means (FCM) methods [23, 27], Dempster-Shafer theory of evidence [1, 8, 21, 32] or other decision-tree based ad-hoc methods [31]. However, the detection performance of the current IDS are generally sensitive to the mismatching between the training and test data; and they could perform very poorly in the case of slight deviation of intrusive data pattern from known patterns or of an unknown attack. It is well-known that the forms of cyber attacks and internet hacking tactics are constantly changing and evolving and new internet "viruses" are created almost every day. Therefore, it is imperative that the performance of IDS is not markedly degraded when a known intrusion is morphed into a different form of attack or a new intrusion has a completely different profile. In this work, we will introduce an innovative two-stage fuzzy classifier embedded with "soft" and error-tolerant classification mechanism for effective detection of various malicious intrusions including unknown attacks. Fuzzy classifiers are known for tolerating training or test data errors or variations due to "soft" clustering and classification techniques involved, but none of the current fuzzy classifiers is capable to effectively detect both known

and unknown attacks simultaneously. In addition, with this proposed IDS, the Dempster Shafer Theory (DST) [28] is seamlessly combined with a distance-weight kNN algorithm by fusing multiple "soft" independent evidences in order to assess the belief value of the input data belonging to each of the known classes. The two-stage classification in the scheme is set firstly to determine whether the incoming traffic is normal or an attack and subsequently to determine whether it is an unknown attack if the first stage detection is positive as abnormal connections. The rest of the paper is organized as follows. Section 2 describes the proposed two-stage fuzzy kNN-DST IDS in detail. Section 3 presents the experimental results using the proposed IDS. Section 4 draws some conclusions on this work.

2. A New Fuzzy kNN-DST Classifier for Unknown Intrusion Detection

The framework of the proposed fuzzy kNN-DST classifier is shown in Figure 1. Since, the new classifier should be capable to detect unknown intrusions and mutated versions of known intrusions; we choose to use the kNN for its robust performance and its tolerance for inaccuracy and random errors in the input data.

In our developed new classifier, a belief value of a test connection associated with a known class is softly measured based on the distance between the input data and the centroid of the class and the DST is incorporated into the framework to fuse multiple evidences generated from a weighted kNN to form a pignistic probability of a test connection belonging to a known class. The centers of known classes are softly defined and computed using a semi-supervised FCM method from the training data. Stage one classification in Figure 1 determines if a connection is normal data or an intrusion. If it is an abnormal intrusion, stage-two classification is needed to determine if it is one of known attacks or unknown attack. The details of the new classifier are given in the next two subsections.

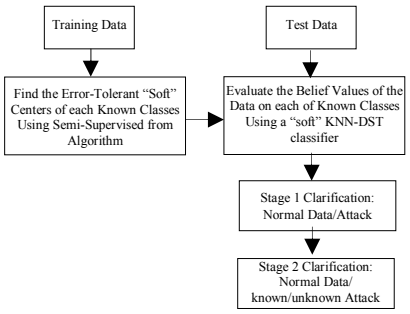

Figure 1. Two-stage fuzzy kNN-DST learning and classifying schemes for unknown attack detection.

2.1. Semi-Supervised FCM Learning Algorithm

Let us assume the training set $X=\{x_1, x_2, x_3, \ldots, x_N\}$ contains N network traffic connections and each of them is either normal connection or known attack. Each connection is represented by a distinct feature vector with positive numeric values. Normally, for computer network connections, the extracted feature vector consists of the source and destination bytes, the connection type or the duration of a connection. The set of features generated from all data connections are assumed to be:

$$F=\{f_1, f_2, f_3, \ldots, f_N\} \tag{1}$$

We denote the set $L=\{l_1, l_2, l_3, \ldots, l_P\}$ as P possible data classes, which include known attacks and the normal data stream. To avoid the crisp definition of a connection belonging to one of the classes, we employ the FCM algorithm allowing one traffic connection to belong to more than one class/cluster with varying membership values. Firstly, we will try to divide the N traffic connections into P clusters/classes and each cluster is represented by its centroid, which is an element of $C=\{c_1, c_2, c_3, \ldots, c_P\}$. In addition, a membership partition matrix U of size $N\times P$ is used to measure the closeness of a data connection to each of the class centers. The membership matrix elements are defined by:

$$u_{iq}=\frac{\|f_i, c_q\|^{\frac{-2}{\beta-1}}}{\sum_{q=1}^{p}\|f_i, c_q\|^{\frac{-2}{\beta-1}}}, \; 1\le i\le N, \; 1\le q\le P \tag{2}$$

