# Rule Extraction Using a Novel Gradient-Based Method and Data Dimensionality Reduction

*Xiuju Fu* and *Lipo Wang*[*]
School of Electrical and Electronic Engineering
Nanyang Technological University
Block S2, Nanyang Avenue, Singapore 639798
Email: {p146793114,elpwang}@ntu.edu.sg
http://www.ntu.edu.sg/home/elpwang
[*]Corresponding author

**Abstract - Data dimensionality reduction is one of the preprocessing procedures carried out before inputting patterns to classifiers. In many cases, irrelevant or redundant attributes are included in data sets, which interfere with knowledge discovery from data sets. In this paper, we propose a novel gradient-based rule-extraction method with a separability-correlation measure (SCM) ranking the importance of attributes. According to the attribute ranking results, the attribute subsets which lead to the best classification results are selected and used as inputs to a classifier, such as an RBF neural network in our paper. The complexity of the classifier can thus be reduced and its classification performance improved. Our method uses the classification results with reduced attribute sets to extract rules. Computer simulations show that our method leads to smaller rule sets with higher accuracies compared with other methods.**

## I. INTRODUCTION

As data available increase in terms of the number of patterns (samples) and the number of attributes (features), there is an increasing need for data dimensionality reduction (DDR). DDR aims at reducing irrelevant or redundant attributes while maintaining concepts of data. DDR has become an important aspect of data mining, since human experts and corporate managers are able to make better use of lower-dimensional data compared to higher-dimensional ones. In addition, with a fewer number of attributes obtained by DDR techniques, concise rules with higher accuracies can be obtained in rule extraction tasks.

In this paper, we propose a novel gradient-based rule-extraction method with DDR, i.e., important features are selected first based on a separability-correlation measure (SCM) for determining the importance of the original attributes. Once attribute importance ranking is obtained using the SCM, a classifier is used to select a subset of the attributes that leads to the lowest classification error.

Usually, a rule consists of an IF part and a THEN part. The premise parts of rules are composed of combinations of attributes. There are three kinds of rule decision boundaries, i.e., hyper-rectangular, hyper-plane, and hyper-ellipse. Due to its explicit form and perceptibility, hyper-rectangular decision boundary is often employed in rule extraction, such as rules extracted from the MLPs [2][6] and from RBF neural networks [7][8]. In order to obtain symbolic rules with hyper-rectangular decision boundaries, a special *interpretable* MLP (IMLP) was constructed in [2]. In an IMLP network, each hidden neuron receives a connection from only one input unit, and the activation function used for the first hidden layer neurons is the threshold function. In [6], the range of each input attribute was divided into intervals. The attribute was then encoded as a binary string accordingly. Rules with hyper-rectangular decision boundaries were thus obtained. Ishibuchi [5] extracted fuzzy IF-THEN rules. To determine the threshold function, sub-intervals, and membership functions, prior knowledge on how to divide the ranges of the attributes is desirable. Unsuitable division of attribute ranges leads to low rule accuracy. The division will then have to be adjusted. The training procedure and the rule extraction procedure will have to be repeated.

In this paper, we use the RBF neural network as a classifier. The rule-extraction method extracts rules from the simplified RBF classifier whose inputs are selected features. In an RBF classifier, the boundary of the receptive field of the kernel function is a hyper-sphere. The

Euclidean distance between a pattern and the center of the cluster measures the probability that a pattern belongs to a class. Rules with hyper-rectangular decision boundaries are extracted based on the training result of an RBF neural network using gradient descent theory.

The paper is organized as follows. The SCM measure for ranking the importance of attributes is proposed in Section II. Section III introduces how to construct the modified RBF neural network classifier efficiently. Section IV presents our novel gradient-based rule-extraction method. Experimental results on reducing data dimensionality and obtaining a simpler architecture of the RBF classifier are shown in Section V. Finally, we conclude the paper in Section VI.

## II. SEPARABILITY-CORRELATION MEASURE FOR FEATURE IMPORTANCE RANKING

### A. A Class Separability Measure

The probability of correct classification is large, when the distances between different classes are large. Therefore, the subset of features which can maximize the separability between classes is a desirable objective of feature selection.

