Anomaly Detection Based on Unsupervised Niche Clustering with Application to Network Intrusion Detection

Elizabeth Leon
Dept. of Electrical & Computer Eng.
The University of Memphis
Email: eleon@memphis.edu

Olfa Nasraoui
Dept. of Electrical & Computer Eng.
The University of Memphis
Email: onasraou@memphis.edu

Jonatan Gomez
Division of Computer Sciences
The University of Memphis and
Universidad Nacional de Colombia
Email: jgomez@memphis.edu

***Abstract*— We present a new approach to anomaly detection based on the Unsupervised Niche Clustering (UNC). The UNC is a genetic niching technique for clustering that can handle noise, and is able to determine the number of clusters automatically. The UNC uses the normal samples for generating a profile of the normal space (clusters). Each cluster can later be characterized by a fuzzy membership function that follows a Gaussian shape defined by the evolved cluster centers and radii. The set of memberships are aggregated using a max-or fuzzy operator in order to determine the normalcy level of a data sample. Experiments on synthetic and real data sets, including a network intrusion detection data set, are performed and some results are analyzed and reported.**

I. Introduction

The anomaly detection problem can be considered as a two-class classification problem (normal versus abnormal) where samples of only one class (normal class) are used for training. Basically there are three different approaches for anomaly detection:

1) *Negative Characterizatio*n: The normal samples are used for building a model of the abnormal space; for example, by generating a set of rules (detectors) that can recognize abnormal patterns [1], [2], [3].
2) *Positive characterization*: The normal samples are used for building a model of the normal space, for example a set of rules that define the normal space [4], [5], [6], [7], [8], [9], [10].
3) *Artificial anomaly generation*: The normal samples are used for generating artificial anomalies, samples that belong to the positive class, and then a classifier learning technique is used for generating a classifier that discriminates between the two classes [11], [12].

Clustering techniques have been applied successfully to the anomaly detection problem, where it is applied to the normal samples to generate a set of clusters that will represent the normal class. Clustering is an unsupervised learning technique of data mining that takes unlabeled data points and tries to group them according to their similarity: points assigned to the same cluster have high similarity, while the similarity between points assigned to different clusters is low [13]. When a clustering algorithm deals with noisy information, the algorithm is called robust [14], [15], [16], and when the number of clusters is determined automatically, it is usually called unsupervised [17]. The Unsupervised Niche Clustering (**UNC**) is a robust and unsupervised clustering algorithm that uses an evolutionary algorithm with a niching strategy [18], [19], [17]. While the evolutionary algorithm allows to find the clusters using a robust density fitness function, the niching technique allows it to create and maintain the niches (candidate clusters). Since UNC is based on genetic optimization, it is much less prone to suboptimal solutions than traditional techniques.

In this paper, we combine the UNC with fuzzy sets theory for anomaly detection and apply it to network intrusion detection. We associate to each cluster generated by the UNC a membership function that follows a Gaussian shape using the evolved cluster center and radius. Such cluster membership functions will define the normalcy level of a data sample. The rest of this paper is organized as follows. In section 2, we give a brief description of the UNC algorithm. In section 3, we introduce a cluster characterization for anomaly detection. In section 4, we present some experimental results using a synthetic data set with different rates of noise in order to determine the robustness of the approach, and with real data sets for anomaly detection including a network intrusion detection data set. Finally, we present our conclusions in section 4.

II. Unsupervised Niche Clustering (UNC)

Unsupervised Niche Clustering is a clustering technique that can handle noise, is able to determine the number of clusters automatically, and has shown good performance in detecting the number of clusters in noisy 2-D data sets and segmenting real color images [18], [19], [17]. In UNC, the clustering problem is reformulated by modifying the objective from searching the solution space for C clusters to searching this space for any one cluster. In order to do this, UNC locates and maintains dense areas (clusters) in the solution space using an evolutionary algorithm (**EA**) and a niching technique

center	scale
$c_i = (c_{i,1}, c_{i,2}, ..., c_{i,n})$	σ_i^2

Fig. 1. Representation of the i^{th} individual

[20], [21]. As in nature, niches in our context correspond to different subspaces of the environment (clusters) that can support different types of life (data samples).

A. Encoding Scheme

An individual represents a candidate cluster determined by its center, an *n*-dimensional vector with *n* being the dimension of the data samples, and a robust measure of its scale (or dispersion) σ^2, as illustred in figure 1.