Where u_{iq} of a value between 0 and 1 is the membership grade of the input data connection i in the cluster q, β (β>1) is the weighting exponent representing the degree of the fuzziness for the membership grades and $\|f_i, c_q\|$ represents the Mahalanobis distance between the data feature vector f_i and the centroid c_q of cluster q and is defined as:

$$\|f_i, c_q\|=\sqrt{(f_i - c_q)^T \Sigma_q^{-1}(f_i - c_q)} \tag{3}$$

Where Σ_q is the covariance matrix of the centroid vector of cluster q. Equation 3 becomes the Euclidean distance when Σ_q is the unity matrix.

The centroid of cluster q is further defined as:

$$c_q=\frac{\sum_{i=1}^{N} u_{iq}^{\beta} f_i}{\sum_{i=1}^{N} u_{iq}^{\beta}} \quad "q=1, 2, L, P \tag{4}$$

The cluster centroids are iteratively optimized by minimizing the following dissimilarity function $J(U, C)$:

$$J(U, C)=\sum_{i=1}^{N}\sum_{q=1}^{P} u_{iq}^{\beta} \|f_i, c_q\|^2$$
$$Subject\ to: \sum_{q=1}^{P} u_{iq}=1, "i \tag{5}$$

With the FCM algorithm, we keep on upgrading c_q and u_{iq} iteratively until the dissimilarity function $J(U, C)$ is minimized. The optimal cluster centroids c_q for the fuzzy classifier are found when the iteration stops with

$\max_{i,\,q}\left|u_{iq}^{(\eta+1)}-u_{iq}^{(\eta)}\right|<\varepsilon$, where ε is a pre-selected threshold between 0 and 1 and η is the number of iterations. The initial values of the cluster centroids in Equation 2 are obtained from the labelled training data directly. Therefore, the iterations in Equations 2, 3, 4 and 5 normally can converge quickly. Since, the class information of the labelled training data is used in the FCM algorithm, the learning process is considered to be semi-supervised.

2.2. A Two-Stage kNN-DST Classifier

With the centroids of the known clusters found through the FCM algorithm, we try to employ a weighted kNN approach in our classifier by considering the kNN of a new test data connection x_v in the training dataset. Let us associate the test data connection x_v with the class l_q of one of the kNN f_q by defining a fuzzy membership function u_{vq} based on the distance between the test data and the class centroid c_q that is similar to Equation 2. However, the association between the test data and class l_q also should be affected by the distance between the test data x_v and the training neighbor f_q. If there is a large distance between x_v and one of the k-nearest training records, the probability of x_v and the training record belonging to the same class is small. Therefore, the membership grade of a test data record belonging to a class should be weighted based on the distance between the test data and its nearest neighbors in the kNN algorithm. We assume that the kNN training data records of x_v are represented by the set of their feature spaces $F=\{f_1, f_2, \ldots, f_k\}$ and $\{\|f_v, f_1\|, \|f_v, f_2\|, \ldots, \|f_v, f_k\|\}$ is the set of the corresponding distances between the test data feature f_v and the K nearest training samples f_v in ascending order. Hence, the membership grade of a test data record belonging to a class in the weighted kNN algorithm should be weighted with the following coefficient.

$$w_\gamma=\begin{cases}\dfrac{\|f_K f_v\|-\|f_r f_v\|}{\|f_K f_v\|-\|f_1 f_v\|} & \|f_K f_v\|\neq\|f_1 f_v\|,\ 1\le\gamma\le K\\ 1 & \|f_K f_v\|=\|f_1 f_v\|\end{cases} \tag{6}$$

