Collateral Representative Subspace Projection Modeling for Supervised Classification

Thiago Quirino, Zongxing Xie, Mei-Ling Shyu
Department of Electrical and Computer Engineering
University of Miami, Coral Gables, FL 33124, USA
t.quirino@umiami.edu, z.xie1@umiami.edu, shyu@miami.edu

Shu-Ching Chen
Distributed Multimedia Information
System Laboratory
School of Computing and Information Sciences
Florida International University
Miami, FL 33199, USA
chens@cs.fiu.edu

LiWu Chang
Center for High Assurance Computer Systems
Naval Research Laboratory
Washington, DC 20375, USA
lchang@itd.nrl.navy.mil

Abstract

In this paper, a novel supervised classification approach called Collateral Representative Subspace Projection Modeling (C-RSPM) is presented. C-RSPM facilitates schemes for collateral class modeling, class-ambiguity solving, and classification, resulting a multi-class supervised classifier with high detection rate and various operational benefits including low training and classification times and low processing power and memory requirements. In addition, C-RSPM is capable of adaptively selecting nonconsecutive principal dimensions from the statistical information of the training data set to achieve an accurate modeling of a representative subspace. Experimental results have shown that the proposed C-RSPM approach outperforms other supervised classification methods such as SIMCA, C4.5 decision tree, Decision Table (DT), Nearest Neighbor (NN), KNN, Support Vector Machine (SVM), 1-NN Best Warping Window DTW, 1-NN DTW with no Warping Window, and the well-known classifier boosting method AdaBoost with SVM.

1 Introduction

During the past decade, classification techniques have been applied to innumerous research areas including market analysis, homeland defense, threat assessment systems, intrusion detection systems, credit card fraud detection, face recognition, etc. Generally speaking, there are two broad types of classification procedures: unsupervised and supervised classification [5]. Unsupervised classification is usually employed when relatively little information is known about the data before classification; while supervised classification requires a class quantity that is high enough to distinguish various class types. Detailed, supervised classification of large areas takes enormous input effort in terms of time, manpower, and money [1].

For the past decade, PCA-based [2] approaches to supervised classification have gained popularity in innumerous research domains, specially due to the dimensionality reduction capabilities of PCA (Principal Component Analysis). For instance, [7][8] have popularized the use of PCA in the domain of face recognition. Other PCA-based methods, such as PCC [9] and SIMCA [12], thrive on the generality of their methods, and attempt to develop a classifier that is as stable as possible to the different distribution of various existing data sets. Although having in common the use of the PCA approach for dimensionality reduction, all the classifiers mentioned above approach the classification task differently. In [7][8], the authors introduced a Bayesian classification approach that utilizes eigenspace decomposition to simplify the estimation of Gaussian probability densities for use as likelihood density estimates in their Bayesian approach. On the other hand, methods such as SIMCA [12] and PCC [9] generated PCA models for each class in a training data set independently and then employed a class-deviation measure that, unlike the Bayesian approach, may result in a testing instance being classified into more than one class. The main difference between SIMCA and PCC is that while SIMCA is concerned with the use of PCA only for dimensionality reduction, PCC combines the intrinsic information provided by both major and minor principal components as well as their respective eigenvalues into its class-deviation score function.

In this paper, a novel supervised classification approach, Collateral Representative Subspace Projection Modeling (C-RSPM), is proposed which reworks PCC [9][10][11][15] into a base class-deviation measure for supervised classification. C-RSPM introduces the Representative Subspace Projection Modeling (RSPM) technique to enhance the advantages of PCC. In contrast to the requirements of existing PCA based methods, in RSPM, a subset of components needs neither to be in consecutive order [7][8][12] nor to retain the entire principal component set [7][8], and are automatically and adaptively selected from the principal component set acquired from each class in a training data set rather than through human observation [7][8]. These nonconsecutive components are labeled representative components and form a representative subspace from which statistics suitable for a classification task are acquired. Unlike PCC [9][10][11][15] which requires the estimation of the distribution of two classification thresholds, namely $C1$ (derived from major principal components) and $C2$ (derived from minor principal components), the RSPM technique allows C-RSPM to employ a single threshold measure C in the multi-class classification task. C is a class deviation quantity whose distribution is acquired from a set of representative nonconsecutive components, rather than from both major and minor components as in PCC. RSPM simplifies the analysis of the false alarm rate of C-RSPM by providing a single threshold function, which in turn also contributes to the efficiency and speed of the statistical computations of the C-RSPM approach.

