Non-stationary data mining: the Network Security issue

Sergio Decherchi, Paolo Gastaldo, Judith Redi, Rodolfo Zunino

Dept. of Biophysical and Electronic Engineering (DIBE), Genoa University
Via Opera Pia 11a, 16145 Genoa, Italy
{sergio.decherchi, paolo.gastaldo, judith.redi, rodolfo.zunino}@unige.it

Abstract. Data mining applications explore large amounts of heterogeneous data in search of consistent information. In such a challenging context, empirical learning methods aim to optimize prediction on unseen data, and an accurate estimate of the generalization error is of paramount importance. The paper shows that the theoretical formulation based on the Vapnik-Chervonenkis dimension (d_{vc}) can be of practical interest when applied to clustering methods for data-mining applications. The presented research adopts the K-Winner Machine (KWM) as a clustering-based, semi-supervised classifier; in addition to fruitful theoretical properties, the model provides a general criterion for evaluating the applicability of Vapnik's generalization predictions in data mining. The general approach is verified experimentally in the practical problem of detecting intrusions in computer networks. Empirical results prove that the KWM model can effectively support such a difficult classification task and combine unsupervised and supervised.

Keywords: Clustering, Data Mining, K-Winner Machine, Intrusion detection systems, Network security

1 Introduction

Data mining exploits clustering methods to arrange huge amounts of data into a structured representation, which eventually support the search for relevant information. The vast datasets and the heterogeneous descriptions of patterns set stringent requirements on the algorithms adopted; in such data-intensive applications, empirical learning methods aim to optimise prediction on unseen data [1]. In this regard, the formulation based on the Vapnik-Chervonenkis dimension (d_{vc}) [2] exhibits a general, theoretical foundation endowed with the widest validity for the accurate estimate of the run-time generalization error.
This paper shows that, in the specific case of clustering methods for data-mining applications, those theoretical results can yet have practical significance. Several aspects seem to favor clustering-based approaches [1]: first, data mining applications are typically rich with patterns and can offer the sample size required to tighten theoretical bounds. Secondly, the associate classifiers prove much simpler

than other approaches [1]; finally, a clustering-based classifier, the K-Winner Machine (KWM) model [3], has been fully characterized in compliance with Vapnik's theory. Theoretical results [3] proved that the KWM model can set very tight bounds to generalization performance. Moreover, the model is independent of the data dimensionality and inherently supports multi-class classification tasks [3]; these features boost the model's application to large masses of high-dimensional data.

The method presented in this paper applies the hybrid paradigm of the KWM model to the complex Anomaly Intrusion Detection (AID) problem, in which 'abnormal' events in computer networks traffic are identified by dynamically modelling 'normal' traffic. Clustering-based implementations of AID [4–9] typically map the network activity into a feature space, and the cluster-based model identifies those space portions that support the distribution of normal traffic, whereas outliers will mark abnormal traffic activities. This research shows that applying the hybrid paradigm of the KWM model to AID can lead to some intriguing results from both a theoretical and an applicative viewpoint.

From a theoretical perspective, the research addresses some issues related to the possible non-stationary distribution of observed data [10], which ultimately gives rise to a discrepancy between the pattern distribution in the training and the test phases. The paper formulates a general criterion to evaluate the consistency and consequent applicability of Vapnik's approach, by measuring the discrepancy between the empirical training set and the test distribution used at run time to control generalization performance.

From an applicative viewpoint, the major result consisted in showing that a KWM could effectively support such a complex classification task, mainly thanks to the model's ability to handle multi-class data distributions. The "KDD Cup 1999" dataset [10] provided the experimental domain for testing the proposed framework. This reliable benchmark is a common testbed for comparing the performances of anomaly-detection algorithms; indeed, experimental results proved that the KWM-based classifier outperformed other clustering-based AID's. The following conventions will be adopted throughout the paper:

1. $C = \{c^{(h)}, h = 1, ..., N_c\}$ is the set of N_c possible pattern classes;
2. $W' = \{(\mathbf{w}_n, c_n), \mathbf{w}_n \in R^D, c_N \in C, n = 1, ..., N_h\}$ is a set of N_h labeled prototypes;
3. $\mathbf{w}^*(\mathbf{x}) = \arg\min_{w \in W'}\{||\mathbf{x} - \mathbf{w}||^2\}$ is the prototype that represents a pattern, $\mathbf{x}$;