Where f_1 is the first nearest neighbor of f_v and f_k is the k^{th} nearest neighbour of f_v and the weight w_Y is assigned to modify the association between the connection f_v and the class of its Y^{th} nearest neighbor. The closer the neighbors are the greater weights they are assigned to. In the weighted kNN algorithm, each of the kNN and its class designation contribute to the classification by providing an independent piece of evidence. Since, DST is an evidence theory that can be used to combine separate pieces of evidence to determine the probability of an incident [24], we will use the DST in our classifier to fuse the class information obtained from all kNN to facilitate the intrusion detection work. The goal is to try to classify the new connection f_v into one of the members in class label set $L=\{1, 2, \ldots, P\}$. DST describes the probability of a test data sample belonging to a class using belief functions and the degree of belief is quantified by a mass function denoted as m. The term $m_y(l_q)$ of f_y can be treated as a piece of evidence that contributes to our belief that f_v belongs to class l_q. Since, only a part of our belief is committed to l_q and represented by $m_y(l_q)$, the rest of the belief is assigned to the whole frame of discernment represented by $m_y(L)$. Specifically, the belief functions of the input data connection f_v belonging to class l_q and the whole frame L due to the evidence from one of its neighbor f_y are defined, respectively, as:

$$m_\gamma(l_q)=\zeta\times w_\gamma\times u_{vq}=\zeta\times w_\gamma\times\frac{\|x_v, c_q\|^{\frac{-2}{\beta-1}}}{\sum_{q=1}^{P}\|x_v, c_q\|^{\frac{-2}{\beta-1}}} \tag{7}$$

$$m_\gamma(L)=1-\sum_{q=1}^{P}m_\gamma(l_q),\ \gamma=1,\ 2,\ \ldots, K \tag{8}$$

Where q is the class number, u_{vq} is the fuzzy membership grade of f_v associated with class l_q and is used to measure the belief of f_v belonging to class l_q, and ξ is a fixed factor used to normalize the total mass function.

Since, there are K nearest neighbours $\{f_y, y=1, 2, \ldots, K\}$ of f_v, each of them can be treated as a piece of evidence supporting our belief that f_v belongs to class l_q. By using the dempster rule of combination [28], we can fuse the mass functions of all kNN f_y belonging to the same class l_q to form a combined mass function through orthogonal sum of the mass functions, represented as:

$$m_{<q>}(l_q)=m_1(l_q)\oplus m_2(l_q)\oplus,\ \cdots,\ \oplus m_K(l_q) \tag{9}$$

Based on the mass functions that are assigned to class q for all training data connections $\{f_y, y=1, 2, \ldots, K\}$ in the kNN, the combined mass functions for the data sample assigned to class q and the whole frame L are given, respectively by:

$$m_{<q>}(l_q)=1-\prod_{r=1}^{K}(1-m_\gamma(l_q)) \tag{10}$$

$$m_{<q>}(L)=\prod_{r=1}^{K}(1-m_\gamma(l_q)) \tag{11}$$

The difference between our work and the classifying algorithm in [24] is that, as shown in Equations 10 and 11, all K-neighbors rather than a subset of the K neighbors contribute to the belief that the test sample belongs to class l_q, making the classification more tolerable to training/test data variations.

A global mass function is further defined by considering all possible classes for the test data sample f_v in estimating the belief value of the sample belonging to class l_q. Hence, the global mass function $m_v=m_v(l_1)\oplus\cdots m_v(l_q)\cdots\oplus m_v(l_P)$ of the test sample belonging to class l_q and the whole frame L are modified as:

$$m_v(l_q)=\frac{m_{<q>}(l_q)\prod_{r\neq q}m_{<r>}(L)}{H} \tag{12}$$

And

$$m_v(L) = \frac{\prod_{q=1}^{P} m_{<q>}(L)}{H} \quad (13)$$

Where H is the normalizing factor, given by:

$$\begin{aligned} H &= \sum_{q=1}^{P} m_{<q>}(l_q) \prod_{r \neq q} m_{<r>}(L) + \prod_{q=1}^{P} m_{<q>}(L) \\ &= \sum_{q=1}^{P} \prod_{r \neq q} m_{<r>}(l_q) + (1-P) \prod_{q=1}^{P} m_{<q>}(L) \end{aligned} \quad (14)$$

