NEURO-FUZZY CREDIT RISK FORECASTING

Eysteinn Finnsson*, Wuyang Duan*, Siddharth Ravi*

Abstract
We propose a novel method for predicting defaults on mortgage loans. The method utilizes a combination of Fuzzy C-means (FCM) clustering and an Adaptive Neuro-Fuzzy Inference System (ANFIS) in order to produce accurate and intuitive predictions. A SMOTE technique was used to increase the number of underrepresented data in the dataset. A range of dimensionality reduction techniques were investigated for the project. The best results were achieved by using Principal Component Analysis (PCA) or sparse PCA coupled with the FCM and ANFIS network. Using sparse PCA yielded the highest sensitivity of 99.99% and using PCA gives the highest precision of 99.95% from cross-validation.

Keywords
Classification, Machine Learning, Fuzzy Logic, Fuzzy C-means Clustering , Neural Networks, ANFIS, Credit Risk, Freddie Mac Dataset, SMOTE, Principal Component Analysis, Sparse Principal Component Analysis

* *Delft Center for Systems and Control, TU Delft*

Contents

Introduction

This report is a final report for the course IN4015 Neural Networks (2015-2016) done as a collaboration between TU Delft and EY Amsterdam.

A default is defined as when a borrower fails to meet his payment obligations. Predicting the risk of a credit default is an integral part of modern banking. Banks hold reserve capital to cover for losses from defaults. It is therefore in their interest to optimize the amounts held, since the other holdings could be used for profitable investments. On the other hand, lesser amounts may mean that the bank would be led to insolvency as they would be unable to meet all their debt obligations. Hence credit risk forecasting is integral to balancing risks and rewards of building capital. The Revised Framework on International Convergence of Capital Measurement and Capital Standards (BASEL II) allows an *"internal ratings based approach"* to predict borrowers' risk of default [1]. Traditionally defaults are predicted by discriminant analysis measures such as the Altman Z-score model, linear regression, logit and probit models. The financial crisis of 2007-2008 prompted a re-evaluation of the credit risk management process [2].

The task of this project is to create an alternate system that will improve and facilitate accurate and precise prediction of defaults for mortgage loans. The authors propose a Neuro-Fuzzy combination system as a viable solution. Fuzzy systems have the advantage of being highly readable and intuitive. The drawbacks of the fuzzy systems generally include time-consuming training and relatively poor performance compared to other methods. Artificial Neural Networks (ANN) on the other hand generally outperform other classification methods by creating a highly nonlinear model to fit the data. The blackbox nature of ANN however makes it difficult to understand the intricacies of the model and makes it difficult to modify once it has been trained [3]. A hybrid system that utilizes the best of both worlds is in this case the optimal solution. A clustering algorithm will weed out individuals that obviously falls in to a group that has previously defaulted while the ANN will take care of classifying those cases which

come out as ambiguous. The Adaptive Neuro-Fuzzy Inference System (ANFIS) architecture is preferred over other ANNs for its quality of being relatively readable while introducing the high levels of non-linearity necessary.

Previous Work

In 2014-2015 Swastanto, et. al [3] took on the same task of default prediction using Extreme Learning Machines (ELM) as the classification system and a Genetic Algorithm (GA) for feature selection [4] [5]. This method yielded good performance predicting the Current Loan Delinquency Status (CLDS) with a sensitivity and precision of 0.8103 and 0.9996 respectively. This report builds upon the work done by Swastanto, most importantly the dataset see Section 1, that has been parsed, normalized and over-sampled, is used for both training and validation on the Fuzzy-Neuro system.

Related Content

Asogbon,et. al in [6] presented a hybrid decision support system using a Neuro-Fuzzy system. The authors used several fuzzy variables Such as Employment Type, Civil Status, and Nature of Occupation that were mapped onto antecedents that assigned a membership degree to each variable (low, moderate, or high). The fuzzy rule base was populated with 30 rules based on experts' knowledge in the field of mortgage loans administration. The method turned out to be successful. Yielding an overall average prediction accuracy of 95.9%. These results are however hard to replicate as the expert rules are not provided for reference. Aida in [7] presents credit risk analysis with K-Nearest Neighbor (K-NN) classifier to predict the defaults in short term loans for a Tunisian commercial bank. The system was trained on a relatively small dataset of 924 loans and the classifier's accuracy was at best 88.63%. There has been considerable interest in using Support Vector Machine (SVM) approaches to financial problems [8]. Wang combined fuzzy set theory and SVM creating a hybrid system to discriminate good creditors from bad ones. The system improved the generalization ability and in some cases the performance of a standard SVM system [9] demonstrating the merits of a hybrid fuzzy systems for classification.

1. Data

The data used for the project is derived from the Freddie Mac Single Family Loan-Level dataset ®, which contains information on fixed rate mortgages in the United States over a thirty year period [10]. The dataset comprises several columns representing different features. These features are reduced to the ten features as presented in Table 3.

1.1 Data Pre-Processing

The data is pre-processed with a Synthetic Minority Over-Sampling Technique (SMOTE) [11] in order to increase the amount of under-represented groups in the dataset. This familiarizes the classifier system to all types of data samples. In SMOTE the minority class is over-sampled by creating "synthetic" examples rather than by over-sampling with replacement or duplication. Swastanto in [3] showed that in this case SMOTE was a superior method of oversampling and yield improved performance when applied to the ELM training data. The algorithm behind SMOTE is seen in Algorithm 4. The total numbers of default data and non-default data within the original and SMOTE'd dataset are seen in Table 1 and 2.

Table 1. Original dataset

	Total	70% training	30% validation
Default	10398	7278	3120
Non Default	194436	136105	58331

Table 2. SMOTE dataset

	Total	70% training	30% validation
Default	41592	29112	12480
Non Default	186563	130154	56409

2. Classifier Systems

2.1 Adaptive Neuro Fuzzy Inference System

Adaptive Neuro Fuzzy Inference System (ANFIS) is a hybrid learning algorithm combining a fuzzy inference system and a neural network [12]. The goal of the system is to train parameters in the antecedents and consequents to form Takagi-Sugeno (TS) type inference rules [13]. This is important since it ensures readability of the output and ensures that the system will be transparent, while preserving the inherent non-linearity present in a classical neural network. An example of a Takagi Sugeno type inference rule is shown below:

$$\text{If velocity is high, then } force = K \cdot (velocity)^2$$

The system utilizes the learning ability of the ANN to tune the membership functions of TS rules automatically. This, when coupled with the knowledge generalizing ability of the fuzzy inference system provides a capable system for credit risk assessment. The structure of ANFIS is shown in the Figure 1.

ANFIS consists of five different layers:

- Layer 1: Linguistic term with membership functions
- Layer 2: T-norm operation that performs a fuzzy AND operation
- Layer 3: Normalization of weights
- Layer 4: Output of each layer defined by consequent parameters
- Layer 5: Summation block

Table 3. Explanation of the features in the reduced dataset. 1. Credit Score is not used since it is acquired using an alternative risk assessment model. 6. Current default status is used as the output feature.

Feature	Explanation
1. Credit Score	Numerical creditworthiness of a person based on expert analysis.
2. Combined LTV	Loan-to-value (LTV) ratio of mortgage in combination with LTV of other mortgages (if any)
3. DTI Ratio	Debt-to-Income ratio represents percentage of income that goes towards paying debts
4. LTV	Loan-to-value ratio represents the ratio of loan to the total value of the asset purchased
5. Interest rate	The Fixed interest rate at the beginning of the loan period
6. Current default status	1 means the loan is currently in default during the final year, 0 otherwise
7. Default count	The total number of times the borrower defaulted during the five years of the loan
8. Mean CLDS	The average, CLDS value of the borrower during the five years of the loan
9. Standard deviation CLDS	The standard deviation of CLDS value of the borrower during the five years of the loan
10. UPB value	Unpaid Principal Balance expresses the loan provided at the start of the mortgage.

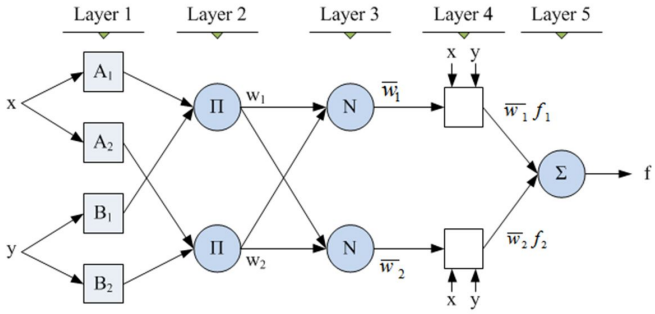


Figure 1. Adaptive Neuro-Fuzzy inference system.