Our experimental results with various data sets from the UCI [13] and UCR archives [4] demonstrate that C-RSPM outperforms many methods such as SIMCA, C4.5 decision tree, Decision Table (DT), NN, KNN, SVM, and AdaBoost with SVM in the area of multi-class supervised classification. Furthermore, C-RSPM's applicability to the network intrusion detection domain, assessed through experiments with KDD Cup 99 [3] data sets, demonstrate that C-RSPM is an ideal solution to misuse detection applications where a high detection rate and lightweight classifier is required, as for instance, in the lightweight agent-based intrusion detection architecture proposed in our previous work [15].

The remainder of this paper is organized as follows. In Section 2, the C-RSPM architecture for supervised classification is presented, including an elaboration on the formulations of the RSPM data set adaptive technique. Experiments and comparison results are discussed in Section 3. Finally, conclusions are given in Section 4.

2 C-RSPM Supervised Classification

2.1 The C-RSPM Architecture

The architecture of the proposed C-RSPM approach is illustrated in Figure 1, which includes the $Classification$ module and $Ambiguity\ Solver$ module.

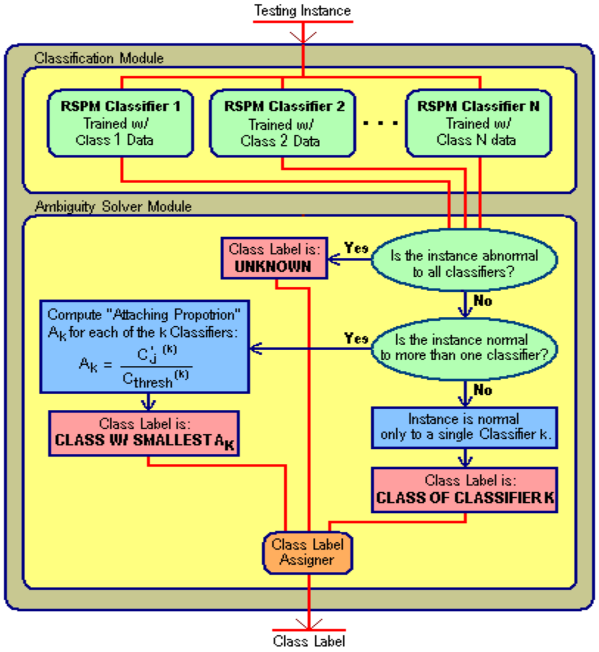

Figure 1. The C-RSPM Architecture

The $Classification$ module is composed of an array of deviation classifiers which are executed collaterally, that is, each of the classifiers receives and classifies the same testing instance simultaneously. The basic idea of the C-RSPM algorithm is that each classifier in the $Classification$ module, as shown in Figure 1, is trained with a different type of known-class data from a training data set, through the employment of the RSPM technique. Thus, training the C-RSPM classifier consists basically of training each individual classifier to recognize instances of a specific class type. The challenge of using the RSPM technique in multi-class supervised classification is in keeping a high true detection rate with an enough low false alarm rate. Theoretically, a testing instance normal to a certain classifier's training data should either be entirely rejected by all the other classifiers or be classified into no more than one class group. In this context, two common issues should be carefully considered and approached if the proposed collateral classification architecture is to function properly:

- Globally unrecognized instance issue: A testing instance is rejected by all classifiers and cannot be assigned a class label.

- Classification ambiguity issue: More than one classifier accepts a testing instance as statistically normal to their training data.

With a low programmed false alarm rate, the first issue usually translates into the testing instance belonging to an unknown class type. In the current C-RSPM approach, the rejected testing instances are simply assigned an "$Unknown$" class label, which can also be addressed via unsupervised classification. The second and most critical issue arises due to the fact that no classifier can ever ensure a 100% classification accuracy, and also due to the fact that some data sets are composed of classes with very similar correlative properties among them. To approach this issue, the $Ambiguity\ Solver$ module is proposed to coordinate and capture classification conflicts. This module defines a class attachment measure A_k called the $Attaching\ Proportion$ for each of the k ambiguous classes, i.e., for all classes that during the classification phase have recognized the testing instance as statistically normal to their training data set. The meaning of the $Attaching\ Proportion$ quantity will be explored later after the parameters necessary for its proper elaboration are presented.