For the i^{th} candidate cluster, σ_i^2 is crucial for determining its boundaries. In fact, for spherical Gaussian clusters, the radius can be given by $K\sigma_i^2$, where K is determined by $\chi^2_{n,0.995}$ according to the data set dimensionality.

B. Genetic Operators and Scale Update

While the center of the cluster is evolved using the EA, the scale is updated using an iterative hill-climbing procedure. In each generation of the EA, two genetic operators (crossover and mutation) are applied over the center for generating the offspring population. The one-point crossover is performed for each dimension of the n-dimensional vector (center) and each bit of the vector can be mutated accordingly to some mutation probability.

Each child inherits the scale of its closest parent. Both, parent population and offspring population update their scales using equation (1) ,

$$\sigma_i^2 = \frac{\sum_{j=1}^{N} w_{ij} d_{ij}^2}{\sum_{j=1}^{N} w_{ij}} \tag{1}$$

where $w_{ij} = \exp\left(-\frac{d_{ij}^2}{2\sigma_i^2}\right)$ is a robust weight that measures how typical data point x_j is in the i^{th} cluster, and d_{ij}^2 is the distance from data point x_j to cluster center c_i, and N is the number of data points. σ_i^2 is updated using w_{ij}, that is computed using the previous value of σ_i^2. Equation (1) maximizes the fitness value given by equation (2) for the i^{th} cluster.

C. Fitness Function

The fitness value, f_i, for the i^{th} candidate center location c_i, is defined as the density of a hypothetical cluster at that location, defined as

$$f_i = \frac{\sum_{j=1}^{N} w_{ij}}{\sigma_i^2}, \tag{2}$$

Algorithm 1 Deterministic Crowding

DETERMINISTICCROWDING(N)
1. P = InitPopulation(N)
2. **for** k=1 to MAXITER **do**
3. P = Shuffle(P), $P' = \{\}$
4. **for** i=1 to $\frac{N}{2}$ **do**
5. $\{c_1, c_2\} = xover\,(P_i, P_{i+1})$
6. $\{c_1', c_2'\}$ =OptionalMutation$\{c_1, c_2\}$
7. **if** $d\,(P_i,\, c_1') + d\,(P_{i+1},\, c_2') \leq d\,(P_i,\, c_2') + d\,(P_{i+1},\, c_1')$**then**
8. **if** $f\,(P_i) < f\,(c_1')$**then** $P' = P' \cup \{c_1'\}$
else $P' = P' \cup \{P_i\}$
9. **if** $f\,(P_{i+1}) < f\,(c_2')$**then** $P' = P' \cup \{c_2'\}$
else $P' = P' \cup \{P_{i+1}\}$
10. **else**
11. **if** $f\,(P_i) < f\,(c_2')$**then** $P' = P' \cup \{c_2'\}$
else $P' = P' \cup \{P_i\}$
12. **if** $f\,(P_{i+1}) < f\,(c_1')$**then** $P' = P' \cup \{c_1'\}$
else $P' = P' \cup \{P_{i+1}\}$
13. **return** P'

D. Niching

UNC uses Deterministic Crowding as the niching technique for creating and maintaining niches [21]. Algorithm 1 presents the basic Deterministic Crowding algorithm.

In UNC, an additional restriction on the deterministic crowding selection (line 7) is introduced for restricting the mating between members of different niches:

IF *dist*$(P_i, P_{i+1}) >$ *Kmax*(σ_i, σ_{i+1}) ***and*** $f(P_i) > f_{min}$ ***and*** $f(P_{i+1}) > f_{min}$ ***THEN*** *Restrict mating.*

This condition allows the EA to reduce the crossover interaction problem of the deterministic crowding [22].

E. Extraction and Refinement

The final cluster center candidates are the individuals in the final population (after convergence) with fitness values greater than a certain threshold. From this set of candidates centers, the best individual per niche will determine the final cluster's center.

To increase the accuracy of the solution (set of cluster prototypes) provided by the genetic optimization, a local refinement procedure can be performed in each cluster independently of the other clusters[17]. The refinement consists of applying an alternative optimization of the centers of each cluster using a robust estimator. The data set is partitioned into C clusters (number of prototypes predicted by the genetic optimization), in such way that each dimensional vector is assigned to the closest prototype. Subsequenly, the i^{th}cluster is given by $\mathcal{X}_i = \left\{\mathbf{x}_{(k)} \in \mathcal{X} \mid d_{ik}^2 < d_{jk}^2\ \forall j \neq i\right\}$, for $1 \leq i \leq C$.