The nodes in the first layer compute the membership degree of the inputs in the antecedent fuzzy sets. The product nodes Π in the second layer represent the antecedent conjunction operator. The normalization node N and the summation node Σ realize the fuzzy-mean operator.

The system utilizes a hybrid learning technique that makes use of a combination of least squares method and back - propagation to tune the parameters in the ANFIS structure. The consequent parameters of TS rules are tuned by a least square method and the parameters of antecedents of TS rules are optimized using a back propagation method. This is so as to preserve speed of training while at the same time retaining capability of convergence to a minima.

2.2 Fuzzy C-means Clustering

Clustering is the process of defining regions that contain data with similar characteristics. The difference between fuzzy clustering such as Fuzzy C-Means (FCM) and crisp clustering such as (K-NN) is that in FCM every data point has a degree of membership to every cluster [14]. The FCM functional cost function is represented as:

$$J(Z;U,V) = \sum_{i=1}^{c}\sum_{k=1}^{N}(\mu_{ik})^m ||z_k - v_i||_A^2 \qquad (1)$$

The value of the cost function can be seen as a measure of the variance of z_k from v_i. The minimization of the FCM functional represents a nonlinear optimization problem solved by iterative procedures, genetic algorithms or simulated annealing. The FCM algorithm used during classification can be seen in Algorithm 3.

The FCM algorithm converges to a local minimum of the c-means functional. Hence, different initialization lead to different results. A singularity in the FCM also occurs when $D_{ikA} = 0$ for some z_k amd one or more v_i (Refer algorithm 3 for explanation of terms). When this occurs, a zero membership is assigned to the clusters with $D_{ikA} > 0$ and memberships are distributed arbitrarily among the clusters for which $D_{ikA} = 0$,

Although there exist other types of fuzzy clustering algorithms such as the Gustafson-Kessel (GK algorithm), this report analyzes the FCM algorithm for fast and precise classification [15]. The one notable difference between the FCM and GK algorithm is that the GK algorithm utilizes ellipsoidal partitions that can vary in shape, while the FCM utilizes circular partitions.

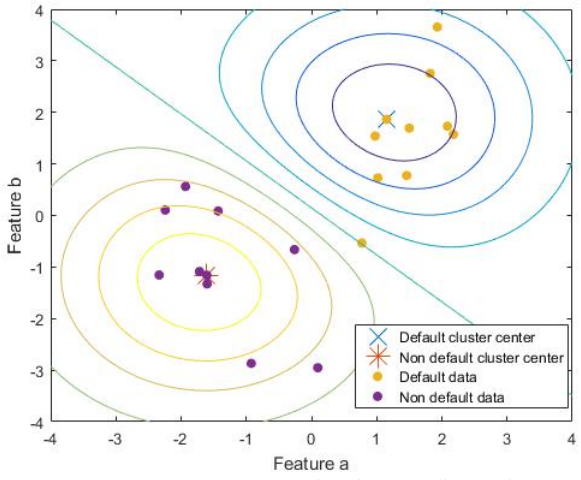


Figure 2. Fuzzy C-means clustering in two dimensional space using one cluster per group.

Algorithm 1 Clustering classifier procedure

Create default and non-default clusters from training data
Introduce subject p
distance from default cluster (D.F.D.C) = ||Default Cluster Centers - p||$_2$
distance from non-default cluster (D.F.N.D.C) = ||Non Default Cluster Centers - p||$_2$
if |1/min(D.F.D.C) - 1/min(D.F.N.D.C)| $\leqslant$ Border Tolerance **then**
    Send to NN
end if

2.3 Hybrid Classifier System

As previously outlined, the primary motive of this report is to obtain a transparent and precise credit risk assessment system. The system should be able to determine whether or not a customer will default on a loan based on the customer history, while at the same time encouraging accountability by being able to put forward reasons towards why a particular decision was made. Since the ANN does not provide enough transparency while being able to give sufficient nonlinearity, ANFIS is utilized to make the best of both worlds.

Although an effective solution, the ANFIS network takes a large amount of time to train and is also computationally intensive. This makes the process difficult to tune and therefore imposes a lot of constraints for the 'trainer' of the network. To alleviate this situation, an alternative solution was suggested, which was to make use of fuzzy clustering in cascade with the ANFIS. The clustering can determine the default status of borrowers who based on previous information will obviously fall into a category of default or non-default. The advantage of fuzzy classifiers are that they are computationally inexpensive when compared with the ANFIS system and can eliminate the obvious cases and leave ambiguous cases to be decided by the highly nonlinear fuzzy neural network. The process is illustrated in Figure 3. The entire algorithms is as seen in Algorithm 2. This method provides improved learning speed as well as the possibility of capturing new and valuable dynamics while still keeping the process human-readable and intuitive.

With clustering, objects (data points) are collected into bins with similar traits. When applying the method to in the hybrid system an individual p is introduced. If p is sufficiently close to a cluster center it is assumed that p has the characteristics of that cluster (i.e. default or non-default) and classification is done. In other words, data from the same class falls in similar regions in the feature space. The clustering classification breaks down at the boundary regions of two or more clusters (see Figure 2), so if an individual falls between clusters he can not be reliably classified using this method. The individual is then forwarded to a Neural Network (ANFIS) that is specialized for classification on the cluster boundary regions. Conceptually, the generalization ability of ANFIS is not as good as a traditional neural network. Thus the

Algorithm 2 Overall Algorithm for training and validation

Step 1: Get data from dataset
Step 2: Split data 70:30 for training:validation, use SMOTE on both sets separately.
Step 3: Use PCA, Sparse PCA or feature selection to reduce dimensionality and simplify model.
Step 4: Train clustering classifier (create clusters) using the reduced training data.
Step 5: Run the reduced training data through the clustering classifier and use the data that is deemed ambiguous to train ANFIS.
Step 4: Feed reduced validation data into fuzzy classifier, find obvious default/non-default cases.
Step 5: Feed ambiguous cases into ANFIS for further classification.

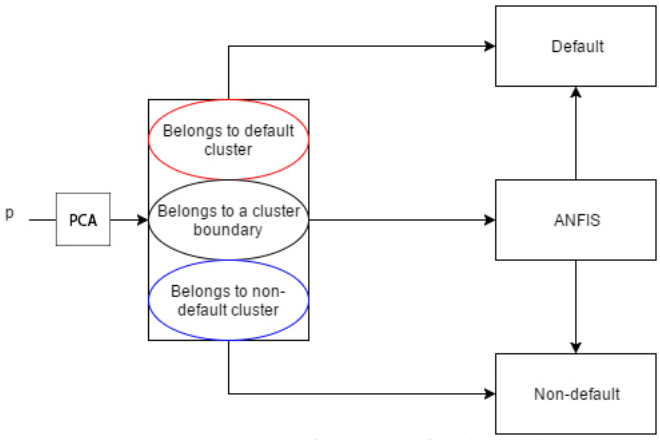


Figure 3. A block diagram of the classification. p is an individual that is to be classified.

ANFIS does not have to learn different characteristics within the dataset making the task easier.
The advantages of this method are that it is very intuitive and fast to train while yielding consistently good results. The ANFIS network can be specialized to recognize the individuals on the cluster boundaries thus reducing the training time and increasing accuracy. The accuracy here is defined as the total number of correct prediction divided by the total number of outputs.

3. Dimensionality Reduction

In theory it would be desirable to include as many features as possible when training the hybrid system. This way the system could extract more information from the training data and ideally lead to more accurate performance. However, using a large pool of features can be troublesome in practice. For instance, training the ANFIS using all ten features is very slow. Similarly, using more features does not seem to correlate to a better clustering performance. There are multiple ways to reduce the number of features, and this report compares

and contrasts three of them, namely feature selection [16], Principal Component Analysis (PCA) [17] and Sparse PCA [18].

3.1 Feature Selection

Feature selection makes use of algorithms such as genetic algorithms to select the most relevant features in the dataset [16]. It is a simple technique providing shorter training times as well as making the system easier to read.