2 The K-Winner Machine Model

The training strategy of the KWM model develops a representation of the data distribution by means of an unsupervised process, then builds a classifier on top of that via some calibration process. The basic design criterion is to model the data distribution by Vector Quantization. In the research presented here, the Plastic Neural Gas (PGAS) [11] algorithm for Vector Quantization models the data distribution. The PGAS approach extends the 'Neural Gas' model [12] by

some crucial advantages: first, both the number and the positions of prototypes are adjusted simultaneously [11]; secondly, the training process prevents the occurrence of 'dead vectors' (void prototypes covering empty partitions). After VQ training positions the codebook prototypes a calibration process [3] labels the resulting Voronoi tessellation of the data space induced by the positions of the prototypes. Such a process categorizes each partition/prototype according to the predominant class; tie cases can be solved by choosing any class from among the best candidates. A detailed outline of the KWM training algorithm is given in [3]. Here we present in algorithm 1 the runtime operation of KWM.

Algorithm 1 The K-Winner Machine run-time operation

1: **procedure** KWM-FORWARD(test pattern $\mathbf{x}$; a calibrated set of N_h prototypes W'; error bounds, $\pi(k)$, for agreement levels, $k = 1, ..., N_h$)
2: `(Pattern Vector Quantization)` Build a sorted set of prototypes, $W''(\mathbf{x})$, arranged in increasing order of distance from $\mathbf{x}$
3: `(Count concurrences)` Determine the largest value of $K, 1 \leq K \leq N_h$, such that all elements in the sequence $\{(\mathbf{w}_k, c_k) \in W^*, k = 1, 2, ..., K\}$ share the same calibration, c^*
4: `(Determine generalization error bound)` Assigne risk bound to the classification outcome of pattern $\mathbf{x}$ $\pi^* = \pi(K)$
5: `(Output)`
- Categorize $\mathbf{x}$ with class c^*
- Associte the prompted output with an error bound, π^*

Statistical Learning Theory can apply to this framework because the KWM model allows one to compute Vapnik's bound to the predicted generalization performance at the local level. In this regard, it has been proved [3] that:

1. The VC-dimension of a Nearest-Prototype Classifier that includes a codebook of N_h prototypes is $d_{vc}^{(1)} = N_h$.
2. The Growth function of a KWM using a codebook of N_h prototypes, N_c classes and a sequence of K concurring elements is: $GF^{(K)}(N_p) = N_c^{\lfloor N_h/K \rfloor}$.

Generalization theory proves that a classifier's performance is upper-bounded by the empirical training error, ν, increased by a penalty. In the latter term, the Growth Function [2], $GF^{(K)}(N_p)$, measures the complexity of the fact that the classifier has been trained with a set of N_p patterns. This theory derives a worst-case bound, π, to the generalization error of the considered classifier:

$$\pi \leq \nu + \frac{\epsilon}{2}\left(1 + \sqrt{1 + \frac{4\nu}{\epsilon}}\right) \tag{1}$$

where $\epsilon = \frac{4}{N_p}[\ln GF(N_p) - ln\frac{\eta}{4}]$, and η is a confidence level. KWM theory [3] proves that one can compute a error bound, $\pi(k)$, for each agreement level, k,

and more importantly, that such a bound is a non-increasing function when k increases. This confirms the intuitive notion that the risk in a classification decision about a given point should be reduced by the concurrence of several neighboring prototypes. As a consequence of 1) and 2), unsupervised prototype positioning sharply reduces the bounding term in (1). By contrast, the KWM training algorithm does not provide any a-priori control over the first term (the empirical training error). This brings about the problem of model selection, which is usually tackled by a tradeoff between accuracy (classification error in training) and complexity (number of prototypes).