The Maximal Density Estimator (**MDE**) [23] is a robust estimator used by UNC for refining the centers and scaling accurately and efficiently based on a local iterative search using equations (3) and (4).

$$c_i = \frac{\sum_{\mathbf{x}_{(j)} \in X_i} w_{ij} \mathbf{x}_j}{\sum_{\mathbf{x}_{(j)} \in X_i} w_{ij}} \tag{3}$$

$$\sigma_i^2 = \frac{\sum_{\mathbf{x}_{(j)} \in X_i} w_{ij} d_{ij}^4}{3 \sum_{\mathbf{x}_{(j)} \in X_i} w_{ij} d_{ij}^2} \quad (4)$$

III. Cluster Characterization

After the clusters are generated (centers and scale), a crisp characterization is used in order to determine the cluster that a sample is assigned to: A new sample x is assigned to the cluster c_i if x falls inside the boundaries of c_i, and the distance from x to the center of c_i is the minimum distance among all the clusters that the sample belongs to. A data point falls into the cluster c_i if the distance of the data point to the cluster center is less than or equal to the cluster radius. As mentioned before, the radius of cluster c_i is given by $K\sigma_i^2$, where K is determined by $\chi^2_{n,0.995}$ according to the data set dimensionality (n)[1].

Since the basic assumption of the UNC is that each cluster follows a Gaussian distribution in terms of distances to the cluster center, it is possible to characterize each predicted cluster using a fuzzy membership function that follows a Gaussian shape. A membership function for c_i is given by equation (5).

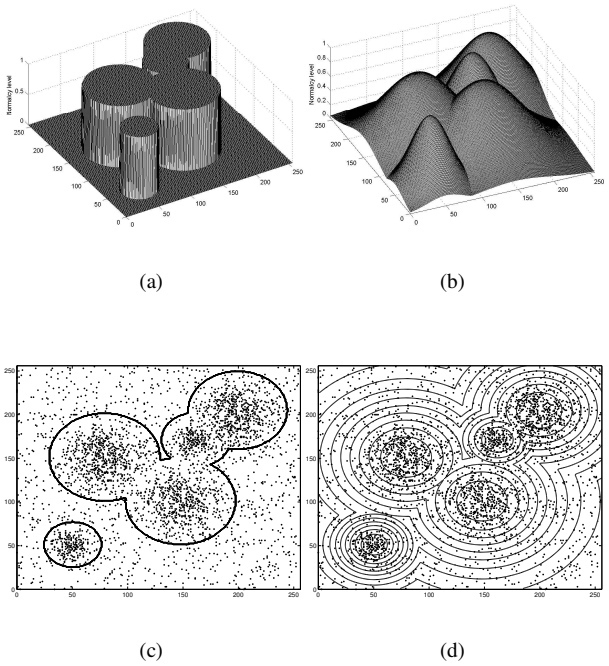
$$\mu_i(x) = exp\left(-\frac{d(x, c_i)^2}{2\sigma_i^2}\right) \quad (5)$$

$$\mu_{normal}(x) = max\left\{\mu_i(x) | \forall i = 1, 2, ..c\right\} \quad (6)$$

In this way, a sample x belongs to each cluster with a certain degree of membership. Several defuzzification techniques, like **MAX**-defuzzification, can be used for determining the final cluster that a sample belongs to. In the MAX-defuzzification a sample is classified in the cluster with the highest membership value. For the anomaly detection problem, it is important to determine if a data sample is assigned to some cluster or not (if it is normal or abnormal), but it is not important to determine to which cluster the data sample was assigned. Therefore, the normal class is defined by the union set of all the clusters generated by the clustering algorithm. In this paper, the normal class is defined as a fuzzy set using the *max-OR* fuzzy operator, see equation (6). A sample x is considered normal with a $\mu_{normal}(x)$ degree. Figure 2 compares the crisp and fuzzy characterization of the clusters generated by UNC for a synthetic data set.

Because a fuzzy characterization of the normal class is generated, we use a thresholding mechanism for obtaining the final crisp characterization of the normal class: If the fuzzy value is greater than a certain threshold θ, x is considered normal, otherwise, it is considered abnormal. In this way, θ is a suitable threshold that takes values in the $[0, 1]$ interval that can be chosen according to the accuracy required.

[1]Although UNC is able to find spherical and ellipsoidal clusters, in this paper we restricted the UNC to find spherical clusters only.