3.2 Principal Component Analysis (PCA)

PCA a way of preserving information while reducing dimensionality and thus the computational intensity [17]. It achieves this through a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of uncorrelated variables called principal components. The number of principal components should be fewer than the original number of features thereby reducing training times required. PCA is mathematically defined as an orthogonal linear transformation that transforms the data into another new coordinate system where the greatest variance by some projection of the data comes on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, etc. The transformation is defined by a set of p-dimensional vectors of weights or loadings $w_k = (w_1, \ldots, w_p)_k$ that maps each row vector x_i of X to a new vector of principal component scores $t_i = (t_1 \ldots t_k)_(i)$ given by:

$$t_{k(i)} = x_i * w_k \tag{2}$$

in such a way that the individual variables of t considered over the data set successively inherit the maximum possible variance from x, with each loading vector w constrained to be a unit vector. For example the first principal component has to satisfy:

$$w_{(1)} = argmax_{||w||=1} \frac{(w^T X^T X W)}{(w^T w)} \tag{3}$$

3.3 Sparse PCA

The downside of PCA is that it mixes up the input features and reduces readability. Sparse PCA on the other hand introduces a sparsity constraint on the input variables. While ordinary PCA introduces linear combinations of *all* input variables, the sparse PCA [18] overcomes this disadvantage by introducing linear combinations of just a few input variables.

Mathematically, if $\Sigma = x^T x$ is the empirical covariance matrix of X, which has dimension p*p, given an integer k with $1 \leqslant k \leqslant p$, the sparse PCA problem could be formulated as maximizing the variance along a direction represented by vector $v \in R^p$ while constraining its cardinality:

$$max(v^T \Sigma v) \tag{4}$$

subject to $||v||_2 = 1$ and $||v||_0 \leqslant k$ The first constraint specifies that v is a unit vector. The second constraint represents the L0 norm of v, which is defined as the number of its non-zero components. So the second constraint $||v||_0$ specifies that the number of non-zero components in v is less than or equal to k, which is typically an integer that is much smaller than dimension p.

4. Process and Results

4.1 Feature Selection+Clustering+ANFIS

The method that preserves the most readability is feature selection. The feature set has eight different features. This yields 255 different possible combinations. Since the clustering is relatively fast it is possible to try all possibilities. A reward function, see Equation 5, is created to have a quantitative measure of the performance of the clustering.

$$Reward = \frac{(Correct + SendToNN \cdot 0.80)}{(Correct + SendToNN + False)} \tag{5}$$

	Clustering definitions
FCM	Fuzzy C-means
CD	Number of default clusters
CND	Number of non default clusters
Correct	Correctly classified by the FCM classifier
False	Incorrectly classified by the FCM classifier
SendToNN	If a data point falls between clusters it is forwarded to a Neural Network. The tolerance of what counts as between is based on the Border Tolerance
Border Tolerance	A measure of how skeptical the FCM classifier is. A high border tolerance will lead to more data points being forwarded to the Neural Network.

The poor performance of 80% is assumed for the Neural Network. This way it is possible to avoid the solutions that dump a lot of data into the Neural Network. A nested for-loop is created that circles through the 255 different input features as well as trying every different combination of CD and CND up to six clusters each.

To reduce the search space the tolerance for selecting ambiguous cases (i.e. Border Tolerance) is kept constant. Furthermore the FCM options are kept constant: Fuzzy partition matrix exponent is at its default value of 2. The clustering will stop when the number of iterations reaches a maximum of 100 or when the objective function improves by less than 0.001 between two consecutive iterations.

Since the problem becomes exponentially harder when more clusters are used it is necessary to reduce the search

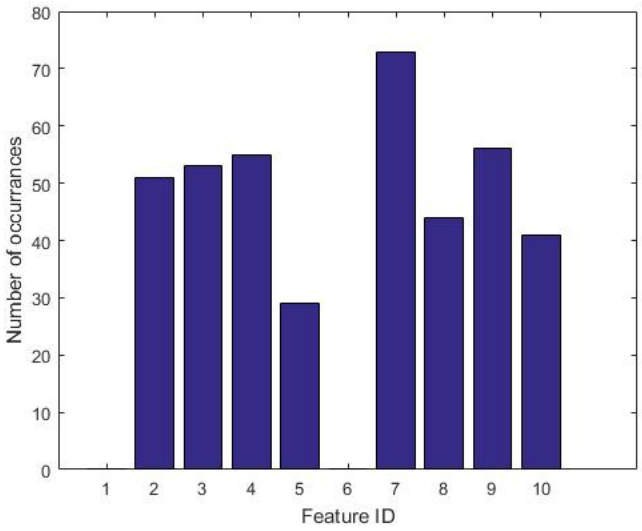


Figure 4. Subset of features that result in Reward > 0.80 and number of False < 100.

space in order to increase the number of clusters. A look-up table is created with the parameters and corresponding Rewards of the first iteration as described above. The solutions with a Reward > 0.80 and number of False < 100 are extracted from the look-up table, see Figure 4, and the subset of features that lead to these solutions used for the next iteration. The procedure is now repeated using 1 - 12 clusters.

This experiment concluded the following optimal result using a validation dataset of 2000 data points.

Features	[4 7 9]
Border Tolerance	4
CND	2
CD	9
Reward	0.87770
Correct	1173
False	99
SendToNN	728

Feature selection for ANFIS is not possible using a brute force method. Therefore the five most prominent features from the clustering [2 3 4 7 9] are used, see Figure 4. This is probably not the optimal feature selection for the ANFIS. Running the combined system using the full SMOTE dataset results in the performance seen in Table 5 and Figure 5. The precision is defined as $\frac{TP}{TP+FP}$ and sensitivity is as $\frac{TP}{TP+FN}$. They are both statistical measurements of the performance of the classification system when there is imbalance within the data set. The confusion matrix is shown in Table 4.

Table 4. Confusion matrix

	Predicted Positive	Predicted Negative
Actual Postive	TP	FN
Actual Negative	FP	TN

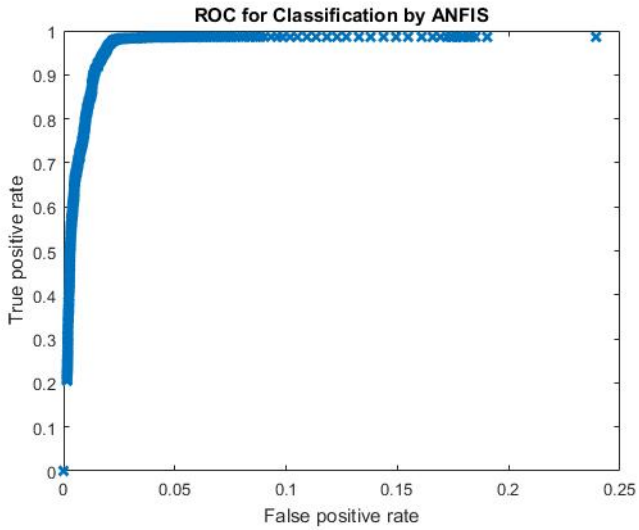


Figure 5. ROC curve for system performance using feature selection.

4.2 PCA+Clustering+ANFIS

From the previous section it could be understood that the best performance using a feature selected model and clustering incorporated with ANFIS is not obtained. The selected original features cannot evade the issue of overlapping between default and non default data. Nor does the ANFIS network handle the overlapping between two classes effectively. This implies that there are correlated features within the original feature set.

To tackle this problem, principal component analysis is introduced before the steps of clustering and training of ANFIS. The PCA applies a singular value decomposition to the co-variance matrix of the input data set. The acquired singular values can be checked to find the first several biggest ones, which indicate the direction of components with largest variances. Usually the first several components are much larger than the rest. This means the corresponding singular vectors need to be taken as the projection matrix, which is then multiplied to the input data set. In this way, the input data is projected onto a lower dimensional space. Each axis is a principal component of the input features and they are not correlated with each other. These data can then be used to implement fuzzy c-means clustering and ANFIS.

4.2.1 Original dataset

The original feature dataset without any oversampling process is used here. 70% of the default and non default data are both taken as the training data. The remaining 30% of two classes of data are used for validation. In this way, the original proportion of default and non default within the dataset is kept.

Table 5. Performance of classifier system using feature selection. SMOTE dataset.