3 Statistical Learning Theory in Data-Mining applications

Vapnik's theory adopts a worst-case analysis, hence the predicted bound (1) often falls in a very wide range that eventually lessens the practical impact of the overall approach. However, data-mining environments typically involve very large datasets, whose cardinality ($N_p >> 10^5$) can actually shrink the complexity penalty term down to reasonable values. Moreover the KWM model intrinsically prevents an uncontrolled increase in the classifier's d_{vc}. Thus data-mining domains seem to comply with basic Statistical Learning Theory quite well. On the other hand, data-intensive applications raise the crucial issue of the stationary nature of the pattern distribution, which is basic assumption for the applicability itself of Statistical Learning Theory. In fact, non-stationary environments are quite frequent in data mining, as is the case for the present research dealing with network traffic monitoring: new attack patterns are continuously generated and, as a result, it is impossible to maintain a knowledge base of empirical samples up to date. Theoretical predictions, indeed, are often verified on a test set that should not enter the training process. This is done for a variety of reasons: either because one uses cross-validation for model selection, or because the test set is partially labeled, or because the test set was not available at the time of training. The stationary-distribution assumption may be rephrased by asserting that the training set instance, $T = \{(\mathbf{x}_l^T, c_l), \mathbf{x}_l^T \in R^D, c_l \in C, l = 1, ..., N_p\}$, and the test set instance, $S = \{(\mathbf{x}_j^S, c_j), \mathbf{x}_j^S \in R^D, c_j \in C, j = 1, ..., N_u\}$, are identically and independently drawn from a common probability distribution, $P(\mathbf{x}, c)$; otherwise, the training set is not representative of the entire population, hence expression (1) may not provide the correct estimate of classification accuracy.
The present work proposes a general yet practical criterion to verify the consistency of the generalization bounds; the method uses the VQ-based paradigm of the KWM model to check on the stationary-distribution assumption, and completes in three steps. First, one uses the prototypes in the trained codebook, W, to classify training and test data. Secondly, one estimates the discrete probability distributions, $\hat{T}$ and $\hat{S}$, of the training set and of the test set, respectively; this is easily attained by counting the number of training/test patterns that lie within the data-space partition spanned by each prototype. Finally, one computes the Kullback-Leibler (KL) divergence to measure the mutual information

between $\hat{T}$ and $\hat{S}$, and therefore decides whether the two associate samples have been drawn from the same distribution. In the discrete case, the KL divergence of probability distribution, $\hat{S}$, from the reference distribution, $\hat{T}$, is defined as:

$$D_{KL}(\hat{S}, \hat{T}) = \sum_{n=1}^{N_h} s_n \log \frac{s_n}{t_n} \tag{2}$$

where s_n and t_n denote the normalized frequencies associated with $\hat{S}$ and $\hat{T}$, respectively. Using the result, $\hat{T}$, of the training process as a reference distribution offers some advantages: it is consistent from a cognitive perspective, since it seems reasonable to adopt an empirical model of the data distribution; in addition, the PGAS algorithm prevents the occurrence of dead vectors during training, hence one has: $t_n > 0, \forall n$; finally, the partitioning schema sets a common ground for comparing the distributions of training and test patterns. The minimum (zero) value of $D_{KL}(\hat{S}, \hat{T})$ marks the ideal situation and indicates perfect coincidence between the training and test distributions. Non-null values, however, typically occur in common practice, and it may be difficult to interpret from such results the significance of the numerical discrepancies measured between the two distributions.

The present research adopts an empirical approach to overcome this issue by building up a 'reference' experiment setting. First, one creates an artificial, stationary distribution, J, that joins training and test data: $J := T \cup S$. Secondly, one uses the discrete distribution J to draw at random a new training set, T_J, and a new test set, S_J, such that $T_J \cap S_J = \emptyset$. Both these sets have the same relative proportions of the original samples. Third, using these sets for a session of training and test yields a pair of discrete distributions, $\hat{S_J}, \hat{T_J}$; finally, one measures the divergence (2) between the new pair of data sets, by computing $D_{KL}(\hat{S_J}, \hat{T_J})$. The latter value provides the numerical reference for assessing the significance of the actual discrepancy value $D_{KL}(\hat{S}, \hat{T})$ by comparison. If the original sample had been drawn from a stationary distribution, then the associate discrepancy value, $D_{KL}(\hat{S}, \hat{T})$, should roughly coincide with the value, $D_{KL}(\hat{S_J}, \hat{T_J})$, computed on the artificial distribution J. In this case, the theoretical assumptions underlying Statistical Learning Theory hold, and the bound formulation (1) can apply. Otherwise, if one verifies that: $D_{KL}(\hat{S_J}, \hat{T_J}) << D_{KL}(\hat{S}, \hat{T})$, then one might infer that the original sampling process was not stationary, hence a direct application of theoretical results (1) is questionable. The overall algorithm for the validation criterion is outlined in Algorithm 2.

