Advanced Engineering Informatics 23 (2009) 348–357

Contents lists available at ScienceDirect

Advanced Engineering Informatics

journal homepage: www.elsevier.com/locate/aei

# A neural network classifier with rough set-based feature selection to classify multiclass IC package products

Y.H. Hung *

*Department of Industrial Engineering and Management, National Chin-Yi University of Technology, 35, Lane 215, Section 1, Chung-Shan Road, Taiping, Taichung, 411 Taiwan, ROC*

ARTICLE INFO

*Article history:*
Received 3 June 2008
Received in revised form 24 April 2009
Accepted 26 April 2009
Available online 29 May 2009

ABSTRACT

The choice of packaging type is important to the process of researching and developing an integrated circuit (IC). Indeed, for an IC chip designer, the importance can be compared to an architect's choice of construction design. Since there are considerable variations in characteristics and in the types of products available, collecting information about packaging technologies and products can be difficult and time-consuming. Therefore, finding the means to provide packaging information to designers quickly and efficiently is necessary and important, as this will not only help designers accurately decide on design methods for an IC, but also significantly reduce processing risks. In this study, existing product information, such as the dimensions, characteristics and design and application criteria, of a product was analyzed. One of the biggest issues when data from multi-dimensional measurements are represented as a feature vector is that the feature space of the raw data often has very large dimensions. This study explores the use of rough set attribute reduction (RSAR) to reduce attributes of the IC package family dataset, and artificial neural networks, to construct an efficient IC package type classifier model. The experimental results show that the features produced by RSAR improve on generalization accuracy: the training and testing set classification accuracy rates were 96.9% and 98.2%, respectively.

## 1. Introduction

The rate of change in methods of communication and entertainment has increased in recent years, and in order to deal with this advancement in technology, data mining (DM) technology has been utilized to play a major role in solving classification problems for the semiconductor industry. Several machine learning algorithms have been applied to DM, and neural networks (NNs) have been discovered to be one of the most effective techniques for classification and regression. The advantage of NNs is that they offer a powerful yet general framework for representing nonlinear mappings from several input variables to several output variables, where the form of the mapping is governed by a number of adjustable parameters [7]. However, NNs involve long training times and are therefore more suitable for applications where long training times are acceptable. In other words, neural networks are able to process a large amount of data, but entail the drawback of longer training time and slower convergence speed. To overcome this disadvantage, we focus on feature selection and how it plays an important role in the classification problem. Feature selection is frequently used as a preprocessing step to machine learning, and has been a fertile field of research and development since the 1970s. A feature selection process can be used to remove terms in the training dataset that are statistically uncorrelated with the class labels, thus improving both efficiency and accuracy [2,11,15,19]. A discernibility-based method, known as rough set theory (RST) [32], has been introduced to deal effectively with features that are redundant or worse. RST can be used for classification, to discover structural relationships within imprecise or noisy data. Most existing rough set-based feature selection approaches, such as rough set attribute reduction (RSAR) [37], rely on the information gathered from the lower approximation of a set, to minimize data. However, unlike other dimensionality reduction methods, RSAR is able to preserve the original meaning of the features after reduction. Furthermore, unlike statistical correlation–reduction approaches, RSAR requires no human input or domain knowledge other than the given datasets. The use of RSAR has already been widely researched and applied in areas such as machine learning, knowledge acquisition, decision analysis, knowledge discovery, and pattern recognition [30]. But RSAR has not yet been applied to the field of IC packaging operation management. Consequently, the benefits of using RSAR in this area have been to date untested.

In recent years, cellular phones, PDAs and other portable devices have grown rapidly in popularity, mostly due to the compact, thin, and light features that have become key requirements for design. As packaging sizes continue to decrease, the level of integration of semiconductor devices continues to increase in

* Corresponding author. Tel.: +886 4 23924505x7628; fax: +886 4 23934620.
*E-mail address:* hys502@ncut.edu.tw

1474-0346/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.aei.2009.04.001

complexity and number of components. Today, IC assemblers have also become more suited to developing the new packaging. Crowley [9] has pointed out that these IC Design Houses depend on innovative packaging developments to offer competitive packaging solutions for demanding applications. IC Design Houses now need to think about the final design even in the planning stages of chip design, to determine the feasibility of the final form. This highlights the importance of selecting the best packaging method during the IC design stage [35].