The employments of both PCA [2] in feature vector dimensionality reduction and some form of class-deviation measure (also known as anomaly detection) is common to most supervised classifiers whose models are based on statistical inferences. For instance, in the SIMCA [12] method, similarly to the methods proposed in this paper, a PCA model is generated for each of the classes in the training data set. Nevertheless, unlike in the C-RSPM approach, in SIMCA, the principal components retained to model each class are required to be consecutive in order and the number of components retained may be decided manually (as in [7]) or through a cross-validation technique. Furthermore, SIMCA uses a measure of class-deviation which, unlike C-RSPM, does not take into account the information contained in the eigenvectors and eigenvalues. Instead, its class-deviation measure is given by an F-test on the linear combination of the Euclidean distances between a testing instance and its projection in eigenspace and to the rectangular-like boundaries generated for each training data class. Although an instance can be rejected by the SIMCA classifier as statistically abnormal to all its trained PCA models, multiple class assignment is allowed. This is another prevailing difference between SIMCA and C-RSPM, where class-label ambiguity is not allowed in C-RSPM.

2.2 Classification Module - The RSPM Technique

The core of the classification module in the C-RSPM supervised classification approach is the Representative Subspace Projection Modeling (RSPM) technique, which is inspired from the observation and comparison of curves derived from the sorted rows of the PCA score matrices [9][10][11][15] of various training data sets. The term "training data PCA score matrix" is referred to the projection of a matrix, whose columns and rows correspond to the instances of a class of a training data set and their various attribute values respectively, onto the eigenspace composed of all principal components acquired from the training data. Each row of a score matrix holds the projections of the training data instances onto the eigenspace dimension specified by a single principal component. Some of the generated curves of the score rows are very smooth, for instance, possessing a visibly horizontal slope representing the similar characteristics of groups of instances in a training data set in an intuitive manner, while other curves rise in an obvious non-zero slope. Furthermore, it can be noticed that the curve derived from the score row vector corresponding to the largest eigenvalue [9][10][11][15] is always smooth, in accordance to the fact that the largest eigenvalue resulting from PCA corresponds to the highest degree of similarity among the most correlated dimensions of the training data set. In our previous studies [9][10][11][15], a novel anomaly detection scheme that uses the robust Principal Component Classifier (PCC) to handle computer network security problems was presented. The main idea of PCC is to detect attacking traffic when it has a large and significant deviation from the normal traffic. That is, in PCC, anomalies are treated as outliers, and an intrusion predictive model is constructed from both the major and minor principal components obtained from normal training data instances. A distance threshold measure from an anomaly to the cluster of normal instances is then calculated in the transformed principal component space. In contrast, some PCA-based methods make use of only major components in a dimensionality reduction task, and do not utilize the inherent information found in the selected principal components subset, or their respective eigenvalues, in the formulation of their predictive model [12].

With these observations in mind, a principal component selection function based on the standard deviation, which reflects the degree of smoothness of a distribution, of the score matrix row vectors is defined to select representative, possibly nonconsecutive, principal components which can sufficiently model the essential features of a training data set adaptively. These chosen representative principal components are then utilized to compute a class-deviation measure, without resorting to empirical formulations based on static and non-adaptive parameters that can affect a predictive model's ability to learn the structure of various kinds of data sets. The experimental results in [9][10][11][15] revealed that the empirical choice of major components that account for a cumulative 50% of the total variation and minor components whose eigenvalues are less than 0.20 can be used to generally and decently represent the distribution of various training data sets, indicating the existence of hidden clues and reasoning, from a statistical point of view, on why these parameters and the flexibility of their values should reflect information acquired statistically from the training data set itself. Furthermore, reducing the number of classification thresholds to a single measure facilitates the analysis of a classifier's false alarm rate by requiring only one model, intended for the distribution of the single threshold measure, to be generated so that false alarm rate statistics can be computed.

For the training data of a given class, let $i = 1, 2, \ldots, p$, $j = 1, 2, \ldots, N$, and $\mathbf{X} = \{\mathbf{x}_{ij}\}$ be a $p \times N$-dimensional matrix containing N p-dimensional column vectors $\boldsymbol{X}_j = (\mathbf{x}_{1j}, \mathbf{x}_{2j}, \ldots, \mathbf{x}_{pj})'$, representing the N training instances of the class. In order to identify the $100\gamma\%$ extreme observations that are to be trimmed, the Mahalanobis distance is adopted for outlier trimming:

$$\mathbf{d}_j{}^2 \ = \ (\boldsymbol{X}_j - \bar{\boldsymbol{X}})'\mathbf{S}^{-1}(\boldsymbol{X}_j - \bar{\boldsymbol{X}}), \text{ where}$$

$$\bar{\boldsymbol{X}} \ = \ \tfrac{1}{N}\sum_{j=1}^{N}\boldsymbol{X}_j \ \text{ and } \ \mathbf{S} \ = \ \tfrac{1}{N-1}\sum_{j=1}^{N}(\boldsymbol{X}_j - \bar{\boldsymbol{X}})(\boldsymbol{X}_j - \bar{\boldsymbol{X}})'$$