The belief function *Bel* is widely used to measure the credibility of a hypothesis in classifying a test data sample. One can assign the mass function in Equation 12 to $Bel_v(l_q)$ as the probability of the input data sample x_v belonging to class l_q. In this work, considering the inaccuracy and randomness in test/training data, we will apply the pignistic probability, which includes a measure of plausibility for more tolerance of test/training data inaccuracy in the classification. The pignistic probability *BetP* for an input sample x_v belonging to class l_q is defined as:

$$BetP_v(l_q) = m_v(l_q) + \frac{m_v(L)}{P}, \; q = 1, 2, \cdots, P \quad (15)$$

Unlike regular intrusion detection algorithms in which the incoming data are classified as either the normal data or one of the known attacks based on the maximum likelihood of all known classes, in this work we will introduce a two-stage intrusion detection mechanism to identify the input data as either normal data, one of known attacks or unknown attack. The first stage detection is to identify if the input data are the normal data or an attack based on the pignistic probability of each class hypothesis. If the class type of the maximum pignistic probability is the normal data l_{norm} or the following equation holds:

$$l_{norm}(x_v) = \arg\max_q BetP_v(l_q) \quad (16)$$

The input data connection x_v is considered to be a normal data connection. Since, the classifier is fully and reliably trained with the labelled normal data, if Equation 16 is true, the classification result becomes final and no further test is needed. However, if Equation 16 is false, the input data are either one of the known attacks or a novel attack with unknown features and the following second-stage entropy-based test is needed to determine the attack type of the incoming data.

$$E_v = \sum_{q=1}^{P} BetP_v(l_q) \log_2 \frac{1}{BetP_v(l_q)} \leq \mu \quad (17)$$

Where μ is predetermined threshold between (0, 1). If the hypothesis in Equation 17 is true, i.e., the entropy of the generated pignistic probabilities is relatively small, the input data connection x_v is strongly correlated to one of known attacks. Therefore, x_v is considered to be one of the known attacks and its class index q^* in the class set is given by:

$$q^*(x_v) = \arg\max_q BetP_v(l_q) \quad (18)$$

However, if Equation 17 is not true, the classification result is not credible and the input data are an unknown attack. The decision tree of the two-stage fuzzy classifier for known and unknown intrusion detection is shown in Figure 2.

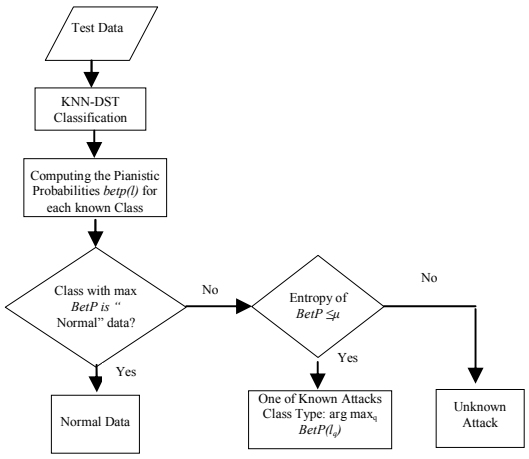

Figure 2. The decision tree of the proposed two-stage fuzzy classifier for unknown attack detection.

3. Experimental Results

We used DARPA KDD99 [16] intrusion detection evaluation dataset as a part of the benchmark for evaluating the performance of the proposed IDS in detecting known intrusions. In addition, we have generated an unknown attack dataset from Software Wireshark [5] to test the performance of the new algorithm for detecting unknown attacks. There are three subset components in the KDD benchmark dataset that includes the whole KDD, 10% KDD and Corrected KDD datasets. The 10% KDD dataset is a concise version of the whole KDD with 22 attack types and more instances of attacks than the normal connections. The corrected KDD dataset contains 14 more attacks with different statistical attack distributions from those of the other two datasets.

3.1. Dataset Selection and Pre-Processing

In this experimental work, the 10% KDD dataset were used as the training dataset to train the FCM algorithm and finalize the optimal feature space centers of the known classes. Subsequently, the two-stage fuzzy classifier is built based on the training results. Finally the corrected KDD 99 dataset and the unknown attack data generated from Wireshark are employed to evaluate the performance of our new classifier. Besides the normal data class, the KDD 99 datasets contain four types of known attacks, including the Denial of Service attacks (DoS), the User to Root attacks (U2R), the Remote to Local attacks (R2L) and the Probing attacks (Probe). However, even for the concise 10%KDD dataset, as its attack distribution statistics, the data sizes are still overwhelmingly large for practical training and test applications. In addition, the data records in KDD 99 for different attacks are not equally represented and those in the same class are