4 Semi-supervised Anomaly Detection in Network Security

Commercial implementations of Intrusion Detection Systems (IDS's) typically rely on a knowledge base of rules to identify malicious traffic. The set of rules, however, is susceptible to inconsistencies, and continuous updating is required to cover previously unseen attack patterns. An alternative approach envisions

Algorithm 2 Criterion for validating the applicability of theoretical bounds

1: **procedure** VALIDATE(a training set including N_T labeled data, $\mathbf{x}_i, c(\mathbf{x}_i)$; a test set including N_S labeled data, $\mathbf{x}_j, c(\mathbf{x}_j)$)

2: `(Training)` Apply a VQ algorithm on the training set and position the set of prototypes: $W' = \{(\mathbf{w}_n, c_n), \mathbf{w}_n \in R^D, c_N \in C, n = 1, ..., N_h\}$

3: `(Probability distribution)`

- Estimate the training discrete probability distribution, $\hat{T}$ as follows: $\hat{T} := \{P_n^{(T)}; n = 1, ..., N_h\}$; where: $P_n^{(T)} = \{\mathbf{x}_i^{(T)} \in R^D : \mathbf{w}^*(\mathbf{x}_i^{(T)}) = \mathbf{w}_n\}$;
- Estimate the test discrete probability distribution, $\hat{S}$ as follows: $\hat{S} := \{P_n^{(S)}; n = 1, ..., N_h\}$; where: $P_n^{(S)} = \{\mathbf{x}_i^{(S)} \in R^D : \mathbf{w}^*(\mathbf{x}_i^{(S)}) = \mathbf{w}_n\}$;

4: `(Measuring mutual information)`

- Compute normalized frequencies: $t_n = \frac{|P_n^{(T)}|}{N_T}$; $s_n = \frac{|P_n^{(S)}|}{N_S}$; $n = 1, ..., N_h$
- Compute the KL divergence between $\hat{T}$ and $\hat{S}$: $D_{KL}(\hat{S}, \hat{T}) = \sum_{n=1}^{N_h} s_n \log \frac{s_n}{t_n}$

5: `(Applicability of generalization theory)`

- Form an artificial discrete distribution by joining training and test data $J := T \cup S$;
- Draw from J at random a training set, T_J, and a test set, S_J, having the same relative proportions of the original data sets;
- Repeat steps (2,3,4) by using the new pair of sets;
- If $D_{KL}(\hat{S}_J, \hat{T}_J) \approx D_{KL}(\hat{S}, \hat{T})$ (ideally≈ 0): than Stationary nature is verified and generalization bounds are validated; else Stationary nature is not verified and generalization bounds are not supported empirically

adaptive systems that maintain a model of 'normal' traffic and generate alerts in the occurrence of 'abnormal' events. Thus, Anomaly Intrusion Detection (AID) systems do not use sets of rules and are capable of time-zero detection of novel attack strategies; to do that, they require a continuous modeling of normal traffic in a dynamic, data-intensive environment. Several approaches based on data mining techniques have been adopted for that purpose [4–9], which typically map network traffic into vector patterns spanning a D-dimensional 'feature' space. The research presented in this paper tackles the anomaly-detection problem by means of the KWM paradigm, mainly because the hybrid KWM model, combining unsupervised clustering with supervised calibration, seems to fit the problem representation that characterizes the anomaly-detection task. Crucial properties contribute to these benefits in the data-mining scenario: 1) the Growth Function of a KWM does not depend on the number of patterns (Theorem 2); 2) the performance properties of the classifiers do not depend on the dimension of the data space: this mitigates the 'curse of dimensionality' and boosts applications in high-dimensional spaces; 3) the KWM paradigm is inherently a multi-class model. In practice, the KWM model supports the anomaly-detection framework as follows. The off-line KWM training algorithm processes an empirical set, P, of N_p traffic data; each datum includes a D-dimensional feature vector and a multiclass indicator: $P = \{(\mathbf{x}_l, c_l), \mathbf{x}_l \in R^D, l = 1, ..., N_p\}$. Class labels, c_l, indicate whether pattern $\mathbf{x}_l$ derives from normal or abnormal traffic; suspect patterns may be further sub-classified according to the various typologies of attacks. This

results in both a codebook W' of N_h labeled prototypes and a set of error bounds, $\pi(k)$, associated with increasing values of the prototype-agreement parameter, k.