(a) (b) (c) (d)

Fig. 2. Membership functions for a typical run of UNC+MDE for data set with 35% noise contamination (a) crisp, (b) fuzzy, (c) crisp contours, (d) fuzzy contours

IV. Experimentation

Tests were conducted using synthetic and real data sets in order to determine the performance of the proposed approach for anomaly detection. There are two elements that define the accuracy of an anomaly detection approach: The detection rate (**DR**) which is the percentage of abnormal samples correctly classified (considered abnormal) and the false alarm rate (**FA**) which is the percentage of the normal samples incorrectly classified (considered abnormal). The Receiver Operating Characteristic (**ROC**) analysis [24] can be applied to an anomaly classifier to evaluate its performance [25]. In the ROC analysis, for anomaly classifier systems that produce a continuous output with respect to some parameter a, the coordinate point $(FP, TP)_a$ is plotted in the cartesian coordinate system. If the ROC curve of classifier A dominates the ROC curve of classifier B then classifier A is considered better than classifier B [24]. Therefore, we plot the ROC curve for the results obtained using the proposed approach in order to determine its performance with respect to the threshold θ.

A. Synthetic data set

We use a five-cluster synthetic data set with 35% noise contamination. It has five Gaussian clusters with centers and radii as shown in table I. The size of the data set is 3230. It has 2078 normal samples (samples that belongs to any cluster) and 1152 noise samples. The purpose of using this synthetic data set is to detect noise as anomalies and show the robustness of the proposed approach to noise.

Centers	(50,50)	(150,100)	(160,170)	(200,200)	(80,150)
Radius	20	40	25	40	40

TABLE I

GENERATING PARAMETERS FOR SYNTHETIC DATA SET WITH FIVE CLUSTERS

Abbreviation	**Approach**
UNC	Unsupervised Niche Clustering
MDE	Maximal Density Estimator
AD	Anomaly Detection using UNC
AD+	Anomaly Detection using UNC+MDE
FAD	Fuzzy Anomaly Detection using UNC
FAD+	Fuzzy Anomaly Detection using UNC+MDE

TABLE II

NOTATION USED FOR OUR APPROACHS

1) Experimental settings: A 10-fold cross-validation was applied. Noisy samples were injected into the training data set at different rates: 0%, 5%, 10% and 20%. The UNC was run with the parameters shown in Table III, with and without MDE refinement. The reported accuracy is the average of the ten folds. We used the notation presented in table II for different versions of UNC testing.

Population	80	Weight threshold	0.3
Generations	30	Fitness extraction	0.25
Mutation probability	0.001	Fitness crossover	0.6
Mutation crossover	1.0	Sigma Factor	13.8
MDE iterations	5		

TABLE III

UNC PARAMETERS

2) Analysis of the Results: Figure 3 shows the ROC curves obtained by the proposed approach with different percentages of noise, using fuzzy or crisp characterization with refinement and without it.

Two conclusions can be drawn from these results. First, the performance reached using fuzzy (FAD and FAD+) is superior to the performance reached using only crisp (AD and AD+). It is possible to increase the DR by slightly increasing the FA using fuzzy analysis regardless of the amount of noise injected in the training data set. Second, when MDE [23] was applied (AD+ and FAD+) the performance reached was higher than without applying it (AD and FAD). it increases the detection rate by an amount of 20% to 30% while keeping the FA low. Figure 3 (e) shows the ROC curves generated by FAD+ for the different rates of noise injected in the training set. As shown, the performance of the proposed approach is almost the same in all the cases. The variation between the ROC curves is less than 3% in DR for the same FA. Table IV shows the performance reached by FAD+ when the threshold θ was set to 0.7. As expected the performance of the proposed approach is similar among the different percentages of noise injected. These results indicate that our approach is robust to noise.

	0%	**5%**	**10%**	**20%**
DR (%)	96.32	96.39	96.13	95.25
FA (%)	7.75	8.61	7.60	9.09
Accuracy (%)	95.70	95.63	95.56	95.25

TABLE IV

ACCURACY FOR THE SYNTHETIC DATA SET USING FAD+, FOR DIFFERENT LEVELS OF NOISE.

B. Machine learning data sets

We use the Wisconsin Breast Cancer and Indian Diabetes (Pima) data sets to determine the performance of our approach on real data sets (see Table V).