unbalanced for training dataset (10%KDD) and the test dataset (Corrected KDD). For instance, the percentages of U2R and R2L attacks in the training dataset are 0.105% and 2.279%, respectively, while those in the testing dataset are 0.733% and 5.204%. The discrepancies in data sizes and class distributions could make the classification results unreliable. Furthermore, in the original KDD datasets, there are invalid and duplicated data records that need to be pre-processed and removed for algorithm training and testing. Therefore, the cleaned and rebalanced KDD 99 datasets with a reduced and manageable size have been created for our experimental work by randomly sampling the 10% KDD and corrected KDD datasets, in which the redundant and invalid data records are removed. The statistics of the size-reduced KDD datasets we generated for this work are listed in Table 1. In addition to the data records generated from the KDD datasets, we created additional 5074 data records containing unknown attacks using a self-built lab networking system and the wire shark software. The types of the unknown attacks are summarized in Table 2 and they include some of the recently created Internet viruses including adware, spyware and their variants. The features of the unknown attacks may or may not be represented by those of the known attacks in KDD datasets. The unknown attack data records are used to test the proposed classifier and evaluate its performance in detecting unknown intrusions without any training.

Table 1. Reduced training and testing datasets UA used in this work.

Data Class	Total	Normal	DoS	Probe	U2R	R2L	UA
Training Sets	145,585	87,831	54,572	2,131	52	999	0
Testing Sets	51,041	47,913	23,568	2,682	215	2913	5074

Table 2. UA used in testing datasets or this work.

UA Type	Number of Connections	Percentage
Botnet	1988	39.18%
Adware	875	17.24%
Spyware	869	17.13%
Backdoor	412	8.12%
Hijacker	275	5.42%
Trackware	181	3.57%
Downloader	187	3.69%
Trojan	287	5.66%
All UA Types	5074	100%

3.2. Dataset Features for Classification

For the KDD datasets, there are 41 different features for each data connection that are used by the classifiers for intrusion detection. Those features are listed in KDD99 datasets. The features can be grouped into four categories: Basic features that derived from packet headers; content features that are obtained from the payload of original packets; time-based traffic features that are used to capture the properties of the data connections over a 2second temporal widow; and host-based traffic features that are used to characterize the connection data properties based on a window of 100 connections. The feature space of each connection in the KDD99 datasets and the unknown attack dataset we generated is composed of 41 feature components. For the KDD training data and unknown attack dataset, there is additional labelling information indicating which type of class the connection belongs to.

3.3. Performance Evaluation

To minimize the variations of our experimental results, we randomly divide both our training and test datasets into 10 subsets of equal sizes, and then we apply the new classifier to each pair of 10 training-testing subsets to evaluate its performance. We measure the performances of classifiers based on False Positive Rate (FPR), Detection Rate (DR) and Overall Error Rate (OER). FPR, DR and OER for detecting one of the known intrusions or an unknown attack (Class l) from a batch of validation connections are defined as follows:

$$FPR_l = \frac{FP_l}{FP_l + TN_l} \tag{19}$$

Where FP_l is the number of the connections that are incorrectly classified as class l and TN_l is the number of the connections that are correctly classified as a class other than class l.

$$DR_l = \frac{TP_l}{TP_l + FN_l} \tag{20}$$

Where TP_l is the number of the connections that are correctly classified as class l and FN_l is the number of the connections that are incorrectly classified as a class other than class l.

$$OER_l = \frac{FP_l + FN_l}{TP_l + TN_l + FP_l + FN_l} \tag{21}$$

To compare the performance of the new classifying algorithm with those of the existing classifiers, we also, apply several popular classifiers including the basic kNN, evidence-theoretic kNN, naïve Bayes and neural network classifiers [3, 26] to the same dataset used by the proposed classifier. The classification results are displayed with the receiver operating characteristics ROC plots, i.e., DR vs. FPR for all classifiers. Specifically, ROC plots are shown in Figures 2, 3, 4, 5 and 6, respectively, for detecting DoS, Probe, U2R, U2L and Unknown Attacks (UA) by using existing classifiers as well as the new classifier. It is found that the new fuzzy DST classifier we are proposing almost outperforms all other existing classifiers by achieving higher DR and lower FDR for all known and unknown attacks. Since, DoS and Probe attacks usually reveal a sequential pattern that is different from normal connections, they can be relatively easily be differentiated from normal data records. However, U2R and R2L attacks do not

possess a similar sequential pattern and they are embedded in the data portions of the packets and normally only appear in a single connection. Therefore, the detection of U2R and R2L attacks from normal connections is more challenging than identifying DoS and Probe attacks; the detection rates of U2R and R2L intrusions with existing classifiers have been mostly unsatisfactory. However, using the new classifier, as shown in Figures 3, 4, 5, 6 and 7, the detection rates of U2R and R2L attacks are significantly improved.