Class Separability may be measured by the intraclass distance (the distance of patterns within class) $S_w$ and the interclass distance (the distance between patterns of different classes) $S_b$ [3]:

$$S_w = \sum_{i=1}^{C} \frac{P_i}{n_i} \sum_{k=1}^{n_i} [(\overline{\vec{X}}_{ik} - \vec{m}_i)(\overline{\vec{X}}_{ik} - \vec{m}_i)^T]^{\frac{1}{2}} \quad , \tag{1}$$

and

$$S_b = \sum_{i=1}^{C} P_i [(\vec{m}_i - \vec{m})(\vec{m}_i - \vec{m})^T]^{\frac{1}{2}} \quad . \tag{2}$$

Here $C$ is the number of classes in the data set. $n_i$ is the number of patterns in the $i$-th class. $P_i$ is the probability of the $i$-th class. $\overline{\vec{X}}_{ik}$ is the normalized data vector, whose $j$-th attribute, $X_{ik}(j)$ is normalized as:

$$\overline{X}_{ik}(j) = \frac{X_{ik}(j)}{\mathrm{Max}(x_j) - \mathrm{Min}(x_j)} \quad , \tag{3}$$

where $\mathrm{Max}(x_j)$ and $\mathrm{Min}(x_j)$ are the maximum and minimum of the $j$-th attribute in the data set respectively. $j = 1, 2, ..., n$. $n$ is the number of attributes. $X_{ik}(j)$ is the original (unnormalized) data. $\vec{m}_i$ is the mean vector of the $i$-th class:

$$\vec{m}_i = \frac{\sum_{k=1}^{n_i} \overline{\vec{X}}_{ik}}{n_i} \quad . \tag{4}$$

$\vec{m}$ is the mean of all patterns in the data set:

$$\vec{m} = \frac{\sum_{i=1}^{c} \sum_{k=1}^{n_i} \overline{\vec{X}}_{ik}}{n} \quad . \tag{5}$$

$N$ is the total number of patterns in the data set, i.e., $N = n_1 + n_2 + ... + n_c$.

If removing attribute $k_1$ from the data set leads to less class separability, i.e., a greater $S_w/S_b$, compared to the case where attribute $k_2$ is removed, one may consider attribute $k_1$ more important for classification of the data set than attribute $k_2$ is, and vice versa. Hence we may rank the importance of the attributes by calculating the intraclass-to-interclass distance ratio with each attribute omitted in turn.

However, the ratio $S_w/S_b$ does not always work well as a class separability measure. For example, consider 2 classes, with one class surrounding the other, but are completely *separable.* Since $\vec{m}_1$, $\vec{m}_2$ and $\vec{m}$ defined in eq. 4 and eq. 5 are equal, $S_b \to 0$, which indicate total *inseparability.* Here there is a need to have other importance measures.

### B. An Attribute-Class Correlation Measure

In addition to the separability of classes in the data set, the correlation between the changes in attributes and their corresponding changes in class labels should be taken into account when ranking the importance of attributes. The correlation measure can be a useful factor by combining together with our class separability measure.

We propose the following correlation between the $k$-th attribute and the class labels in the data set:

$$C_k = \sum_{i \neq j} |\overline{\vec{X}}_{ik} - \overline{\vec{X}}_{jk}| \cdot \mathrm{magn}(y_i - y_j) \quad , \tag{6}$$

where $\overline{\vec{X}}_{ik}$ and $\overline{\vec{X}}_{jk}$ are the $k$-th attributes of the $i$-th pattern and the $j$-th pattern, respectively. $y_i$ and $y_j$ are the class labels of the $i$-th pattern and the $j$-th pattern respectively. For any $y$, $\mathrm{magn}(y) = 1$ if $|y| > 0$ and $\mathrm{magn}(y) = -0.05$ if $|y| = 0$. A great magnitude of $C_k$ shows that there is a close correlation between class labels and the $k$-th attribute, which indicates the great importance of attribute $k$ in classifying the patterns, and vice versa.