Data Set	**Size**	**Normal**	**Abnormal**	**Dimensions**
Breast cancer	699	458	241	9
Pima	768	500	268	8

TABLE V

REAL MEDICAL DATA SETS

1) Experimental settings: A 10-folding cross validation, as explained in the previous section, was applied. The accuracy of the trained classifier was calculated over the average of these ten tests.

2) Analysis of the Results: Figure 4 shows the ROC curves generated by FAD and FAD+ for the Wisconsin breast cancer data set. As expected, the performance of FAD+ is higher than the performance of the other three methods. Moreover, the fuzzy versions outperform the crisp versions while the refined versions outperform the unrefined ones.

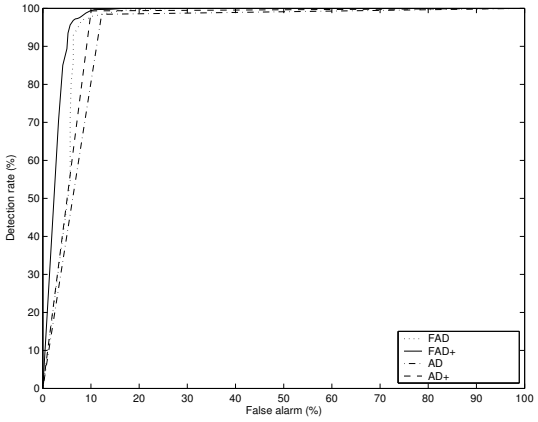
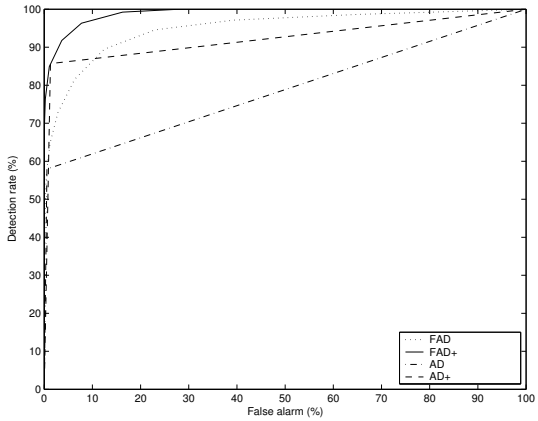


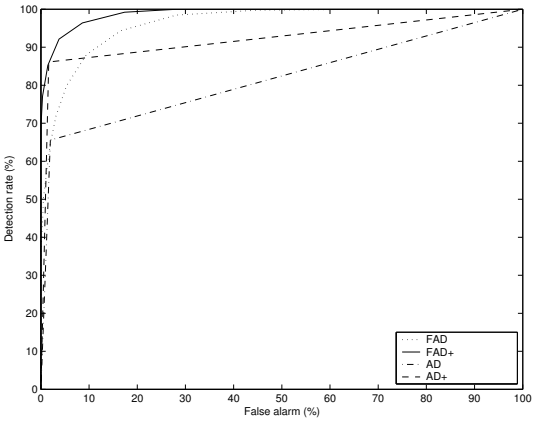
Fig. 4. ROC curve for the Wisconsin breast-cancer

The accuracy reached by FAD+ on the real medical data sets is shown in Table VI(a) and compared against some results previously reported in the literature in table VI(b)[2]. As shown,

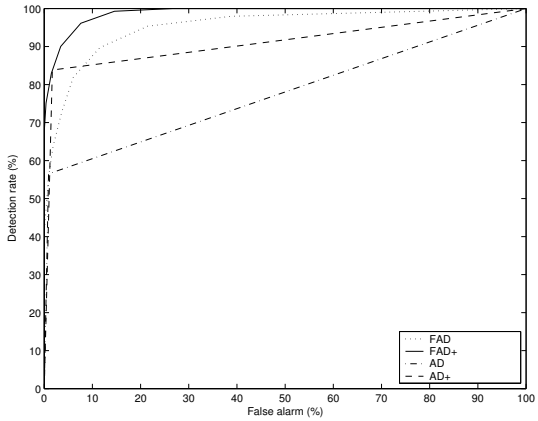
[2]Results reported for QDA, LDA, C4.5, kNN, SSV and FSM taken from [26]. Results for WM, GIL, ABD, and ABA taken from [27].