Accordingly, for a given γ value (e.g., γ= 0.005), those observations corresponding to the γ*N largest values in vector $\boldsymbol{D} = \{\mathbf{d}_j{}^2\}$ will be removed.

Assume that after trimming, the training data set has L instances ($L < N$) with elements $\mathbf{x}_{ij}$, where $i = 1, 2, \ldots, p$ and $j = 1, 2, \ldots, L$, and $\bar{\mu}_i$ and s_{ii} are the sample mean and variance of the i^{th} row (i.e., i^{th} feature) in this training data set. We can generate the normalized training data set $\mathbf{Z} = \{\mathbf{z}_{ij}\}$ using Equation (1), and $\boldsymbol{Z}_j = (\mathbf{z}_{1j}, \mathbf{z}_{2j}, \ldots, \mathbf{z}_{pj})'$ is the corresponding column vectors of $\mathbf{Z}$.

$$\mathbf{z}_{ij} \ = \ \frac{\mathbf{x}_{ij} - \bar{\mu}_i}{\sqrt{s_{ii}}}. \tag{1}$$

Next, compute a new robust estimate of the correlation matrix $\mathbf{S}$ using matrix $\mathbf{Z}$. Let (λ_1, $\boldsymbol{E}_1$), (λ_2, $\boldsymbol{E}_2$), …, (λ_p, $\boldsymbol{E}_p$) be the p eigenvalue-eigenvector pairs of the robust correlation matrix $\mathbf{S}$, where $\lambda_1 \ \geq \lambda_2 \ \geq \ldots \ \ \geq \lambda_p \ \geq \ 0$. Also, let matrix $\mathbf{Y}$= $\{\mathbf{y}_{ij}\}$ be the projection of matrix $\mathbf{Z}$ onto the p-dimensional eigenspace, where $\mathbf{Y}$ is defined as the training data score matrix with the score column vectors $\boldsymbol{Y}_j$=$(\mathbf{y}_{1j}, \mathbf{y}_{2j}, \ldots, \mathbf{y}_{pj})'$ which correspond to the projection of each of the L normalized training instances onto the eigenspace composed of all p principal components acquired from the robust correlation matrix $\mathbf{S}$.

After all these steps, a principal component selection function is defined as given in Equation (2), which is based on the distribution of the eigenspace features, or the score row vectors $\boldsymbol{R}_i$ = $(\mathbf{y}_{i1}, \mathbf{y}_{i2}, \ldots, \mathbf{y}_{iL})$ of matrix $\mathbf{Y}$. The principal components whose corresponding score row vectors $\boldsymbol{R}_i$ satisfy this selection function are selected as the representative components, while all other principal components are discarded.

$$\phi < STD(\boldsymbol{R}_m) < \mathrm{a} + \mathrm{b} \times (1 - \exp(-\alpha)), \tag{2}$$

where ϕ is an adjustable coefficient used to filter out the score row vectors in $\mathbf{Y}$ with very little variability, i.e., with extremely small standard deviation values. Based on our empirical studies, ϕ is set to 0.0001 throughout our experiments. The variable α is the pre-set false alarm rate which is also adjustable, $STD(\mathrm{R_m})$ is the standard deviation of the score row vectors satisfying the selection function and corresponding to the $(m)^{th}$ principal component ($m \in \mathbf{M}$), $\mathbf{M}$ is defined as the selected principal component space. Finally, a and b are adjustable coefficients. From our empirical studies, a and b are both set to the arithmetic mean of the standard deviation values from those score row vectors $\boldsymbol{R}_i \in \mathbf{Y}$ whose standard deviation values are greater than the refinement threshold ϕ. In Equation (2), the value of the right-hand side inequality should be in $[a, a+b)$. To avoid the inequality goes to infinite, a and b are restricted to finite values. Please also note that the right-hand side value of Equation (2) increases as α increases, resulting in a decrease in the restrictions to outlier detection with an increase of the false alarm rate.