Parameters: cd = 9, cnd = 2, threshold = 4, threshold_anfis = 0.5			
FCM Features = [4 7 9], ANFIS Features = [2 3 4 7 9]			
Wrongly classified by clustering	4354	default data closer to non default clusters(FN)	1673
Correctly classified by clustering	38432	non default data closer to default clusters(FP)	38815
Ambiguous data sent to ANFIS	26103	Overall prediction accuracy	93.54 %
Wrongly classified by ANFIS	784	Precision	89.26 %
Correctly classified by ANFIS	25319	Sensitivity	75.87 %

From the result of the PCA, the first three singular values are dominant while the last four is of slightly lower value. Since 3 principal components are easy to visualize, they are firstly picked to implement the our two layer classifier system. The classification result is shown in table 6.

n is the number of principal components used, *cd* is the number of default clusters and *cnd* is the number of non default clusters. *threshold* (i.e. Border Tolerance) is the parameter taken to decide which data are ambiguous and sent to the ANFIS as mentioned in the last section. The threshold of ANFIS is taken as 0.5 which would transform the continuous output of ANFIS into binary labels. The choice of *cd*, *cnd* and *threshold* are tuned to yield the best performance as shown in the Table 6 With only three principal components used, there is an obvious overlapping between the two classes since nearly 2500 non default data points are closer to the default cluster centers, which indicates FP. The performance in terms of sensitivity can also still be improved. The fourth principal component is thus added to see if there is any improvement.

Using four principal components, both the performances of clustering and ANFIS are improved with less wrong classifications. The precision and sensitivity are also both improved. Although it would be difficult to visualize the projected dataset, we decide to stick with the choice of four principal components, this being a logical choice according to the system performance.

4.2.2 SMOTE dataset

Mind that there are only 5% of default data within the original feature set, so an overall accuracy of 99% is not very impressive. And it is very possible that the system is not trained sufficiently because of the minority of default data. For this reason, the SMOTE data is further used to check the performance of the system. As mentioned in the section on data pre-processing, the SMOTE technique expands the original data around the neighborhood. In this case, four times more of the minority data is generated, which increases the default data to around 20%. The two layer classifier system is trained and validated again. The performance is shown in table 8. Note that the validation dataset is also (separately) processed with the SMOTE technique to make sure that there are four times more default data. This makes the performance measured by percentage more reliable, since it avoids possible correlations from occurring between the two (training and validation) datasets.

Comparing table 7 and table 8, both precision and sensitivity are increased. Since there are more default data, the classifier system can better generalize the structure and default data. Hence it is deemed preferable to stick with the SMOTE dataset. Since the first layer of clustering gives all correct prediction, the performance of ANFIS is checked at this point. The comparison of ANFIS output and real label are shown in Figure 6. The ROC curve is seen in Figure 7. Note that the validation dataset is totally unused before the validation phase. The Receiver Operating Characteristic (ROC) curve indicates excellent performance considering both precision and sensitivity.

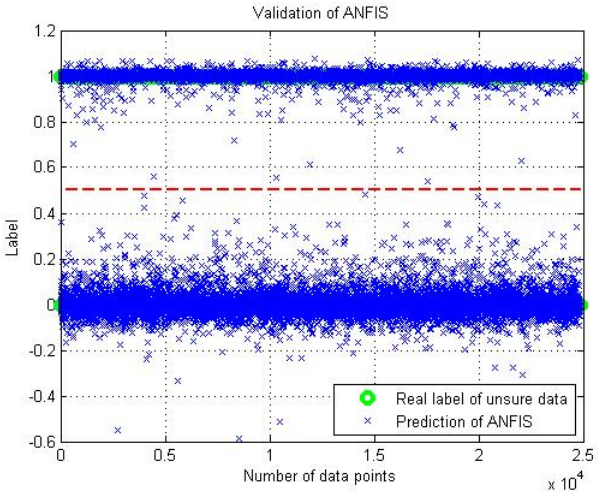


Figure 6. Predicted output of the ANFIS on SMOTE dataset

4.2.3 Visualization

With the trained classifier system available, the clustering and the ANFIS are visualized. The clustering on the projected dataset $(PCA\ applied)$ is firstly visualized. With PCA applied, the principal components are now well separable as shown in Figure 16a and Figure 16b in the Appendix. This confirms our motivation that the PCA transforms original features into uncorrelated principal components. The green data points are the ambiguous data that cannot be clearly decided by clustering and are sent to ANFIS. They are mainly located outside the default and non default clusters and the intermediate regions between the default and non default clusters. When training the ANFIS, two membership functions of high and low are chosen. This yields $2^4 = 16$ fuzzy TS rules. The first inference rule of the ANFIS is visualized here. To for-

Table 6. Performance of classifier system using 3 principal components

Parameters: n = 3, cd= 6, cnd = 3, threshold = 4, threshold_anfis = 0.5			
Wrongly classified by clustering	111	default data closer to non default clusters(FN)	41
Correctly classified by clustering	42450	non default data closer to default clusters(FP)	2516
Ambiguous data sent to ANFIS	18890	Overall prediction accuracy	99.65%
Wrongly classified by ANFIS	107	Precision	93.19%
Correctly classified by ANFIS	18783	Sensitivity	89.47%

Table 7. Performance of classifier system using 4 principal components

Parameters: n = 4, cd= 6, cnd = 3, threshold = 4, threshold_anfis = 0.5			
Wrongly classified by clustering	0	default data closer to non default clusters(FN)	14
Correctly classified by clustering	39681	non default data closer to default clusters(FP)	1284
Ambiguous data sent to ANFIS	21770	Overall prediction accuracy	99.98%
Wrongly classified by ANFIS	8	Precision	99.82%
Correctly classified by ANFIS	21762	Sensitivity	99.48%

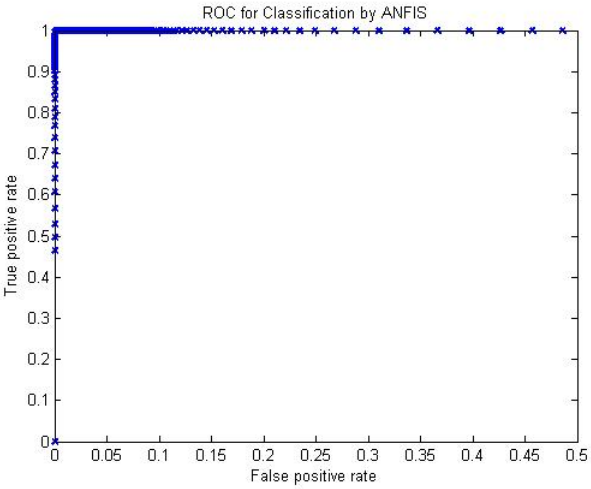


Figure 7. ROC curve of ANFIS on SMOTE dataset

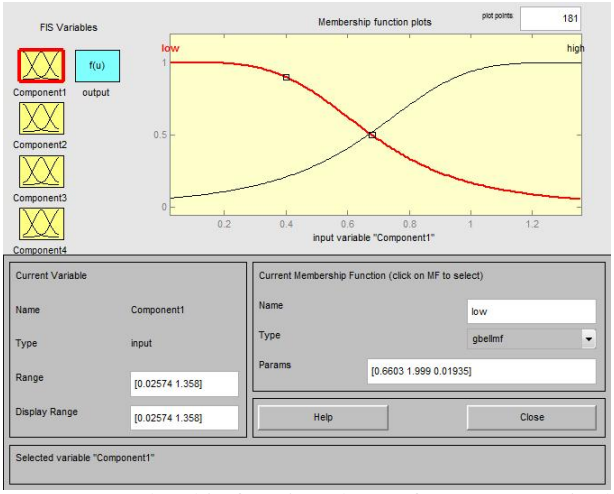


Figure 8. Membership function "low" of component 1 in the first rule

mulate the first rule in a specific Takagi-Sugeno form, the parameters of consequents shown in Figure 9 are used. The rule looks like as follows:

If c_1 is low, c_2 is ..., c_3 is ..., c_4 is ..., then f_1 is
$-0.3827c_1 + 0.5468c_2 + 0.00073c_3 + 0.04894c_4 + 0.8826$

Here $c_1 \cdots c_4$ denotes the four principal components. How the lingustic term *low* for the first principal component is defined is shown in Figure 8. The form of TS rule is straightforward. However from the inference rules, the readability of the system is not clear when the PCA is used. The PCA increases our performance in clustering and ANFIS but decreases the overall readability. A proper trade-off needs to be determined.