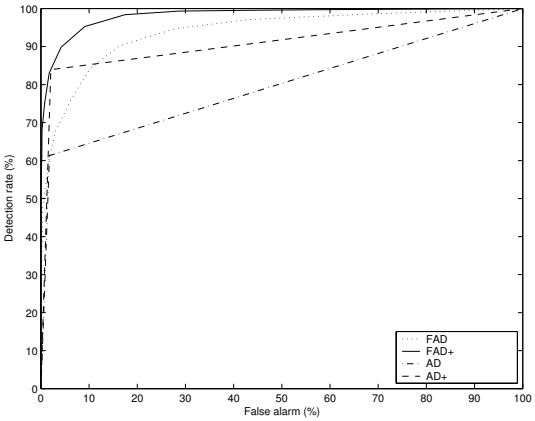
(a)



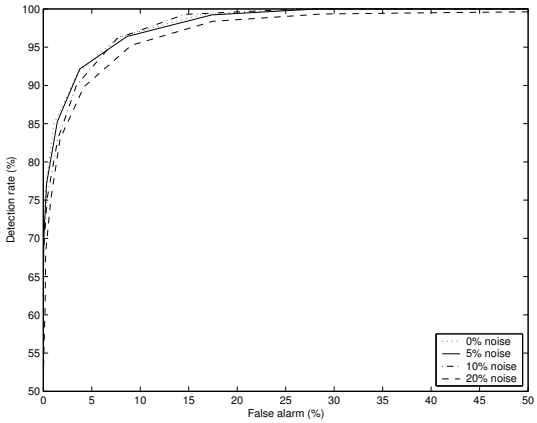
(b)



(c)



(d)



(e)

Fig. 3. ROC for synthetic data sets: (a) 0%, noise, (b) 5% noise, (c)10%, (d) 20%, (e)FAD+ for all noise rates

CLASS	SUB-CLASSES	SAMPLES	%
Normal		95278	19.3%
U2R	buffer_overflow, loadmodule, multihop, perl, rootkit	59	0.01%
R2L	ftp_write, guess_passwd, imap, phf, spy, warezclient, warezmaster	1119	0.23%
DOS	back, land, neptune, pod, smurf, teardrop	391458	79.5%
PRB	ipsweep, nmap, portsweep, satan	4107	0.83%

TABLE VII
CLASSES IN THE 10 % OF THE KDD'CUP 99 DATA SET

our results fare well with existing techniques [3].

	DR %	FA %	Accuracy %	θ
Breast-cancer	97.96	8.08	96.99	0.5
Pima	82.16	46.6	77.63	0.05

(a)

Method	BREAST	PIMA	Statistical Test
FAD+	**96.99**	**77.63**	**10-fold cross-validation**
QDA	94.90	74.80	Leave-one-out
LDA	96.00	77.20	Leave-one-out
C4.5	94.70	73.00	Leave-one-out
kNN	96.90	71.90	Leave-one-out
SSV	96.30	73.70	10-fold cross-validation
FSM	96.90	-	10-fold cross-validation
WM	87.10	71.30	-
GIL	90.10	73.10	-
ABD	96.00	75.90	-
ABA	95.10	74.80	random (50-50)%

(b)

TABLE VI
PERFORMANCE OF FAD+ IN MACHINE LEARNING DATA SETS: (A) ACCURACY, (B) COMPARATIVE PERFORMANCE.

C. Network Intrusion Detection Data Set

The KDDCup'99 data set is a version of the 1998 DARPA intrusion detection evaluation data set prepared and managed by MIT Lincoln Labs. In this data set, 42 attributes (or fields) that usually characterize network traffic behavior compose each record. Some of these attributes are categorical (8) while others are discrete or numerical (34). The total number of records in the 10% data set is 492021. This data set contains 22 different types of attacks that can be classified in four main intrusion classes, as shown in table VII. As can be noticed, the proportion of samples per class is not uniform, for example from class U2R, the number of samples in the training data set is only 59, while from class DOS, the number of samples is 391458.

1) Experimental settings: We used a reduced version of the KDDCup'99 data set that only includes the non-zero numerical features (33). Also, we reduce the dimensionality of this data set using Principal Component Analysis (**PCA**) [13], the data set was reduced to 21 features. PCA is a preprocessing method that can be used to transform an initial data set into a lower dimensionallity set with the most important features. We use 5000 normal samples as training data set, while 40% of the data set (normal and abnormal) was used for testing.

[3]Although all of these results were obtained with different statistical validation methods (leave-one-out, or 10-fold cross-validation) or no statistical validation, the values reported here are indicative of the performance of our proposed approach.