In this manner, those principal components whose corresponding score row vectors in matrix $\mathbf{Y}$ satisfy the threshold function given in Equation (2) are selected automatically as representative components, and are used to calculate the distance threshold measure $\mathrm{C_{thresh}}$, replacing the $C1$ and $C2$ thresholds proposed in [9][10][11]. Therefore, $\mathrm{C_{thresh}}$ is used as the discriminator in the multi-class classification process. To determine this threshold from the L projected training instances in matrix $\mathbf{Y}$, compute the distance vector $\boldsymbol{C} = \{\mathbf{c}_j\}$, $j = 1, 2, \ldots, L$ using the class-deviation function:

$$\mathbf{c}_j = \sum_{m \in M} \tfrac{(\mathbf{y}_{mj})^2}{\lambda_m},$$

where $m \in \mathbf{M}$ is the index of the features (or rows) corresponding to the representative principal components, λ_m is the eigenvalue of the m^{th} principal component, and $\mathbf{y}_{mj}$ is the score value of the m^{th} eigenspace feature for the j^{th} projected training data instance. The $\boldsymbol{C}$ vector holds the distribution of the class-deviation scores of the L projected training data set observations.

Then, the elements in vector $\boldsymbol{C}$ are sorted in the ascending order of values, where the sorted score vector

SC is now called an empirical distribution. With the desired programmed false alarm rate α of the classifier, find the element in ***C*** that corresponds to the approximately $((1-\alpha)*100\%)^{th}$ percentile of its empirical distribution, that is, the element in ***SC*** corresponding to the nearest integer value to $((1-\alpha)*L)$. This value corresponds to the threshold value $\mathrm{C}_{\mathrm{thresh}}$ of that class with the programmed false alarm rate α, and will be used in the classification phase.

Now, let **X'**= $\left\{\mathbf{x}'_{ij}\right\}$, $i = 1,2,\ldots,p$ and $j = 1,2,\ldots,N'$, be a $p \times N'$-dimensional matrix containing N' p-dimensional column vectors $\boldsymbol{X}'_j$=$(\mathbf{x}'_{1j},\mathbf{x}'_{2j},\ldots,\mathbf{x}'_{pj})'$, representing the N' testing instances of a class. Next, let $\boldsymbol{Y}'_j$=$(\mathbf{y}'_{1j},\mathbf{y}'_{2j},\ldots,\mathbf{y}'_{pj})'$ correspond to the projection of each of the N' testing instances, normalized through Equation (1) with the same $\bar{\mu}_i$ and s_{ii} values acquired from the training data set after trimming, onto the eigenspace composed of all p principal components from the robust correlation matrix **S**. Next, compute the distance vector ***C'*** = $\left\{\mathbf{c}'_j\right\}$ ($j = 1,2,\ldots,N'$) using the class-deviation function:

$$\mathbf{c}'_j = \sum_{m\in M} \frac{(\mathbf{y}'_{mj})^2}{\lambda_m}$$

for all N' projected testing instances, where λ_m is the eigenvalue of the m^{th} representative component, $m \in \mathbf{M}$ is the index of those representative components, and $\mathbf{y}'_{mj}$ is the the score value of the m^{th} eigenspace feature of the normalized and projected original instance $\boldsymbol{X}'_j$. Finally, the classification rules classify each of the observations $\boldsymbol{X}'_j$ ($j = 1,2,\ldots,N'$) using the following decision rules.

The j^{th} instance is classified as statistically abnormal to the class training data if

$$\mathbf{c}'_j > \mathrm{C}_{\mathrm{thresh}},$$

and classified as otherwise normal to the class if

$$\mathbf{c}'_j \leqslant \mathrm{C}_{\mathrm{thresh}}.$$

2.3 Ambiguity Solver Module

Assuming there are more than one classifier trained by the RSPM technique, as shown in the *Classification Module* of Figure 1, which classify an unknown testing instance as statistically normal to the training data of their respective classes. To solve this consequent classification ambiguity issue, the *Attaching Proportion* measure A_k is developed for each of the k ambiguous classes in Equation (3).