### C. The Separability-Correlation Measure for Attribute Importance Ranking

We propose the following separability-correlation measure (SCM) to evaluate the importance levels of attributes by combining the above two measures:

$$R_k = \chi \overline{S_k} + (1-\chi)\overline{C_k} \quad , \tag{7}$$

where $S_k = \frac{S_{wk}}{S_{bk}}$, $\overline{S_k} = \frac{S_k - \mathrm{Min}(S_k)}{\mathrm{Max}(S_k) - \mathrm{Min}(S_k)}$ is the normalization of $S_k$. $\mathrm{Max}(S_k)$ and $\mathrm{Min}(S_k)$ are the maximum and minimum of all $S_k$, respectively. $k = 1, 2, ..., n$. $n$ is the number of attributes. $S_{wk}$ and $S_{bk}$ are intraclass and interclass distances calculated with the $k$-th attribute omitted from each pattern, respectively. For example, the $i$-th pattern $\vec{X_i} = \{x_{i1}, x_{i2}, ...x_{ik}, x_{ik+1}, ...x_{in}\}$ becomes $\vec{X_i'}$=$\{$ $x_{i1}$, $x_{i2}$, ..., $x_{ik-1}$, $x_{ik+1}$, ..., $x_{in}\}$ when $R_k$ is calculated. $\overline{C_k} = \frac{C_k - \mathrm{Min}(C_k)}{\mathrm{Max}(C_k) - \mathrm{Min}(C_k)}$ is the normalization of $C_k$. $\chi$ is the parameter to weight the two items for the final measure. Here $0 \geq \chi \geq 1$ and $\chi$ is determined empirically: the best choice of $\chi$ should lead to a subset of attributes which results in the highest classification accuracy.

The importance levels of attributes are ranked using the values of $R_k$. The greater the magnitude of $R_k$, the more important the $k$-th attribute. We will demonstrate the use of our SCM method in Section IV.

We use a combination of two measures, i.e., class separability and attribute class correlation, because either of them alone does not work well, as shown in our experimental results presented later in the paper.

Due to the computational burden of optimal search methods, one has to resort to suboptimal feature selection methods. In classification tasks, since the goal is to obtain better classification accuracy with less complicated construction of classifiers, the strategy of using the classification accuracy as evaluation for selecting features is used widely. We use suboptimal search and RBF classifiers as evaluators in this paper.

## III. CONSTRUCTING AN EFFICIENT RBF CLASSIFIER

There are three layers in the RBF neural network, i.e., the input layer, the hidden layer with Gaussian activation functions, and the output layer. In this paper, we use the RBF network for classification. If there are M classes in the data set, we write the m-th output of the network as follows:

$$y_m(\mathbf{X}) = \sum_{j=1}^{K} w_{mj} \o_j(\mathbf{X}) + w_{m0} b_m \quad . \tag{8}$$

Here $\mathbf{X}$ is the $n$-dimensional input pattern vector, $m = 1, 2, ..., M$, $K$ is the number of hidden units. $M$ is the number of output. $w_{mj}$ is the weight connecting the $j$-th hidden unit to the $m$-th output node. $b_m$ is the bias. $w_{m0}$ is the weight connecting the bias and the $m$-th output node. $\o_j(\mathbf{X})$ is the activation function of the $j$-th hidden unit:

$$\o_j(\mathbf{X}) = \mathrm{e}^{\frac{||\mathbf{X}-\mathbf{C_j}||^2}{2\sigma_\mathbf{j}{}^2}} \quad , \tag{9}$$

where $\mathbf{C_j}$ and $\sigma_j$ are the center and the width for the j-th hidden unit, respectively, which are adjusted during learning. The weights connecting the hidden layer and the output layer can be determined by a linear least square (LLS) method [1], which is fast and free of local minima, in contrast to the multilayer perceptron neural network.

Based on the attribute importance ranking, we further propose to reduce the structural complexity and to improve the performance of the RBF network as follows. According to the rank of importance level obtained by the algorithm described in Section II, $J$ most important attributes are used for classification with the RBF neural network for $J = 1, 2, ..., N-1, N$. The classification error rate is calculated for each $J$. Thus, $N$ classification error rates are calculated corresponding to $N$ subsets of attributes. For small $J$, classification error rate decreases as $J$ increases until all relevant attributes are included. As $J$ increases further, the classification error rate may remain unchanged or even increase because redundant or irrelevant attributes are included. The best subset of attributes is the one with the smallest classification error rate.

## IV. A NOVEL RULE-EXTRACTION METHOD

The rule extraction algorithm proposed here is based on the widths and the centers of the Gaussian kernel functions, and the weights connecting the hidden neurons to the output layer. Each hidden neuron of the RBF neural network is responsive to a subset of input patterns (instances).