Algorithm	FA%	DR%
FAD+ with PCA	2.20	99.20
FAD+ without PCA	7.84	94.09
RIPPER-AA [11]	2.02	94.26
SMARTSIFTER [28]	-	82.0

(a)

	DOS	PRB	R2L	U2R
DR %	95.9	93.9	98.6	90.9
FA %	1.0	12.2	28.6	20.6

(b)

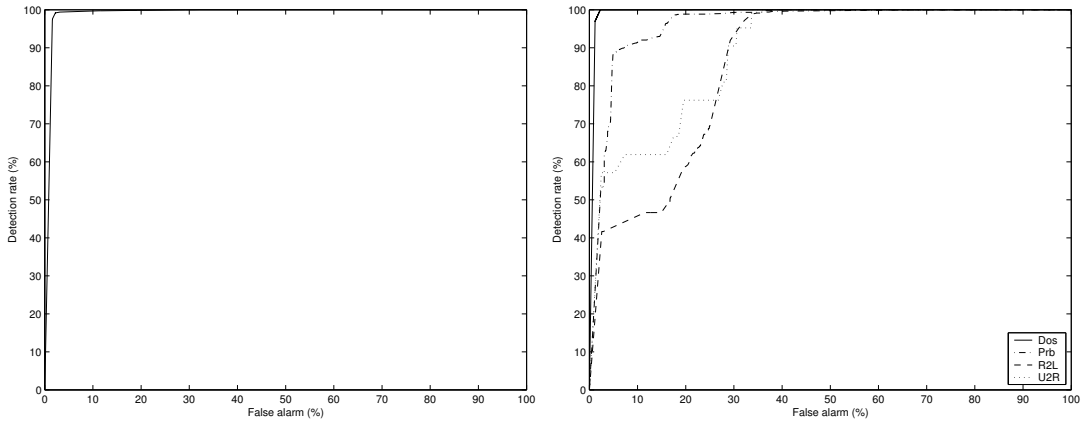
TABLE VIII
PERFORMANCE OF FAD+ IN KDDCUP'99: (A) COMPARISON OF FAD+ WITH PREVIOUSLY REPORTED RESULTS, (B) PERFORMANCE PER ATTACK.

2) Analysis of the Results: The performance reached by FDA+ (with and without PCA) in the KDDCup'99 data sets is reported in Table VIII(a) (rows 1 and 2). As can be noticed, PCA allows the proposed approach to find better results. It is possible that by applying PCA, real normal clusters are transformed into a spherical shape, allowing the proposed approach to produce a better discrimination between the normal and abnormal boundary. Also, Table VIII(a) compares the performance of the proposed approach against some results reported in the literature. FDA+ resulted in DR of 99.2% compared to 94.26% in RIPPER.

Figure 5 shows the ROC curve generated by FAD+ for the PCA-KDDCup'99 data set. Clearly, the performance on this data set is biased by the performance reached in discriminating the DoS attack. However, our approach performs well in the other types of attacks, see Table VIII(b).

V. CONCLUSIONS

In this paper, the UNC algorithm[18], [19], [17] was applied to anomaly detection. A characterization of the model generated by UNC algorithm based on fuzzy set theory was proposed. Several experiments with synthetic and real data sets were performed. The results show that using fuzzy analysis detection increases the performance reached by the UNC (with and without refinement). Also, the MDE refinement strategy[23] improves the quality of the solution. Despite the inherent difficulties in the real data sets, our approach was able to produce comparatively good results. The reduction of the dimensionality for the KDDCup'99 data set, using PCA, reduced the complexity of the data set and further improved the performance of the proposed approach.



(a) (b)

Fig. 5. KDDCup'99 ROC curve using PCA and FAD+: (a) all attacks, (b) per attack

ACKNOWLEDGMENT

This work is supported by National Science Foundation CAREER Award IIS-0133948 to O. Nasraoui.

REFERENCES

[1] S. Forrest, A. Perelson, L. Allen, and R. Cherukury, "Self-nonself discrimination in a computer," in *Proceedings of the IEEE Symp. on research in security and privacy*, 1994.

[2] S. Singh, "Anomaly detection using negative selection based on the r-contiguous matching rule," in *1st International Conference on Artificial Immune Systems(ICARIS)*, pp. 99–106, 2002.

[3] F. Gonzalez and D. Dasgupta, "An immunogenetic technique to detect anomalies in network traffic," in *Proceedings of the genetic and evolutionary compuation conference, GECCO 2002*, pp. 1081–1088, Morgan Kaufman Publishers, 2002.