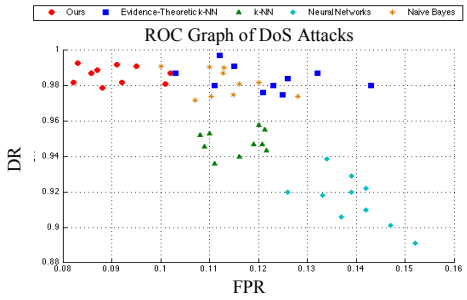

Figure 3. ROC plot of detecting "DoS" attacks using the new classifier and various existing classifiers.

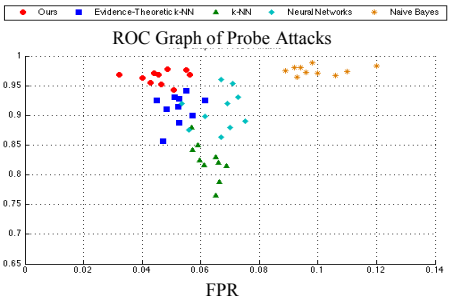

Figure 4. ROC plot of detecting "Probe" attacks using the new classifier and various existing classifiers.

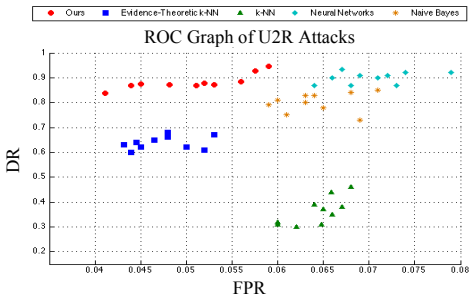

Figure 5. ROC plot of detecting U2R attacks using the new classifier and various existing classifiers.

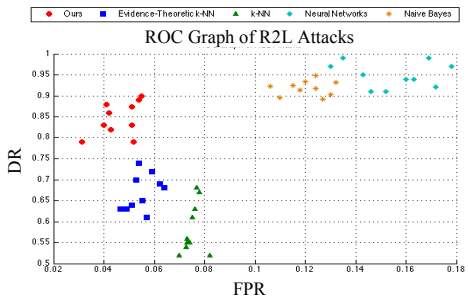

Figure 6. ROC plot of detecting R2L attacks using the new classifier and various existing classifiers.

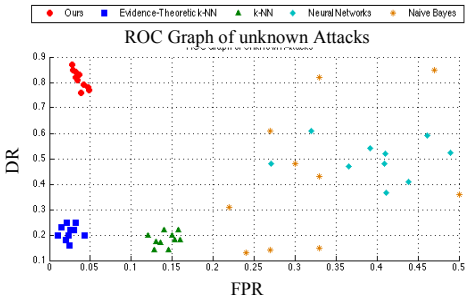

Figure 7. ROC plot of detecting unknown attacks using the new classifier and various existing classifiers.

Table 3 lists the OER in detecting different known and unknown attacks by using our new classifier and other existing classifiers. OER, as defined in Equation 20, includes the effects of both DR and FPR for a classifier, therefore, is a better indicator of classification performance. The results in Table 5 show that the OER of the new algorithm is significantly lower than those of other existing classifiers in detecting the known and unknown intrusions. In implementing the two-stage fuzzy kNN-DST classifier we choose the predetermined threshold μ to be 0.85 in the second-stage entropy-based classification. The second-stage detection is used to determine if an attack is unknown or one of known attacks, and it is only needed if the first-stage detection result is an attack. The experimental results demonstrate that the new classifier is effective in identifying unknown attacks as well as detecting typical known attacks from normal data traffic.

Table 3. The overall detection error rates of the new method and other IDS for detecting various known and unknown attacks (Evidence-Theoretic k-NN (ET k-NN); Neural Networks (NN); Naïve Bayes (NB)).