Until recently, no research has been dedicated to IC packaging classification in the packaging industry. In this study, we constructed an efficient classification system through the analysis of existing product information, such as dimensions, characteristics, design of application criteria. However, the volume of this data was extremely large, and impracticable to contend with, without using adequate computational tools such as RSAR. In addition, RSAR and NNs can be linked together to provide better results and faster performance than are achievable using the classical neural network approach alone. In this study, we employed a hybrid system of RSAR and neural networks to improve classification accuracy for large IC package type databases. Our study therefore, makes the following contributions:

(1) We demonstrate that RSAR is an effective preprocessing technique for reducing the number of features before using data to train a neural network in selecting the best IC package type problem. We then compare the performance of the neural networks, with and without rough set preprocessing, and demonstrate that through the removal of noisy or misleading features, we can increase classifier accuracy by using feature selection.
(2) Since the design process must be focused not only on the IC level but also on the substrate, and subsequent PCB levels, we found that the IC package type classification model can be used to assist an IC designer in understanding the packaging design process and its key considerations, as well as focusing on their packaging options and shortening the overall IC design process.

## 2. Literature review

### 2.1. Data mining applications for classification problems

Applying the data mining approach to industrial management problems is likely to be a real challenge for industry in the years to come. IC package product classification, an important problem in machine learning, may be viewed as a supervised learning process. Some commonly used DM techniques are: statistical methods, decision trees (DTs), artificial neural networks (ANNs), rough sets (RS), Bayesian classifiers (BC), case-based reasoning (CBR) and support vector machines (SVM). In recent years, ANNs have been widely applied and have proven to be effective in performing complex functions in manufacturing management fields. Chen et al. [5] have proposed a hybrid ANN-based approach to pattern recognition for control charts. Acciani et al. [1] have pointed out that ANNs can be structured to perform classification in semiconductor manufacturing problems, to approximate equations [17], to classify mean shifts from multivariate $\chi^2$ chart signals [3], to cluster analysis in industrial market segmentation [21], to model the test performance for small signal modeling RF/microwave active devices [23], and to predict values [4,38]. There are many different types of NNs and neural network algorithms, with the most popular being backpropagation neural networks (BPNN). BPNNs use gradient descent to iteratively learn a set of weights for predicting the class label of tuples, converging to a local minimum within training error with respect to network weights [12]. However, Li and Wong [25] point out that BPNNs have two obvious shortcomings: firstly, they require a long time to be trained within a large database, and secondly, they lack explanation facilities for their knowledge. Building a BPNN model is complicated by the presence of many training factors, which may include hidden neurons, training tolerance, initial weight distribution and function gradient [13]. The time required to train a satisfactory BPNN model increases dramatically with the number of features, and a large number of features also degrades the accuracy of a prediction [40]. Thangavel et al. [39] applied the feature selection algorithm to construct an efficient BPNN classifier model; their research showed that the BPNN approach creates a model based on a training dataset. During the training process, the weights and biases of the network are iteratively adjusted to minimize the network performance function. The statistical performance function used for feed-forward networks is known as the mean square error (MSE) [38].

### 2.2. Feature selection methods

Feature selection and feature extraction are used to improve the efficiency of learning algorithms by dimensionality reduction or by finding an optimal subset of features. Feature extraction is a dimension reduction technique in which a transformation is applied to the vector of all input data followed by the selection of the best subset of transformed features. Feature extraction algorithms include principal component analysis (PCA) based algorithms, linear discriminant analysis (LDA) and nonlinear transformation [10]. PCA algorithms are aimed at finding a subspace whose basis vectors correspond to the maximum-variance directions inside the original space. Feature selection algorithms, on the other hand, are a process used for finding the optimal subset of features that satisfy a certain criterion. Cios et al. [8] and Kittler [18] pointed out in their research that one of the fundamental steps in classifier design is the reduction of pattern dimensions by feature selection. Feature selection should diminish the cardinality of the feature subset and ensure that classification accuracy does not decrease significantly [24,26]. In other words, feature selection is often isolated as a separate step when processing pattern sets. Its aim is to remove unexpected noise data from the original attributes. Its function is like Kwak's [22] mutual information based on Parzen Window, Miyamoto's [27] fisher criterion and Kononenko's [20] relief. Noisy and irrelevant features usually contain little discriminant information. Dash and Liu [11] provides a detailed survey and overview of the existing methods for feature selection.