5 Performance Measurements in Network Security

The well-known "KDD Cup 1999" dataset [10, KDD] provided the experimental domain for the proposed framework. The data spanned a 41-dimensional feature space; each pattern encompassed cumulative information about a connection session. In addition to "normal" traffic, attacks belonged to four principle macro-classes, namely, "DoS" (denial-of-service), "R2L" (unauthorized access from a remote machine), "U2R" (unauthorized access to local "super user" privileges), "probing" (surveillance and other probing such as port scanning). For simplicity, the experimental sessions in this research involved the "10% training set," provided by the KDDCup'99 benchmark, which had been obtained by subsampling original training data at a 10% rate. The resulting training set included 494,021 patterns and preserved the original proportions among the five principal macro-categories cited above. The test set provided by the KDD challenge contained 311,029 patterns, and featured 17 'novel' attack schemes that were not covered by the training set. The pattern descriptors that took on categorical values, most notably "Protocol" and "Service", were remapped into a numerical representation. "Protocol" could assume three different values (TCP, UDP, ICMP) and was therefore encoded by a triplet of bits; each element of the triplet was associated to a protocol, and only one of those could be non-null. The "Service" descriptor took on eleven possible values, and was remapped accordingly into eleven mutually exclusive coordinates. In summary, the patterns forming the eventual dataset used in the experiments included 53-dimensional feature vectors.

5.1 Experimental validation of generalization bounds

The procedure described in Section 3 allows one to verify the stationary nature of the observed data distribution. Thus, the coverage spanned by the original distribution (T, S) was compared with the representation supported by the exhaustive distribution, $J = T \cup S$, that approximated a stationary situation. The artificial, reference training and test sets, T_J and S_J, were obtained by random resampling J. The KL divergence between the training and test coverages for both distributions (T, S) and (T_J, S_J) completed the validation process. Figure 1 reports on the empirical results obtained for increasing codebook sizes.

The results highlights two main aspects: first, the KL divergence for the original distribution (T, S) always resulted to be much larger than the divergence measured when training and test data were drawn from the stationary distribution (T_J, S_J).
Secondly, the sizes of the codebooks differed significantly in the two situations: when training and test data were drawn from a common distribution, J, the

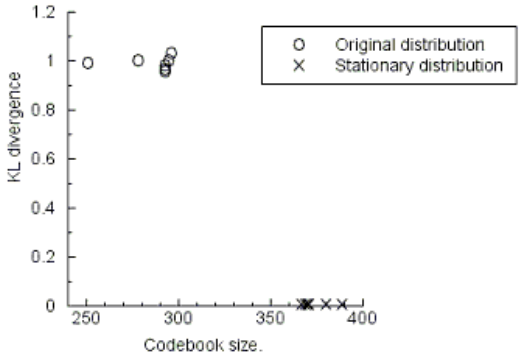

Fig. 1. Stationary Vs Original dataset

probability support was wider, hence the VQ algorithm required a larger number of prototypes to cover the data space. Conversely, original training data, T, were drawn from a limited sector of the actual support region, thus a smaller codebook was sufficient to represent the sample distribution.

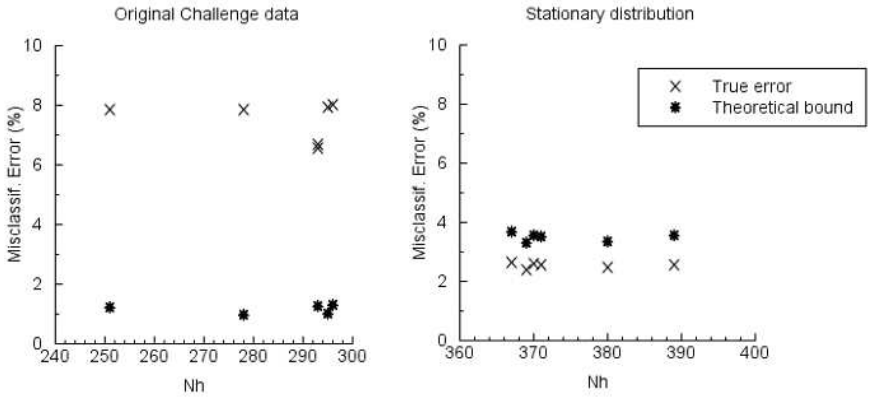

Fig. 2. Error bounds assessment: original data, stationary distribution

5.2 Prediction accuracy in intrusion detection

A complementary experimental perspective aimed to assess the effectiveness of the KWM method in terms of the classification accuracy on the actual dataset from the KDDCup'99 competition. The Vector Quantization set-up phase by using the PGAS clustering algorithm on training data indicated best performance with a codebook of 293 prototypes and an associated digital cost of 0.54%.