Class	Ours Method	ET kNN	kNN	NN	NB
DOS	5.22%	6.87%	11.18%	8.39%	6.57%
Probe	3.90%	7.01%	12.07%	7.80%	6.21%
U2R	8.12%	20.47%	35.06%	8.33%	13.13%
R2L	9.98%	19.31%	24.60%	10.29%	10.45%
UA	11.25%	40.68%	47.94%	44.88%	44.90%

4. Conclusions

An innovative two-stage fuzzy kNN DST classifier has been developed for effective detection of unknown intrusions and the variants of known intrusions. The new algorithm overcomes the rigid requirement of feature vector similarity between the training data and the test data in current IDS by introducing fuzziness, "soft" distance-based neighbouring concepts and the DST-based evidence fusion method into the learning and classification schemes. Furthermore, the two-stage entropy-based classification approach is employed to identify unknown attack in the incoming connections without any pre-training data or labelled information for the attack. The robustness and effectiveness of the new approach are demonstrated by the application results of the new classifier to the traditional KDD99 intrusion data and the newly simulated data containing both known and unknown attacks. The experimental

results also, show that the new classifier outperforms the existing classification algorithms in identifying known and unknown attacks from network traffic.

References

[1] Altincay H., "Ensembling Evidential K-Nearest Neighbor Classifiers Through Multi-Modal Perturbation," *Applied Soft Computing*, vol. 7, no. 3, pp. 1072-1083, 2007.

[2] Ambwani T., "Multi Class Support Vector Machine Implementation to Intrusion Detection," *in Proceedings of the International Joint Conference on Neural Networks*, pp. 2300-2305, 2003.

[3] Berardi V. and Zhang G., "The Effect of Misclassification Costs on Neural Network Classifiers," *Decision Science*, vol. 30, no. 3, pp. 659-682, 1999.

[4] Cherkassky V., "The Nature of Statistical Learning Theory," *IEEE Transactions on Neural Networks*, vol. 8, no. 6, pp. 1564-1564, 1997.

[5] De Biasi M., Snickars C., Landernas K., and Isaksson A., "Simulation of Process Control With Wireless HART Networks Subject to Packet Losses," *in Proceedings of IEEE CASE*, Arlington, pp. 548-553, 2008.

[6] De Castro L. and Von Zuben F., "Learning And Optimization using the Clonal Selection Principle," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 3, pp. 239-251, 2002.

[7] Denatious D. and John A., "Survey on Data Mining Techniques to Enhance Intrusion Detection," *in Proceedings of International Conference on Computer Communication and Informatics*, Coimbatore, pp. 1-5, 2012.

[8] Denoeux T., "A K-Nearest Neighbor Classification Rule Based on Dempster-Shafer Theory," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 25, no. 5, pp. 804-813, 1995.

[9] Golovko V. and Kochurko P., "Intrusion Recognition Using Neural Networks," *in Proceedings of Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, Sofia, pp. 108-111, 2005.

[10] Goyal M., Aggarwal A., and Jain N., "Effect of Change in Rate of Genetic Algorithm Operator on Composition of Signatures for Misuse Intrusion Detection System," *in Proceedings of the 2nd IEEE International Conference on Parallel Distributed and Grid Computing*, Solan, pp. 669-672, 2012.

[11] Guangyou Y., "A Modified Particle Swarm Optimizer Algorithm," *in Proceedings of the 8th International Conference on Electronic Measurement and Instruments*, Xi'an, pp. 675-679, 2007.

[12] Han W., Xiong W., Xiao Y., Ellabidy M., Vasilakos A., and Xiong N., "A Class of Non-Statistical Traffic Anomaly Detection in Complex Network Systems," *in Proceedings of the 32nd International Conference on Distributed Computing Systems Workshops*, Macau, pp. 640-646, 2012.

[13] Hofmann A., Schmitz C., and Sick B., "Rule Extraction from Neural Networks for Intrusion Detection in Computer Networks," *in Proceedings of International Conference on Systems, Man and Cybernetics*, pp. 1259-1265, 2003.

[14] Hwang W. and Wen K., "Fast kNN Classification Algorithm Based on Partial Distance Search," *Electronics Letters*, vol. 34, no. 21, pp. 2062-2063, 1998.

[15] Jiang L., Zhang H., and Cai Z., "A Novel Bayes Model: Hidden Naïve Bayes," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 10, pp. 1361-1371, 2009.