$$A_k = \frac{{\mathbf{c}'_j}^{(k)}}{{\mathrm{C}_{\mathrm{thresh}}}^{(k)}}, \tag{3}$$

where A_k is calculated for each of the k ambiguous classes, ${\mathrm{C}_{\mathrm{thresh}}}^{(k)}$ is the threshold value of the classifier trained with the data of the class type k, ${\mathbf{c}'_j}^{(k)}$ is the class-deviation score of the j^{th} testing instance calculated under the classifier trained with the data of the class type k. The class label of the classifier with the lowest A_k value is assigned to the ambiguous testing instance since the lowest attaching proportion measure reflects a closer resemblance of the testing instance to the specific class type. A_k can be viewed as a measure of the degree of normality of an instance with respect to a class type, or in other words, the lower the A_k value is, the lower the distance of the testing instance in question to those instances lying close to the center of the spatial distribution of class k is. The ratio A_k ranges from [0, 1], since the maximum deviation score of the j^{th} normal instance under the class type k is ${\mathbf{c}'_j}^{(k)}$ = ${\mathrm{C}_{\mathrm{thresh}}}^{(k)}$, and it indicates in which percentile of normality, that is, within which percentile of the empirical distribution of the sorted class-deviation score vector of the classifier of class type k, the testing instance lies in. Hence, a lower A_k value indicates lower percentiles and greater similarity to the data of class type k.

2.4 Advantages of C-RSPM

The advantages of our proposed C-RSPM supervised classification approach are summarized as follows.

- $\mathrm{C}_{\mathrm{thresh}}$ is based on the automatically selected representative components from the principal component set derived from the training data set, rather than from the pre-defined choice of major and/or minor components originated from empirical formulations. Also, only a single threshold $\mathrm{C}_{\mathrm{thresh}}$ needs to be used for classification.

- Eliminates any dependency on statistic distributional assumptions about the training data sets. Unlike many statistical-based methods that need to assume a normal distribution of the parameters [7] or resort to the use of the central limit theorem by requiring the number of features to be greater than 30 [16][17], the RSPM technique does not have these constraints.

- It has significantly increased robustness to various kinds of training data sets possessing different distributions of principal components and eigenvalues, especially in domains where high dimensional data sets are used.

- It is lightweight, and has lower memory, storage, and processing power requirements than most of the other methods.

3 Performance Evaluation

To generalize performance evaluation, various data sets with different distributions are employed to assess the performance of C-RSPM in multiclass classification and misuse detection. The data sets for multiclass classification are from the UCI KDD Archive [13] and the UCR Time Series Classification/Clustering repository [4]; while the data set for misuse detection is the KDD Cup 1999 data [3]. C-RSPM is implemented in MatLab [6] as a supervised classification approach.

Please note that throughout the experiments, the array of classifier components of C-RSPM were programmed with a typical false alarm rate α value of 0.1%, that is, the values of the $99.9\%^{th}$ percentile of the empirical sorted scores distributions ***SC*** of the training data of each classifier component were chosen as the threshold values to be employed during classification. Also, for each classifier, 10-fold cross-validation experiments were executed for each data set. The C-RSPM is compared with SIMCA and methods implemented in the WEKA [14] software package, including C4.5 decision trees, Decision table (DT), NN, KNN (K=5), SVM, and AdaBoost with SVM.

3.1 Experimental results for Multiclass Classification

The data sets used for multiclass classification include:

- Wine data set: Acquired from the UCI KDD Archive [13] and composed of 3 classes. $\frac{2}{3}$ of each class data instances are randomly selected to train the classifier and the remaining $\frac{1}{3}$ of the instances are used to test the classifier.

- Pham (Synthetic Control) data set: Composed of 6 classes, and acquired from the UCR Time Series Classification/Clustering repository [4]. The data set is made available into already separate training and testing data sets. $\frac{2}{3}$ of training data are randomly selected to do the classification for cross-validation purpose.

- Xi (Face all) data set: Composed of 14 classes and acquired from the UCR Time Series Classification/Clustering repository [4]. The data set is made available into already separate training and testing data sets. $\frac{2}{3}$ of training data are randomly selected to do the classification for cross-validation purpose.

The classification accuracy and their corresponding standard deviations on the three selected data sets for C-RSPM and several other classification methods are presented in Table 1. As shown in this table, C-RSPM maintains a high classification accuracy ($> 96\%$) and outperforms the other selected supervised classification methods. This is indicative that the use of RSPM to generate predictive models for different class types allows the C-RSPM architecture to distinguish among distinct class types with high accuracy. Another observation that can be attained from these experimental results is the fact that the classification accuracy of some methods vary significantly among the training data sets used. For example, the accuracy of C4.5 ranges from 56.28% to 93.25%, and the accuracy of Decision Table (DT) ranges from 34.36% to 90.44%. These results indicate that these methods do not generate predictive models that are well fit for various types of training data sets. In contrast, C-RSPM maintains a high classification accuracy of over 96% for all the training data sets, generating predictive models well fit for all the data sets, as the results indicate. In addition, it needs to be pointed out that the classification results for the 1-NN Best Warping Window DTW (NNBWWDTW) and 1-NN DTW with no Warping Window (NNDTW) methods shown in Table 1 are acquired directly from [4] for the UCR data sets only, and no cross-validation results are available.