4.3 SPCA+Clustering+ANFIS

Alternatively, the sparse PCA can be implemented in the data pre-processing. Since sparsity is introduced in the projection matrix, every principal component would be a linear combination of fewer original features. Every one of the principal components are constructed as a combination of a constrained number of features, namely two in number.

4.3.1 Original dataset

Consider the original dataset. There are 8 features available upon omission of credit score. If the number of principal components are chosen as two and the cardinality of every component is two, the projection matrix (6) of default training data is acquired.

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.9407 & 0 & -0.3394 \\ 0.8155 & 0 & 0 & 0 \\ 0.5788 & 0 & 0 & 0 \\ 0 & 0 & 0.691 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.7229 & 0 \\ 0 & 0.3394 & 0 & 0.9407 \end{bmatrix} \tag{6}$$

If the original feature set is multiplied by this projection matrix, the first principal component is then simply $0.8155 \cdot$

Table 8. Performance of classifier system using 4 principal components. SMOTE dataset

Parameters: n = 4, cd= 6, cnd = 3, threshold = 4, threshold_anfis = 0.5			
Wrongly classified by clustering	0	default data closer to non default clusters(FN)	25
Correctly classified by clustering	44053	non default data closer to default clusters(FP)	872
Ambiguous data sent to ANFIS	24836	Overall prediction accuracy	99.99%
Wrongly classified by ANFIS	8	Precision	99.95%
Correctly classified by ANFIS	24828	Sensitivity	99.95%

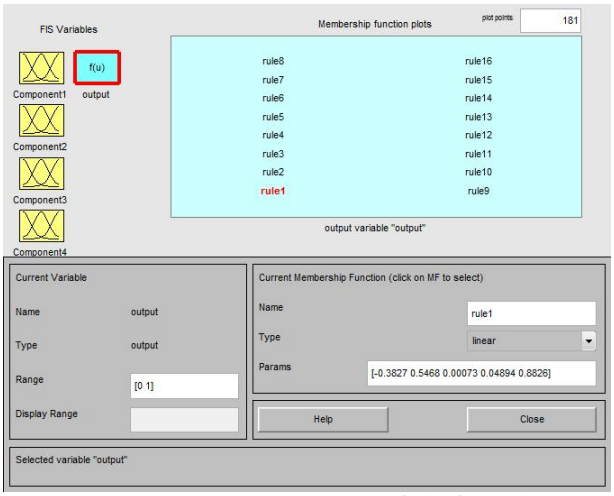


Figure 9. Output parameters of the first rule

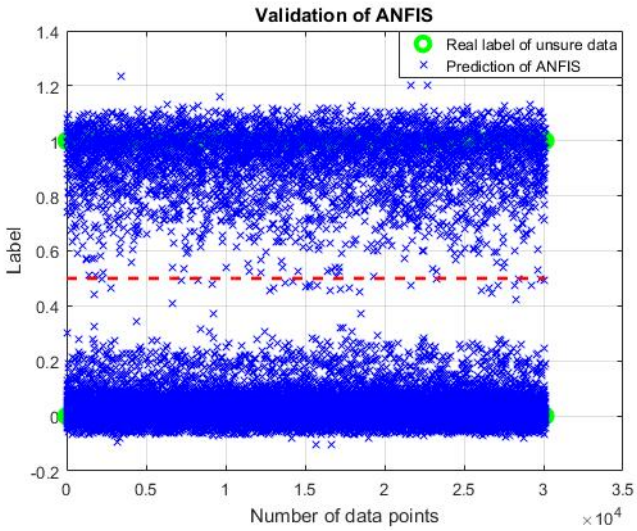


Figure 10. Predicted output of the ANFIS on SMOTE dataset

$LTV + 0.5788 \cdot Interest\ rate$. Note that the original features are all normalized. Using the sparse principal components, the two layer classifier system is trained and validated. The validation result is shown in Table 9.

Comparing to the previous case where PCA is used, the data points located closer to the other class which indicates FN and FP are decreased significantly. This would provide better performance than PCA and ANFIS.

4.3.2 SMOTE dataset

For the same reason mentioned in Section 4.2.2, the SMOTE dataset with four times more default data is used. Once again 70% of both default and non default data are used to train the classifier system and 30% remaining data are used for validation. The validation result is shown in Table 10.

Contrary to the PCA case, there is no obvious difference using the original dataset and SMOTE dataset. The sensitivity is seen to be only slightly improved. Since the clustering gives good performance, the performance of ANFIS is checked. The comparison of prediction by ANFIS and real label is shown in Figure 10 in the Appendix. The ROC curve is in Figure 11.

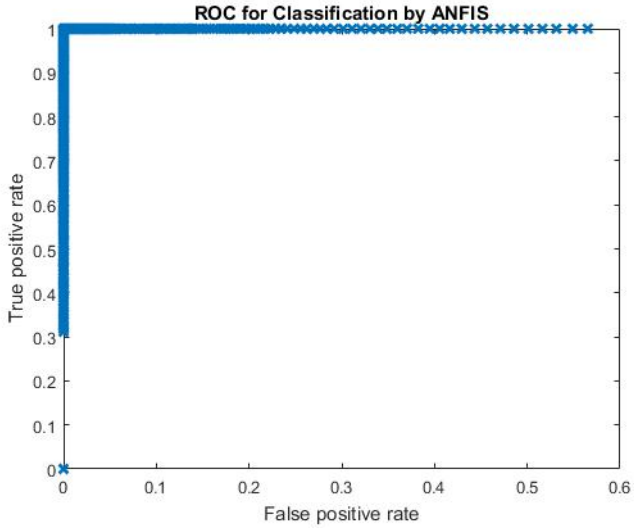


Figure 11. ROC curve of ANFIS on SMOTE dataset

From the ROC curve and the result shown in Table 10, the sparse PCA does not decrease the performance of ANFIS, compared to the PCA. Because every principal component is a combination of two original features, the acquired two layer classifier system keeps reasonable readability while yielding excellent performance.

Table 9. Performance of classifier system using 4 sparse principal components.

Parameters: n = 4, cd= 6, cnd = 3, threshold = 4, threshold_anfis = 0.5			
Wrongly classified by clustering	0	default data closer to non default clusters(FN)	2
Correctly classified by clustering	33517	non default data closer to default clusters(FP)	0
Ambiguous data sent to ANFIS	27934	Overall prediction accuracy	99.97%
Wrongly classified by ANFIS	17	Precision	99.99%
Correctly classified by ANFIS	27917	Sensitivity	99.27%

Table 10. Performance of classifier system using 4 sparse principal components. SMOTE dataset

Parameters: n = 4, cd= 6, cnd = 3, threshold = 4, threshold_anfis = 0.5			
Wrongly classified by clustering	0	default data closer to non default clusters(FN)	0
Correctly classified by clustering	38781	non default data closer to default clusters(FP)	1
Ambiguous data sent to ANFIS	30108	Overall prediction accuracy	99.96%
Wrongly classified by ANFIS	31	Precision	99.99%
Correctly classified by ANFIS	30077	Sensitivity	99.29%

4.3.3 Visualization

To provide a transparent and intuitive understanding of the classifier system, the clustering and ANFIS part are both visualized. The original dataset is multiplied by the projection matrix and transformed in the form of sparse principal components. The result of fuzzy c-means clustering using 3 of the principal components is presented in Figure 17a and Figure 17b. Note that all the data points are from the validation dataset. Plots of other sparse principal components are similar and they are thus not included in this report. Compared to PCA, the sparse PCA is already sufficient to decrease the correlations among the original feature set. The sparse principal components are also now more linear separable, which makes the clustering algorithm truly effective.

The ambiguous data points are marked as green. These cases are sent to the ANFIS for further analysis. The overall inference system can be visualized as in Figure 12. Every row represents one TS rule and every column represents the membership functions of the corresponding component. For example, the first entry of rule 1 in Figure 12 contains two membership functions, namely low and high. These membership functions define how the first sparse principal components are evaluated as high and low.