The objective of tuning the rule premises is to determine the boundaries of rules so that a high rule accuracy is obtained for the testing data set. Before starting the tuning process, all of the premises of the rules must be initialized. Let us assume that the number of attributes is $n$. The number of rules equals to the number of hidden neurons in the trained RBF network. The number of the premises of rules equals to $n$. The upper limit $U_{ji}$ and the lower limit $L_{ji}$ of the $j$th premise in the $i$th rule are initialized according to the trained RBF classifier as:

$$U_{ji}^{(0)} = \mu_{ji} + \sigma_i \quad , \tag{10}$$

$$L_{ji}^{(0)} = \mu_{ji} - \sigma_i \quad , \tag{11}$$

where $\mu_{ji}$ is the jth item of the center of the ith kernel function. $\sigma_i$ is the width of the $i$th kernel function.

We introduce the following notations. Suppose $\eta^{(t)}$ is the tuning rate at time $t$. Initially $\eta^{(0)} = 1/N_I$, where $N_I$ is the number of iteration steps for adjusting a premise. $N_I$ is set to be 20 in our experiments. $E$ is the rule error rate.

$$Q_{ji}^{(t)} \equiv \frac{\partial E}{\partial U_{ji}}|_t \quad , \tag{12}$$

$$A_{ji}^{(t)} \equiv \frac{\partial E}{\partial L_{ji}} \;|_t \quad . \tag{13}$$

$U_{ji}^{(t)}$ and $L_{ji}^{(t)}$, the upper and lower limits at time $t$, are tuned as follows.

$$U_{ji}^{(t+1)} = U_{ji}^{(t)} + \Delta U_{ji}^{(t)} \quad , \tag{14}$$

$$L_{ji}^{(t+1)} = L_{ji}^{(t)} + \Delta L_{ji}^{(t)} \quad . \tag{15}$$

Initially, we let

$$\Delta U_{ji}^{(0)} = \eta^{(0)} \quad . \tag{16}$$

$$\Delta L_{ji}^{(0)} = -\eta^{(0)} \quad . \tag{17}$$

Subsequent $\Delta U_{ji}^{(t)}$ and $\Delta L_{ji}^{(t)}$ are calculated as follows.

$$\Delta W_{ji}^{(t)} = \begin{cases} \eta^{(t)} & , \text{ if } Q_{ji}^{(t-1)} < 0 \\ -\eta^{(t)} & , \text{ if } Q_{ji}^{(t-1)} > 0 \\ \Delta W_{ji}^{(t-1)} & , \text{ if } Q_{ji}^{(t-1)} = 0 \\ -\Delta W_{ji}^{(t-1)} & , \text{ if } Q_{ji}^{(t-1)} = 0 \text{ for} \\ & , \ \tfrac{1}{3}N_I \text{ consecutive} \\ & , \text{ iterations,} \end{cases} \tag{18}$$

where $W = U, L$. When $Q_{ji}^{(t)} = 0$ consecutively for $\frac{1}{3}N_I$ time steps, this means that the current direction of premise adjustment is fruitless. $\Delta W_{ji}^{(t)}$ changes its sign as shown in the 4th line of eq. 18. In this situation, we also let $\eta^{(t)} = 1.1\eta^{(t-1)}$, which helps to keep the progress from being trapped. Otherwise $\eta^{(t)}$ remains unchanged.

Two rule tuning stages are used in our method. In the first tuning stage, the premises of $m$ rules ($m$ is the number of hidden neurons of the trained RBF network) are adjusted using gradient descent theory for minimizing the rule error rate. Since overlaps exist between clusters of the same class, some hidden neurons may be overlapped completely when a hyper-rectangular rule is formed using gradient descent method. Thus, the rules overlapped completely are redundant for representing data and should be removed from the rule set at the second tuning stage.

## V. EXPERIMENTAL RESULTS

Iris, Monk3, Breast Cancer data sets [9] are used in this paper to test our algorithms for ranking attribute importance and constructing a simplified RBF network. Each data set is divided into 3 parts, i.e., training, validation, and test sets. Each experiment is repeated 5 times with different initial conditions and the average results are recorded.

Attribute importance rankings using the SCM with different $\chi$'s (eq.7) are shown in Table I, which shows that $\chi$ affects the order of attribute importance ranking. 5 $\chi$'s are used, i.e., $\chi = 0.0, 0.4, 0.5, 0.7, 1.0$. In order to determine which order is better, different subset of attributes are input to the RBF classifier for each order, so as to find the best subset for that order. We select the subset of attributes corresponding to the lowest classification error rate for each data set and each ranking order. According to the experimental results, when $\chi = 0.4$, the importance ranking results for the three data sets lead to the lowest or nearly the lowest validation error rates with the smallest attribute subsets.