Table 1. Multiclass classification accuracy comparison among C-RSPM, SIMCA, NN, C4.5, DT, KNN (K=5), SVM, AdaBoost with SVM, NNBWWDTW, and NNDTW using the Wine, Pham, and Xi data sets. Standard deviations are shown in parentheses.

Accuracy ($\%$)	Wine	Pham	Xi
C-RSPM	96.12% (±0.10)	99.78% (±0.05)	99.05% (±0.08)
SIMCA	87.64% (±4.97)	99.60% (±0.15)	88.68% (±3.24)
NN	84.26% (±2.20)	88.00% (±3.72)	71.40% (±5.35)
C4.5	93.25% (±0.44)	91.57% (±0.87)	56.28% (±8.99)
DT	90.44% (±1.97)	68.33% (±6.86)	34.36% (±10.93)
KNN	84.26% (±5.32)	87.66% (±4.07)	64.38% (±5.22)
SVM	95.74% (±0.20)	96.33% (±1.03)	92.28% (±2.16)
AdaBoost with SVM	94.74% (±0.18)	96.33% (±0.56)	93.09% (±0.91)
NNBWWDTW	N/A	98.30%	80.80%
NNDTW	N/A	99.30%	80.80%

Another important observation is that C-RSPM has significantly lower training and classification times than all the other methods, specially when compared to those of SVM and SIMCA, the latter whose training time requirement is exacerbated by the use of cross-validation in the principal component selection process. In addition, C-RSPM requires less memory and storage for the components required for the classification phase, in contrast to the instance-based methods such as NN and KNN, which require storage for hundreds of training data instances.

3.2 Experimental results for Misuse Detection

Misuse detection, a specific application of multiclass supervised classification, is conducted to further validate the feasibility and effectiveness of C-RSPM in the network intrusion detection application domain which has gained increasingly momentum and demand in recent years. The network intrusion data set from the KDD Cup 1999 data [3] is used in the experiments and is composed of four classes of network attacks:

- Back data set: Composed of 441 training and 111 testing instances.
- Teardrop data set: Composed of 194 training and 77 testing instances.
- Smurf data set: Composed of 5000 training and 23000 testing instances.
- Neptune data set: Composed of 5000 training and 5719 testing instances.

Table 2. Misuse detection accuracy comparison among C-RSPM, SIMCA, C4.5, DT, NN, KNN, SVM, and AdaBoost with SVM for the KDD data. Standard deviations are shown in parentheses.

Accuracy ($\%$)	**KDD**
C-RSPM	$99.91(\pm0.05)\%$
SIMCA	$95.41(\pm1.10)\%$
C4.5	$92.94(\pm2.10)\%$
DT	$81.79(\pm3.60)\%$
NN	$99.11(\pm0.60)\%$
KNN	$99.30(\pm0.52)\%$
SVM	$99.70(\pm0.23)\%$
AdaBoost with SVM	$99.78(\pm0.12)\%$

Table 2 shows the classification accuracy and their corresponding standard deviations for the 10-fold cross-validation experiments for C-RSPM, SIMCA, C4.5 decision tree, DT, NN, KNN, SVM, and AdaBoost with SVM [14]. As shown in Table 2, C-RSPM maintains a high classification accuracy ($> 99\%$) and outperforms all the other selected algorithms. Furthermore, as previously mentioned, C-RSPM was observed to require lower training and classification times than the methods used in the evaluation. These results clearly depict the excellent performance of the C-RSPM approach. Moreover, the promising experimental results suggest that C-RSPM seems to offer a very suitable lightweight and highly accurate solution to the pending high detection rate and lightweight misuse detection method required by many intrusion detection applications [15].