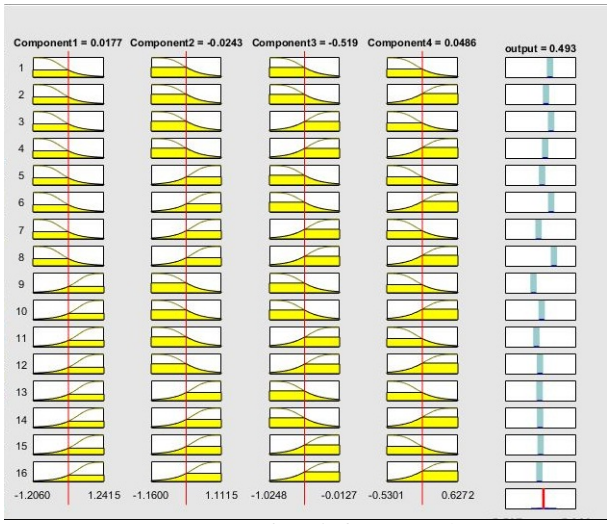


Figure 12. 16 TS inference rules.

The membership functions of the first rule and the parameters of the consequents are shown in Figure 13 and Figure 14 respectively.

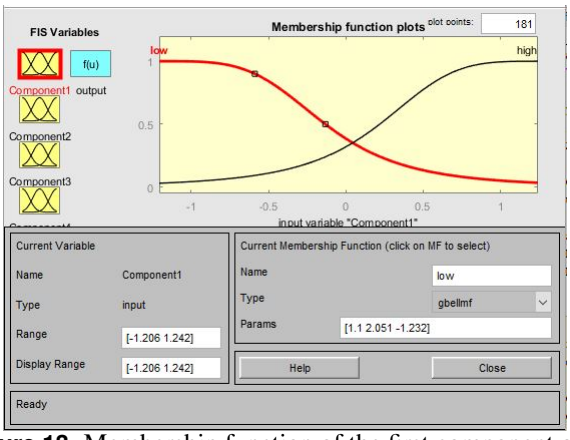


Figure 13. Membership function of the first component of the first rule.

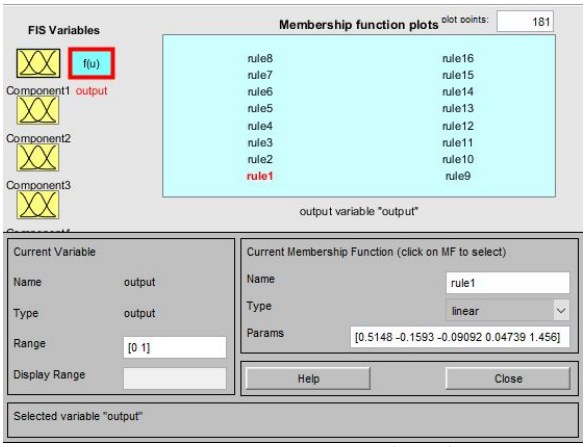


Figure 14. Output parameters of the first rule.

Similar to the projection matrix seen in equation 6, the projection matrix of the SMOTE dataset is calculated. How the original features are linearly combined is then clear. Now the first Takagi-Sugeno inference can be formulated in a specific way as shown in Table 11.

Table 11. The first TS inference rule

If	$c_1 = 0.8184 \cdot LTV + 0.5747 \cdot Interest\ rate$ is Low_1, $c_2 = 0.9842 \cdot DTI\ ratio + 0.1770 \cdot Interest\ rate$ is Low_2, $c_3 = -0.0449 \cdot Interest\ rate - 0.9990 \cdot UPB\ value$ is Low_3 and $c_4 = 0.5747 \cdot LTV - 0.8184 \cdot Interest\ rate$ is Low_4
then	the output f_1 is $0.5148 \cdot c_1 - 0.1593 \cdot c_2 - 0.0910 \cdot c_3 + 0.0474 \cdot c_4 + 1.456$

The remaining TS inference rules can be checked in the same way. They are not included here for the sake of conciseness. These 16 rules construct the ANFIS and maps the input sparse principal components into one single output. This mapping can be further visualized.

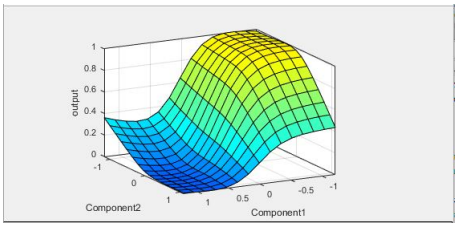
Considering the plots in Figure 15, every plot shows the mapping from two sparse principal components to the output. Contrary to the traditional black box model of neural network, the two layer classifier system can be completely visualized. It provides considerable insight into the reasoning behind the decision making process.

4.4 Comparison

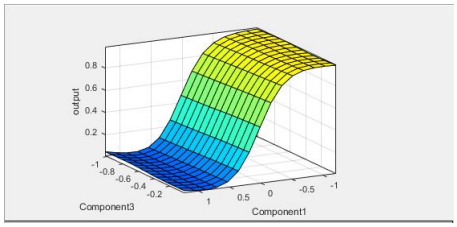
Applying SMOTE to the validation dataset and running it through the systems gives a comparison of the three dimensionality reduction techniques as is presented in table 12.

Table 12. Comparison between the three different systems.

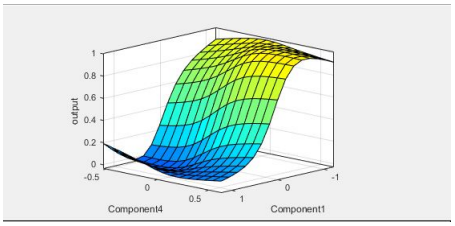
	Feature Selection	Sparse PCA	PCA
Transparency	High	Fair	Low
sensitivity [%]	75.87	99.99	99.95
Precision [%]	89.26	99.29	99.95



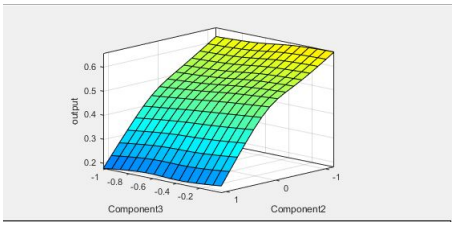
(a) I/O mapping of sparse principal component 1, 2



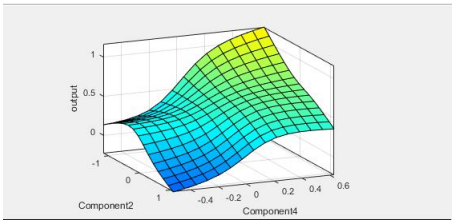
(b) I/O mapping of sparse principal component 1, 3



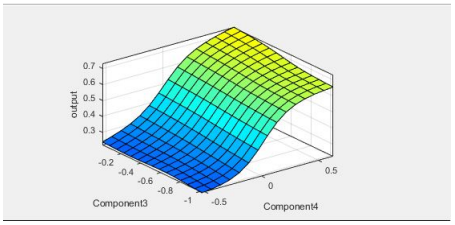
(c) I/O mapping of sparse principal component 1, 4



(d) I/O mapping of sparse principal component 2, 3



(e) I/O mapping of sparse principal component 2, 4



(f) I/O mapping of sparse principal component 3, 4

Figure 15. I/O mapping - sparse principal components

The Feature Selection+Clustering+ANFIS scheme gives the best transparency during the classification process. But the sensitivity is not good enough. This indicates that the False Negative prediction happens frequently. Default customers might be detected as non default customers, which is not in favor of the bank. The PCA+Clustering+ANFIS scheme gives the best overall accuracy, but the transparency and readability is poor. Every principal component used in clustering and ANFIS is a combination of all 8 features. The sparse PCA scheme balances the transparency and the accuracy. It yields very good performance with a reasonably transparent structure.

The choice among these three scheme depends on the needs of the user. If a totally transparent structure is necessary, the first scheme would clearly be the best choice. If the overall accuracy is what matters the most to the user, then PCA+Clustering+ANFIS would be the best. If the user requires both readability and accuracy for the classifier system, then the SPCA+Clustering+ANFIS would be the proper one.

5. Future Work

Future work would be geared towards the implementation of cluster indices such as the Xie-Beni index to find out the optimal amount of cluster partitions for the system [19]. At the expense of readability, in case higher nonlinearity and accuracy is preferred, other networks such as deep neural networks could be preferred over neuro-fuzzy systems. Because of the property of variable hyperplanar shapes Gustafson-Kessel clustering could be more effective than the fuzzy c-means algorithm, and this could also be investigated [15]. The effectiveness of Support Vector Machines could also be explored as an alternative for clustering techniques. The effectiveness of bagging and ensemble methods for classification could also be investigated, and the results compared [20].

References

[1] Basel Committee et al. An explanatory note on the basel ii irb risk weight functions. 2004.