### A. Iris Data Set

There are 4 attributes in Iris data set. patterns of Iris data set are divided into 3 sets, i.e., 90 patterns for training, 30 for validation, and 30 for testing. $\chi = 0.4$ is selected for that it leads to the smallest attribute subset $\{3, 4\}$ with the nearly lowest classification error rate (Ta-

TABLE I

Attribute importance ranking using the SCM with different $\chi$.

| $\chi$ | Iris | Monk3 | Breast |
|---|---|---|---|
| 0.0 | 4,3,1,2 | 5,4,2,1,6,3 | 7,2,4,3,8,9,5,6,1 |
| 0.4 | 4,3,1,2 | 5,2,4,1,6,3 | 2,7,3,4,9,5,8,6,1 |
| 0.5 | 4,1,3,2 | 5,2,4,1,6,3 | 2,7,3,4,9,5,1,8,6 |
| 0.7 | 1,4,2,3 | 5,2,4,1,6,3 | 2,7,1,3,4,9,5,8,6 |
| 1.0 | 1,2,4,3 | 5,2,3,6,4,1 | 1,2,7,3,4,9,5,8,6 |

ble II. We obtain 2 rules, 2 antecedents per rule for Iris data set. The accuracy is 100% for testing data set. We compare our rule extraction results for Iris with other methods in Table III.

TABLE II

Classification error rates for Iris data set with different attribute subsets when $\chi = 0.4$.

| Attributes used | Error Rate | | |
|---|---|---|---|
| | Training | Validation | Test |
| 4 | 0.1222 | 0.0667 | 0.1333 |
| **4,3** | **0.0333** | **0.0000** | **0.0333** |
| 4,3,1 | 0.0556 | 0.0333 | 0.1000 |
| 4,3,1,2 | 0.0889 | 0.1000 | 0.1000 |

### B. Monk3 Data Set

There are 6 attributes in Monk3 data set. Monk3 data has a training set with 122 patterns and a test set with 421 patterns. We divide the test set into 200 patterns for validation and 221 patterns for testing. $\chi = 0.4$ is selected for that it leads to the smallest attribute subset $\{2, 4, 5\}$ with the lowest classification error rates, which is shown in Table IV. We obtain 3 rules, 3 antecedents per rule for Monks data set. The rule accuracy is 98% for testing data set. Setiono [10] extracted 2 rules, 5.83 antecedents per rule, and 100% rule accuracy for Monk3 data set based on the pruned MLP. We obtain 3 rules with 3 antecedents per rule.

TABLE III

A comparison of rules for Iris data set obtained with different methods.

| Methodology | accuracy | boundary |
|---|---|---|
| Modified RX algorithm (MLP)[4] | 97.33% | hyper-plane |
| IMLP[2] | 97.33% | hyper-rectangular |
| RBF [8] | 80% | hyper-rectangular |
| RBF [7] | 100% | hyper-rectangular |
| Our algorithm | 100% | hyper-rectangular |

TABLE IV

Classification error rates for Monk3 data set with different attribute subsets when $\chi = 0.4$.

| Attributes used | Error Rate | | |
|---|---|---|---|
| | Training | Validation | Test |
| 5 | 0.1880 | 0.3000 | 0.2870 |
| 5,2 | 0.1780 | 0.2830 | 0.2690 |
| **5,2,4** | **0.0242** | **0.0585** | **0.067** |
| 5,2,4,1 | 0.0899 | 0.3360 | 0.1830 |
| 5,2,4,1,6 | 0.0498 | 0.1897 | 0.1320 |
| 5,2,4,1,6,3 | 0.0328 | 0.2030 | 0.1240 |

### C. Breast Cancer Data Set

There are and 9 attributes in Breast cancer data set. There are 699 patterns in Breast cancer data set. 16 patterns with losing attribute are removed. Of the 683 patterns left, 444 were benign, and the rest were malign. In 683 patterns, 274 patterns for training, 204 for validation, 205 for testing. $\chi = 0.4$ is selected for that it leads to the smallest attribute subset $\{2, 3, 7\}$ with the lowest classification error rates (Table V). We obtain 3 rules for class 2 (malignant), and a default rule for class 1 (benign). On average, 2 antecedents per rule for Breast cancer data set. The rule accuracy is 96.6% for testing data set. Setiono [10] extracted 2.9 rules and obtained 94.04% accuracy for Breast cancer data set based on the pruned MLP. The rules for Breast cancer data set are below:

Rule 1: if Uniformity of Cell Shape is within [2, 10], and Bland Chromatin is within [4, 10], then this case is Malignant.