[4] S. Kumar and E. H. Spafford, "A pattern matching model for misuse intrusion detection," in *In 17th National Computer Security Conference*, pp. 11–21, 1994.

[5] S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff, "A sense of self for Unix processes," in *Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy*, pp. 120–128, IEEE Computer Society Press, 1996.

[6] T. Lane and C. E. Brodley, "An application of machine learning to anomaly detection," in *Proc. 20th NIST-NCSC National Information Systems Security Conference*, pp. 366–380, 1997.

[7] T. Lane and C. E. Brodley, "Sequence matching and learning in anomaly detection for computer security," in *AI Approaches to Fraud Detection and Risk Management* (Fawcett, Haimowitz, Provost, and Stolfo, eds.), pp. 43–49, AAAI Press, 1997.

[8] W. Lee, S. Stolfo, and K. Mok, "Mining Audit Data to Build Intrusion Detection Models," in *International Conference Knowledge Discovery and Data Mining (KDD'98)*, pp. 66–72, 1998.

[9] S. Hofmeyr, A. Somayaji, and S. Forrest, "Intrusion Detection Using Sequences of Systems Call," *Computer Security*, no. 6, pp. 151–180, 1998.

[10] M. Mahoney and P. Chan, "Learning nonstationary models of normal network traffic for detecting novel attacks," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 23–26, 2002.

[11] W. Fan, W. Lee, M. Miller, S. Stolfo, and P. Chan, "Using artificial anomalies to detect unknown and known network intrusions," in *Proceedings of the first IEEE International conference on Data Mining*, 2001.

[12] F. Gonzalez and D. Dasgupta, "Neuro-immune and self-organising map approaches to anomaly detection: A comparison," in *First International Conference on Artificial Immune Systems*, Sept. 2002.

[13] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*. NY: Wiley Interscience, 1973.

[14] R. Krishnapuram and J. M. Keller, "A possibilistic approach to clustering," *IEEE Trans. Fuzzy Syst.*, vol. 1, pp. 98–110, May 1993.

[15] J. M. Jolion, P. Meer, and S. Bataouche, "Robust clustering with applications in computer vision," vol. 13, pp. 791–802, Aug. 1991.

[16] O. Nasraoui and R. Krishnapuram, "Clustering using a genetic fuzzy least median of squares algorithm," in *North American Fuzzy Information Processing Society Conference*, (Syracuse NY), Sep. 1997.

[17] O. Nasraoui and R. Krishnapuram, "A novel approach to unsupervised robust clustering using genetic niching," in *In Proceedings of the Ninth IEEE International Conference on Fuzzy Systems*, pp. 170–175, 2000.

[18] O. Nasraoui, E. Leon, and R. Krishnapuram, "Unsupervised niche clustering: Discovering an unknown number of clusters in noisy data sets," in *Evolutionary Computing in Data Mining*, Invited chapter, A. Ghosh and L. C. Jain, Eds, Springer Verlag, 2004.

[19] O. Nasraoui and E. Leon, "Improved niching and encoding strategies for clustering noisy data sets," in *Proceedings of Genetic and Evolutionary Computation Conference, Seattle, WA*, 2004.

[20] D. Goldberg and J. J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Proceedings Second International Conference on Genetic Algorithm*, pp. 41–49, 1987.

[21] S. W. Mahfoud, "Crowding and preselection revisited," in *Proceedings Second Conference Parallel Problem Solving from Nature*, 1992.

[22] S. W. Mahfoud, "A comparison of parallel and sequential niching methods," in *Proceedings of the Sixth International Conference on Genetic Algorithms*, 1995.

[23] O. Nasraoui and R. Krishnapuram, "A robust estimator based on density and scale optimization, and its application to clustering," in *IEEE International Conference on Fuzzy Systems*, (New Orleans), pp. 1031–1035, Sep. 1996.

[24] F. Provost, T. Fawcett, and R. Kohavi, "The case against accuracy estimation for comparing induction algorithms," in *Proceedings of 15th intenational conference on machine learning*, p. 445;453, 1998.

[25] F. Provost and T. Fawcett, "Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions," in *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, 1997.

[26] D. Wlodzislaw, "Data sets used for classification: Comparison of results," in *http://www.phys.uni.torun.pl/kmk/projects/datasets.html*.

[27] F. Hoffmann, "Boosting a genetic fuzzy classifier," in *BICS Seminar, Royal Institue of Technology*, 2001.

[28] K. Yamanishi, J. Takeuchi, and G. Williams, "On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms," in *In Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000.