[16] KDD Cup 1999 Data., available at: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, last visited 2013

[17] Keller J., Gray M., and Givens J., "A Fuzzy K-Nearest Neighbor Algorithm," *IEEE Transaction System Man, and Cybernetics*, vol. 15, no. 4, pp. 580-585, 1985.

[18] Kim D., Nguyen H., and Park J., "Genetic Algorithm to Improve SVM based Network Intrusion Detection System," *in Proceedings of the 19th International Conference on Advanced Information Networking and Applications*, pp. 155-158, 2005.

[19] Kotsiantis S., "Integrating Global and Local Application of Naive Bayes Classifier," *International Arab Journal of Information Technology*, vol. 11, no. 3, pp. 300-307, 2014.

[20] Lee S. and Heinbuch D., "Training a Neural-Network based Intrusion Detector to Recognize Novel Attacks," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 31, no. 4, pp. 294-299, 2001.

[21] Li H., Wen G., and Cai X., "Subspace Local Mean Evidence Classifier," *Journal of Computational Information System*, vol. 8, pp. 8985-8992, 2012.

[22] Li J., Zhang G., and Gu G., "The Research And Implementation of Intelligent Intrusion Detection System Based on Artificial Neural Network," *in Proceedings of International Conference on Machine Learning and Cybernetics*, pp. 3178-3182, 2004.

[23] Liu Z., Dezert J., Mercier G., and Pan Q., "Belief C-Means: An EXTENSION of Fuzzy C-Means Algorithm in Belief Function Framework",

Pattern Recognition Letters, vol. 33, no. 3, pp. 291-300, 2012.

[24] Mitchell R. and Chen I., "Effect of Intrusion Detection and Response on Reliability of Cyber Physical Systems," *IEEE Transactions on Reliability*, vol. 62, no. 1, pp. 199-210, 2013.

[25] Mukkamala S., Janoski G., and Sung A., "Intrusion detection using neural networks and Support Vector Machines," *in Proceedings of International Joint Conference on Neural Networks*, Honolulu, pp. 1702-1707, 2002.

[26] Roomi M. and Saranya S., "Bayesian Classification of Fabrics using Binary Co-Occurrence Matrix," *International Journal of Information Science and Techniques*, vol. 2, no. 2, pp. 1-9, 2012.

[27] Setnes M. and Babuska R., "Fuzzy relational Classifier Trained by Fuzzy Clustering," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics,* vol. 29, no. 5, pp. 619-625, 1999.

[28] Shafer G., *A Mathematical Theory of Evidence*, Princeton University Press, 1979.

[29] Shazzad K. and Park J., "Optimization of Intrusion Detection through Fast Hybrid Feature Selection," *in Proceedings of the 6th International Conference on Parallel and Distributed Computing Applications and Technologies*, pp. 264-267, 2005.

[30] Singh J., Kaur L., and Gupta S., "A Cross-Layer Based Intrusion Detection Technique for Wireless Networks," *the International Arab Journal of Information Technology*, vol. 9, no. 3, pp. 201-207, 2012.

[31] Tadjudin S. and Landgrebe D., "A Decision Tree Classifier Design For High-Dimensional Data With Limited Training Samples," *in Proceedings of International Geoscience and Remote Sensing Symposium*, Lincoln, pp. 790-792, 1996.

[32] Yager R., "Generalized Probabilities of Fuzzy Events from Fuzzy Belief Structures," *Information Sciences*, vol. 28, no. 1, pp. 45-62, 1982.

Xueyan Jing currently is a PhD candidate in the Department of Electrical and Computer Engineering at Florida International University, USA. Her research interests include building next-generation tools using data-mining algorithms to detect stealth intruders in networking systems and applications of machine learning techniques in securing wireless networks.

Yingtao Bi is a research assistant professor in the Feinberg School of Medicine at Northwestern University, USA. His recent research focuses mainly on developing data-mining algorithms and informatics approaches for solving problems in biology and medicine for cancer treatment.

Hai Deng received the PhD degree in Electrical Engineering from University of Texas at Austin in 2000. He has been with Department of Electrical and Computer Engineering at Florida International University, USA since 2009. His research interests include radar sensor networks, MIMO radar, biomedical signal processing and VLSI design.