Rule 2: if Uniformity of Cell Shape is within [5, 10], and Bland Chromatin is within [2, 10], then this case is Malignant.

Rule 3: if Uniformity of Cell Size is within [3, 10], and Uniformity of Cell Shape is within [3, 10], then this case

TABLE V

Classification error rates for Breast cancer data set with different attribute subsets when $\chi = 0.4$.

| Attributes used | Error Rate | | |
|---|---|---|---|
| | Training | Validation | Test |
| 2 | 0.1100 | 0.0803 | 0.1022 |
| 2,7 | 0.0709 | 0.0657 | 0.0876 |
| **2,7,3** | **0.0269** | **0.0365** | **0.0073** |
| 2,7,3,4 | 0.0391 | 0.0438 | 0.0365 |
| 2,7,3,4,9 | 0.0269 | 0.0365 | 0.0219 |
| 2,7,3,4,9,5 | 0.0342 | 0.0365 | 0.0146 |
| 2,7,3,4,9,5,8 | 0.0293 | 0.0438 | 0.0073 |
| 2,7,3,4,9,5,8,6 | 0.0269 | 0.0438 | 0.0146 |
| 2,7,3,4,9,5,8,6,1 | 0.0342 | 0.0365 | 0.0146 |

is Malignant.

Default rule: this case is benign.

## VI. CONCLUSIONS

In this paper, rule extraction is carried out to express data sets. A SCM is used to rank the importance of attributes first. According to the ranking results, different attribute subsets are used as inputs to RBF classifiers. The attribute subsets with the lowest classification error rates and the least numbers of attributes are selected. Rules are extracted based on a novel gradient-based method and feature subsets selected. Compared to other methods, more concise and accurate rules are extracted for Iris and Breast cancer data sets, while for Monk3 data set, the rule accuracy is lower. But the antecedents per rule for Monk3 data set is smaller than other methods. In addition, rules extracted by our algorithm have hyper-rectangular decision boundaries, which is desirable due to its explicit perceptibility. Our approach eliminates the need for an error-prone transformation from continuous attributes into discrete ones as required in MLP-based methods.

## References

[1] C. M. Bishop, *Neural network for pattern recognition*, Oxford University Press Inc., New York, 1995.

[2] G. Bologna and C. Pellegrini, "Constraining the MLP power of expression to facilitate symbolic rule extraction", *IEEE World Congress on Computational Intelligence*, vol.1, pp. 146-151, 1998.

[3] P. A. Devijver and J. Kittler, *Pattern recognition: a statistical approach*, Prentice-Hall International, Inc. London, 1982.

[4] E. R. Hruschka and N. F. F. Ebecken, "Rule extraction from neural networks: modified RX algorithm", *Proc. International Joint Conference on Neural Networks*, Vol. 4, pp. 2504-2508, 1999.

[5] H. Ishibuchi and M. Nii, "Generating fuzzy if-then rules from trained neural networks: linguistic analysis of neural networks", *IEEE International Conference on Neural Networks*, vol. 2, pp. 1133-1138, 1996.

[6] H. J. Lu, R. Setiono, and H. Liu, "Effective data mining using neural networks", *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no. 6, Dec. 1996.

[7] K. J. McGarry, S. Wermter, and J. MacIntyre, "Knowledge extraction from radial basis function networks and multilayer perceptrons", *Proc. International Joint Conference on Neural Networks*, vol. 4, pp. 2494-2497, 1999.

[8] K. J. McGarry and J. MacIntyre, "Knowledge extraction and insertion from radial basis function networks", *IEE Colloquium on Applied Statistical Pattern Recognition (Ref. no. 1999/063)*, pp. 15/1-15/6, 1999.

[9] P. M. Murphy, and D. W. Aha, (1994). *UCI Repository of machine learning databases*, [www.ics.uci.edu/ mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science.

[10] R. Setiono, "Extracting M-of-N rules from trained neural networks", *IEEE Transactions on Neural Networks*, vol. 11, no. 2, pp. 512 -519, March, 2000.