4 Conclusion

A novel supervised classification approach called C-RSPM is proposed in this paper. C-RSPM utilizes the RSPM technique, which is based on the powerful PCA tool and adaptive representative principal component selection technique. Furthermore, the RSPM technique is integrated with collateral classification and ambiguity solving modules into the C-RSPM architecture, capable of performing high accuracy supervised classification. The performance of the C-RSPM supervised classification approach was evaluated against the performance of many well known supervised classification algorithms such as SIMCA, NN, KNN, C4.5 decision tree, Decision Table, 1-NN Best Warping Window DTW, 1-NN DTW with no Warping Window, SVM and AdaBoost with SVM using various data sets from the KDD, UCI, and UCR archives. Experimental results have demonstrated that the C-RSPM approach performs the best among all the selected methods, maintaining an accuracy of over 96% on all experiments. In addition, the results also indicate that the C-RSPM approach yields a predictive model with higher stability and lower bias than the other selected methods, capable of maintaining a high detection rate for all the employed data sets, as opposed to the other methods, which present large variances in their detection rates for the same group of data sets employed. Furthermore, the novel C-RSPM and its main component, the RSPM technique, present high accuracy, as the experimental results indicate, and various operational benefits such as lower memory and processing power requirements than the other supervised classification methods in the performance comparison, all due to the fact that only very little information about the principal components and computed thresholds have to be stored for the execution of the classification stage. These benefits provide C-RSPM with a lightweight characteristic that makes its utilization suitable in real-time demanding applications.

5 Acknowledgments

For Mei-Ling Shyu, this research was supported in part by NSF ITR (Medium) IIS-0325260. For Shu-Ching Chen, this research was supported in part by NSF EIA-0220562 and NSF HRD-0317692.

References

[1] R. Fuller, G. Groom, and A. Jones. Photogramm. engg. *Remote Sensing*, 60:553–562, 1994.

[2] I. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, New York, 2nd edition, 2002.

[3] KDD. KDD Cup 1999 Data. *http://kdd.ics.uci.edu/databases/kddcup99/*, 1999.

[4] E. Keogh, X. Xi, L. Wei, and C. Ratanamahatana. The UCR time series classification/clustering. *http://www.cs.ucr.edu/∼eamonn/time_series_data/*, 2006.

[5] P. Laskov, P. Dussel, C. Schafer, and K. Rieck. Learning intrusion detection: supervised or unsupervised? In *ICIAP*, Cagliari, ITALY, October 2005.

[6] Mathworks. Matlab. *http://www.mathworks.com/matlabcentral/*.

[7] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):696–710, 1997.

[8] B. Moghaddam, W. Wahid, and A. Pentland. Beyond eigenfaces: Probabilistic matching for face recognition. In *Proc. of Int'l Conf. on Automatic Face and Gesture Recognition (FG'98)*, pages 30–35, Nara, Japan, April 1998.

[9] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang. A novel anomaly detection scheme based on principal component classifier. In *IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with ICDM'03*, pages 171–179, Melbourne, Florida, USA, November 2003.

[10] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang. Principal component-based anomaly detection scheme. In *Foundations and Novel Approaches in Data Mining*, volume 9, pages 311–329. T.Y. Lin, S. Ohsuga, C.J. Liau, and X. Hu, editors, Springer-Verlag, 2006.

[11] M.-L. Shyu, K. Sarinnapakorn, I. Kuruppu-Appuhamilage, S.-C. Chen, L. Chang, and T. Goldring. Handling nominal features in anomaly intrusion detection problems. In *the 15th International Workshop on Research Issues on Data Engineering (RIDE-SDMA'2005), in conjunction with ICDE 2005*, pages 55–62, National Center of Sciences, Tokyo, Japan, April 2005.

[12] S.Wold. *Pattern Recognition*. 8th edition, 1976.

[13] UCI. UCI KDD Archive. *http://kdd.ics.uci.edu/*, 2005.

[14] Weka. *http://www.cs.waikato.ac.nz/ml/weka/*.

[15] Z. Xie, T. Quirino, M.-L. Shyu, S.-C. Chen, and L. Chang. A distributed agent-based approach to intrusion detection using the lightweight PCC anomaly detection classier. In *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC2006)*, pages 446–453, Taichung, Taiwan, R.O.C., June 5-7 2006.

[16] N. Ye and Q. Chen. An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems. *Quality and Reliability Eng.*, 17(2):105–112, 2001.

[17] N. Ye, S. Emran, Q. Chen, and S. Vilbert. Multivariate statistical analysis of audit trails for host-based intrusion detection. *IEEE Transactions on Computers*, 51(7):810–820, July 2002.