[2] Edward I. Altman. Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *Journal of Finance: 189–209*, September 1968.

[3] M. Nayak B.A Swastanto, L. Xiaoran. Default prediction in mortgage loans. *Neural networks project report, TU Delft*, 2015.

[4] Guang bin Huang, Qin yu Zhu, and Chee kheong Siew. Extreme learning machine: Theory and applications, 2006.

[5] Peter; Keller Robert; Francone Frank Banzhaf, Wolfgang; Nordin. Genetic programming – an introduction. *San Francisco, CA: Morgan Kaufmann. ISBN 978-1558605107*, 1998.

[6] O. C. Agbonifo O. W. Samuel V. I. Yemi-Peters M. G. Asogbon, O. Olabode. Adaptive neuro-fuzzy inference system for mortgage loan risk assessment. *International Journal of Intelligent Information Systems*, 2016.

[7] K. A. Aida. Bank credit risk analysis with k-nearest neighbor classifier: Case of tunisian banks. *Accounting and Management Information Systems*, 14:79–106, 2015.

[8] Ning Chen, Bernardete Ribeiro, and An Chen. Financial credit risk assessment: a recent review. *Artificial Intelligence Review*, 45(1):1–23, 2016.

[9] Yongqiao Wang, Shouyang Wang, and K. K. Lai. A new fuzzy support vector machine to evaluate credit risk. *IEEE Transactions on Fuzzy Systems*, 13(6):820–831, Dec 2005.

[10] Freddie Mac ®. Single family loan-level dataset general user guide, 2016.

[11] K. W.; Hall L.O. Chawla, N. V.; Bowyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research 16 321–357*, 2002.

[12] J. S. R. Jang and Chuen-Tsai Sun. Neuro-fuzzy modeling and control. *Proceedings of the IEEE*, 83(3):378–406, Mar 1995.

[13] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-15(1):116–132, Jan 1985.

[14] Z. Ye and H. Mohamadian. Enhancing decision support for pattern classification via fuzzy entropy based fuzzy c-means clustering. In *52nd IEEE Conference on Decision and Control*, pages 7432–7436, Dec 2013.

[15] D. E. Gustafson and W. C. Kessel. Fuzzy clustering with a fuzzy covariance matrix. In *Decision and Control including the 17th Symposium on Adaptive Processes, 1978 IEEE Conference on*, pages 761–766, Jan 1978.

[16] André Guyon, Isabelle; Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research 3*, 2003.

[17] Jolliffe I.T. Principal component analysis. *Series: Springer Series in Statistics , 2nd ed., Springer, NY, XXIX, 487 p. 28 illus. ISBN 978-0-387-95442-4*, 2002.

[18] H. Zou; T. Hastie; R. Tibshirani. Sparse principal component analysis. *Journal of Computational and Graphical Statistics 15 (2): 262–286.*, 2006.

[19] K. Honda M. Muranishi and A. Notsu. Xie-beni-type fuzzy cluster validation in fuzzy co-clustering of documents and keyword. *Graduate School of Engineering, Osaka Prefecture University, 1-1 Gakuen-cho, Nakaku, Sakai, Osaka 599-8531 Japan*, 1998.

[20] R. Maclin D. Opitz. Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research 11, 169-198*, 1999.

Appendix - Algorithms and Visualization

Algorithm 3 Fuzzy c-means algorithm

Given the data set Z, choose the number of clusters $1 < c < N$, the weighting ex- ponent $m > 1$, the termination tolerance $\varepsilon > 0$ and the norm-inducing matrix A. Initialize the partition matrix randomly, such that $U^{(0)} \in$ Mfc.

Repeat for $i = 1, 2, ..$

Step 1: Compute Cluster Prototypes(means).

$$v_i^{(l)} = \frac{\sum_{k=1}^{N} (\mu_{ik}^{(l-1)})^m z_k}{\sum_{k=1}^{N} (\mu_{ik}^{(l-1)})^m}, 1 \leqslant i \leqslant c \tag{7}$$

Step 2: Compute distances.

$$D_{ikA}^2 = (z_k - v_i^{(l)})^T A (z_k - v_i^{(l)}), 1 \leqslant i \leqslant c, 1 \leqslant k \leqslant N. \tag{8}$$

Step 3: Update the partition matrix: for $i \leqslant k \leqslant N$

if $D_{ikA} > 0$ for all i = 1,2,...,c

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^{c} (D_{ikA}/D_{jkA})^{2/(m-1)}} \tag{9}$$

otherwise

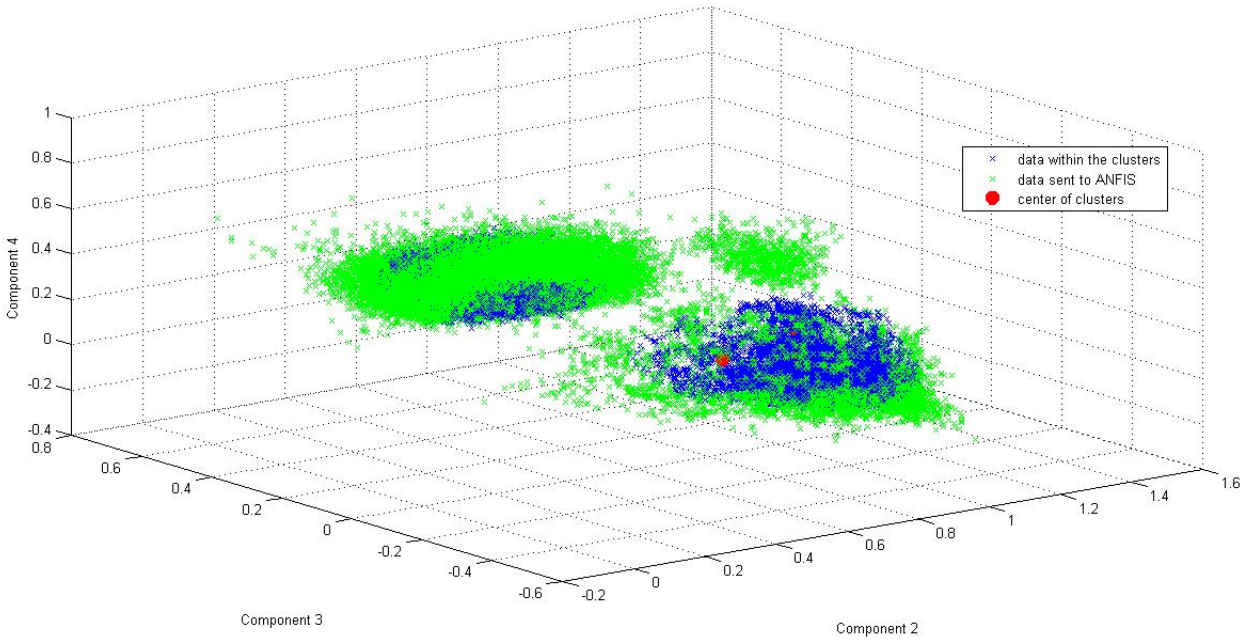
$\mu_{ik}^{(l)} = 0$ if $D_{ikA} > 0$ and $\mu_{ik}^{(l)} \in [0,1]$ with $\sum_{i=1}^{c} \mu_{ik}^{(l)} = 1$