Table 1. KWM results

ACTUAL	PREDICTED					
	Normal	Probing	DoS	U2R	R2L	% correct
Normal	59118	152	1261	1	61	97.57%
Probing	720	3179	215	0	52	76.31%
DoS	1274	154	228425	0	0	99.38%
U2R	85	136	0	4	3	1.75%
R2L	14838	25	1317	0	9	0.06%
%correct	77.75%	87.19%	98.79%	80%	7.20%	

Table 2. KDD99 winner results

ACTUAL	PREDICTED					
	Normal	Probing	DoS	U2R	R2L	% correct
Normal	60262	243	78	4	6	99.5%
Probing	511	3471	184	0	0	83.3%
DoS	5299	1328	223226	0	0	97.1%
U2R	168	20	0	30	10	13.2%
R2L	14527	294	0	8	1360	8.4%
%correct	74.6%	64.8%	99.9%	71.4%	98.8%	

Such empirical evidence, mainly due to the marked discrepancies between training and test data sets, clearly seemed to invalidate the applicability of the theoretical bounds from Statistical Learning Theory for the KDD'99 dataset. As a result, the outcome of the validation criterion was that Vapnik's bound would not hold for the original challenge data. For the sake of completeness, Figure 2 compares the actual classification error with the theoretical bound for the original and the stationary distribution. The obtained results show that theoretical predictions fail in bounding the generalization performance for the original data sets, whereas provide good approximations when the data distribution is artificially reduced to the stationary case.

Such a conclusion gave both an empirical support and a numerical justification to a fact that has often been reported in the literature, namely, the considerable discrepancy between training and test patterns in the KDD dataset. Such a critical issue had been hinted at by the proponents themselves of the competition dataset [13, 14], and possibly explains the intrinsic difficulty of the challenge classification problem. In the subsequent validation phase, test data were classified, and the resulting performance was measured by using the scoring rules adopted for the KDD99 competition. In spite of the very low training error, the test error rate was 6.52%. In view of the discussion presented in the previous Section, such a phenomenon seems depend on the non-stationary nature of the data distribution underlying the original challenge datasets. Table 1 compares the confusion matrix for the obtained results with the corresponding matrix for the winning method [13]. When applying the error-weighting scheme of the KDD99 competition [14], the KWM-based approach achieved a score of 0.2229, which slightly improved on the result attained by the winner method, which scored 0.2331.

Such a result seems to be interesting especially in view of the semi-supervised nature of the proposed approach, as compared with the winning method that adopted a fully supervised strategy (decision trees) and therefore was subject to less stringent bounds to the predicted generalization performance.

References

1. Mirkin, B.: Clustering for Data Mining: a Data-recovery Approach. (2006)
2. Vapnik, V.: Estimation of Dependences Based on Empirical Data. Springer-Verlag (1982)
3. Ridella, S., Rovetta, S., Zunino, R.: K-winner machines for pattern classification. IEEE Trans. on Neural Networks **12** (2001) 371–385
4. Kemmerer, R., Vigna, G.: Intrusion detection: a brief history and overview. Computer **35** (2002) 27–30
5. Portnoy, L., Eskin, E., Stolfo, S.J.: Intrusion detection with unlabeled data using clustering. In: Proc. ACM CSS Workshop on Data Mining Applied to Security. (2001) 123–130
6. Eskin, E., Arnold, A., Prerau, M.: A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. Applications of Data Mining in Computer Security (2002)
7. Oh, S.H., Lee, W.S.: An anomaly intrusion detection method by clustering normal user behavior. Computers and Security **22** (2003) 596–612
8. Lee, W., Stolfo, S., Mok, K.: Adaptive intrusion detection: a data mining approach. Artificial Intelligence Review **14** (2000) 533–567
9. Zheng, J., Hu, M.: An anomaly intrusion detection system based on vector quantization. IEICE Trans. Inf. and Syst. **E89-D** (2006) 201–210
10. KDD Cup 1999 Intrusion detection dataset: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
11. Ridella, S., Rovetta, S., Zunino, R.: Plastic algorithm for adaptive vector quantization. Neural Computing and Applications **7** (1998) 37–51
12. TM, M., SG, B., KJ, S.: Neural gas network for vector quantization and its application to time-series prediction. IEEE Trans. Neural Networks **4** (1993) 558–569
13. Pfahringer, B.: Winning the kdd99 classification cup: bagged boosting. SIGKDD Explorations **1** (2000) 65–66
14. Results of the KDD'99 Classifier Learning Contest: http://www-cse.ucsd.edu/users/elkan/clresults.html.