until $\left|\left| U^{(l)} - U^{(l-1)} \right|\right| < \varepsilon$

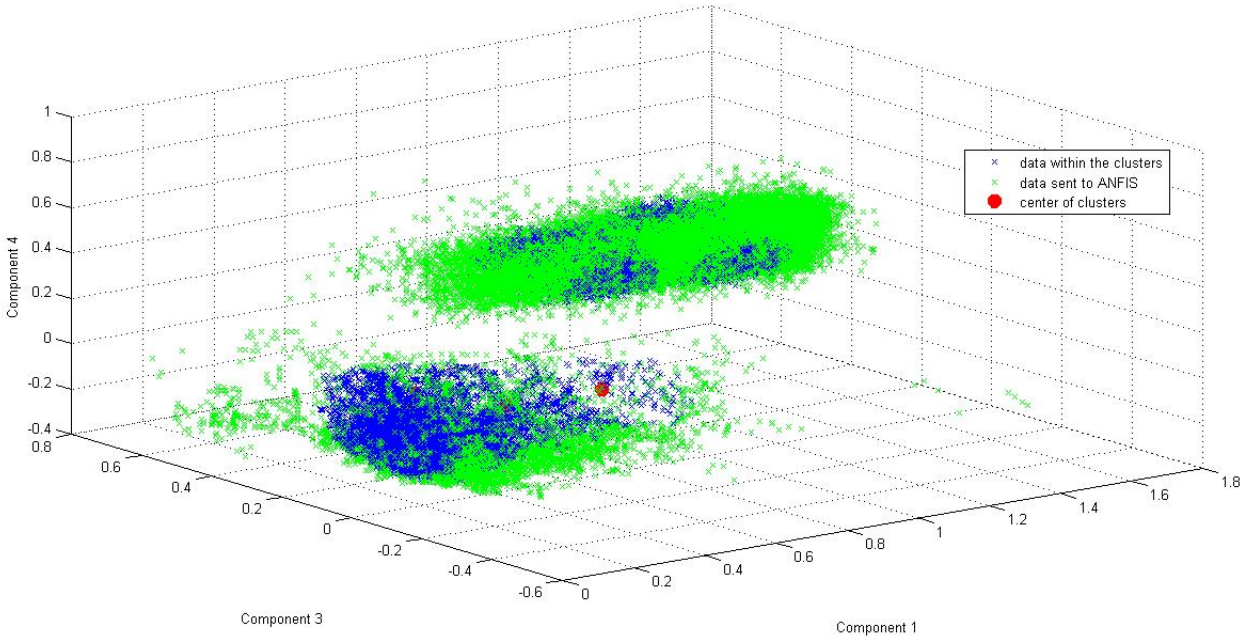
where $\mu_{ik}^{(l)}$ is a membership function of a fuzzy partition matrix of Z, V is a vector of cluster prototypes, D_{ikA}^2 is a squared inner product distance norm and U is the fuzzy partition matrix of Z.

Algorithm 4 Pseudocode for SMOTE

```
if N < 100 then
    Randomize the T minority class samples
    T = (N/100)*T
    N = 100
end if
N = (int)(N/100) (* The amount of SMOTE is assumed to be in integral multiples of 100. *)
k = Number of nearest neighbors
numattrs = Number of attributes
Sample[][]: array for original minority class samples
newindex: keeps a count of number of synthetic samples generated, initialized to 0.
Synthetic[][]: array for synthetic samples (* Compute k nearest neighbors for each minority class sample only. *)
for i ← 1 to T do
    Compute k nearest neighbors for i, and save the indices in the nnarray
    Populate(N, i, nnarray)
end for
Populate(N, i, nnarray) (* Function to generate the synthetic samples. *)
while N ≠ 0 do
    Choose a random number between 1 and k, call it nn. This step chooses one of the k nearest neighbors of i.
    for attr ← 1 to numattrs do
        Compute: dif = Sample[nnarray[nn]][attr] - Sample[i][attr]
        Compute: gap = random number between 0 and 1
        Synthetic[newindex][attr] = Sample[i][attr] + gap*dif
    end for
    newindex++
    N–
end while
return (*End of populate*)
```

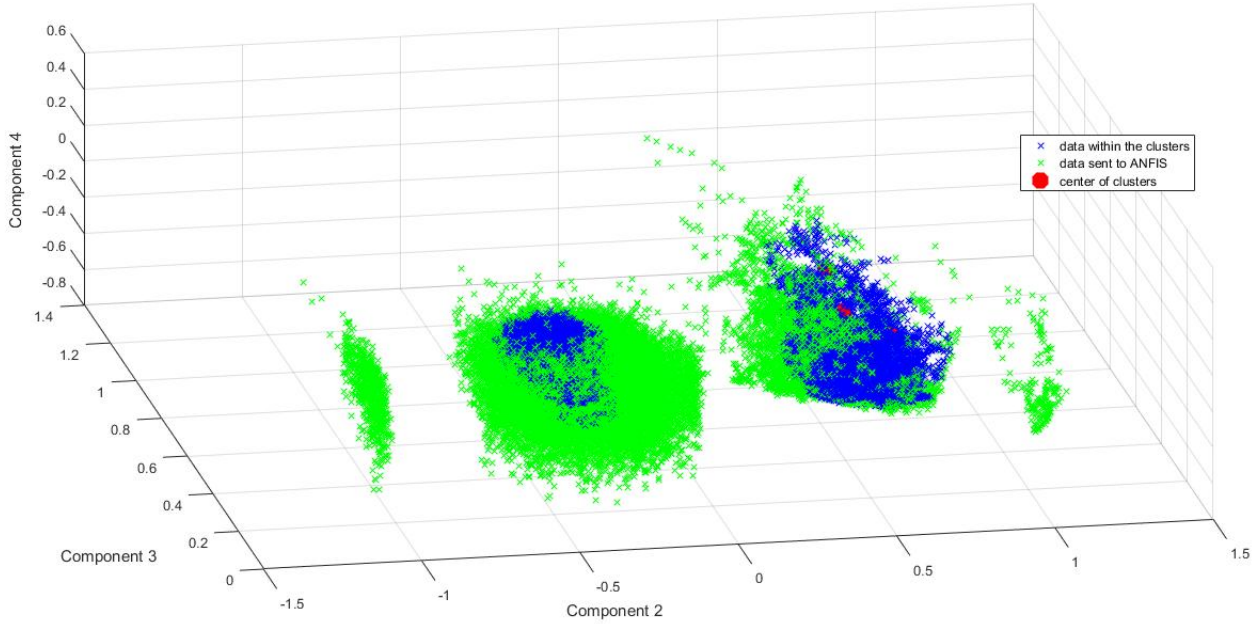


(a) Clustering: components 2,3,4

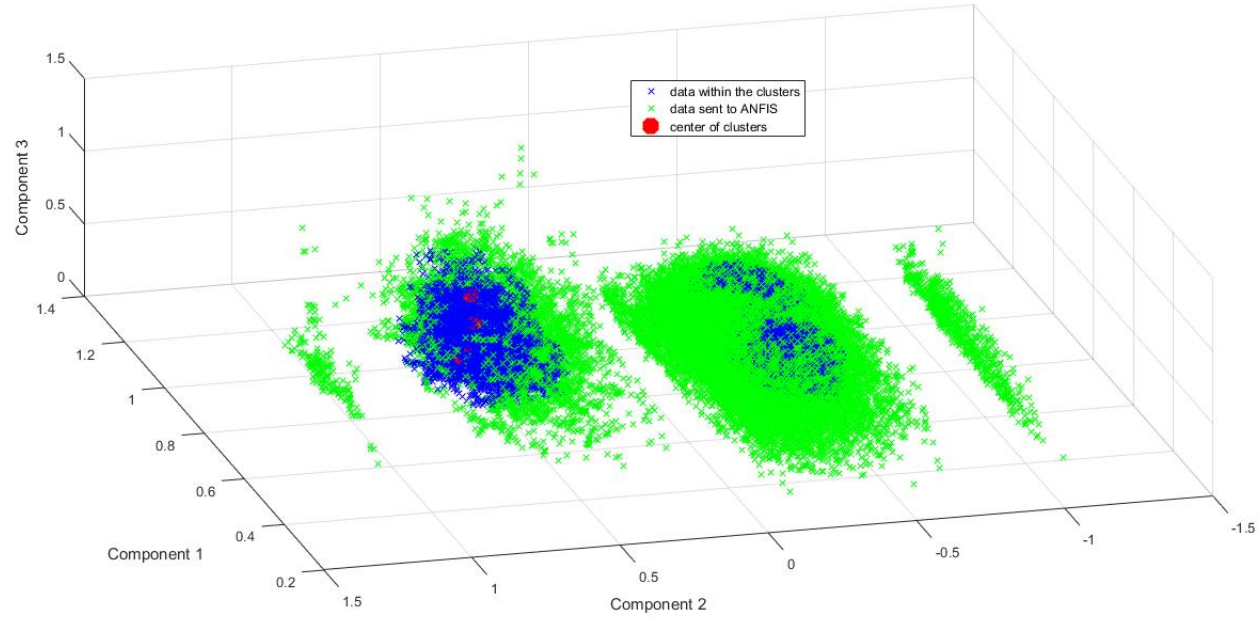


(b) Clustering: components 1,3,4

Figure 16. Clustering on principal components



(a) Clustering on sparse principal component 2,3,4



(b) Clustering on sparse principal component 1,3,4

Figure 17. Clustering on sparse principal components