Thangavel et al. [39] have discussed feature selection methods in terms of three main approaches: (1) the filter approach; (2) the wrapper approach; and (3) the embedded approach. The filter approach is independent of the selection features used by the induction algorithm as it relies only on the characteristics of the features themselves. The filter approach is known to be the fastest of the three, but has some blind areas and performance inductions which have not been considered. The wrapper approach, on the other hand, assesses the subsets of variables according to their relevance to a given predictor. The feature selection is then “wrapped around” an induction algorithm so that the bias of the operators that defined the search and that of the induction algorithm, can interact mutually. This method conducts the search for a good subset by using the embedded approach, to perform variable selection as part of its learning procedure, usually specific to a given learning mechanism. Zhong et al. [41] propose that the optimal feature subsets be obtained by using the wrapper approach; however, this lacks ease and convenience of use because of its spatial and temporal complexity.

As a filter model, the RSAR approach is one of the best effective attribute reduction methods that can preserve the meaning of the attributes [31]. It provides a filter-based tool by which knowledge may be extracted from a domain in a concise way; retaining the information content while reducing the amount of knowledge involved. Moreover, the main limitation of RSAR in the literature is the restrictive requirement that all data be discrete. RSAR is also a formal methodology that can be employed to reduce the dimensionality of datasets as a preprocessing step in training a learning system from the data. Pawlak [33] pointed out that RSAR not only deals with the classification analysis of data tables to extract decision rules, but can also be used as a tool to mine data dependencies and reduce the number of attributes included in a data search.

RSAR selects the most information-rich features in a dataset by using simple set operations, not by transforming the data, nor by removing the information needed for the classification task at hand. This means that it is highly efficient and is suitable as a preprocessor for use with much more complex techniques [37]. Li et al. [26] pointed out that RSAR makes the assumption that the data is time-independent and exists in immutable clusters in datasets, but this unfortunately is not the case in real time. Moreover, the reduced attributes are regarded as a significant omission. The advantages of RSAR extend to the runtime of the system; the RSAR learning system becomes more compact and responsive by requiring fewer observations per sample. The cost of obtaining physically measured data decreases, since fewer measurable items need to be maintained, while the overall robustness of the system increases since, with fewer measurable items, the chances of instrumentation malfunction leading to spurious readings are reduced dramatically [6]. Polkowski et al. [36] and Peters et al. [34] also point out that the study of various forms of rough neurons is addressed in a growing number of papers on neural networks that are based on rough sets. The use of RSAR can result in smaller subset sizes and therefore be highly efficient in terms of computational effort. Since it is based on simple set operations, it becomes suitable as a preprocessor for techniques that are more complex [29]. After eliminating the noise input and compressing the remaining feature set based on RSAR as the input dataset for a neural network model, the input vector of a neural network classifier becomes much smaller. Therefore, the optimal feature subset can be used to construct a good BPNN classifier to increase accuracy and to save on computation time.

Neural networks based on RSAR are an important technique to apply to IC package product classification problems. In this research, our objective was to employ RSAR to find a reduced data set (reduct) with a minimal number of attributes, to then reduce the network's input vectors to scale down the size of the whole architecture of the network, and finally, to construct a BPNN neural network for the classification of IC package types. There are usually several subsets of attributes. Those that are minimal are called "reducts". A reduct is a minimal attribute subset of the original data that has the same discernibility power as all of the attributes in the rough set framework. A more detailed definition of "reduct" can be found in [26].

## 3. Data: IC package family dataset

The evolution of IC package technology can be divided into four stages. The first stage consisted of PTH package technologies such as DIP, SIP, ZIP, S-DIP, SK-DIP and PGA. Surface Mount Technology (SMT) then emerged, consisting of QFP (Quad Flat Package), TSOP, FPG, LCC, PLCC and QFN. During the second stage of development, decreasing package volume and increasing I/O were the key aims, but the SMT IC packages were all of the lead-frame type, which is limited by the number of I/O counts. The SMT packages also used gold wire to connect the chip's pad to the carrier, which in turn belonged to the peripheral package type. By the third stage, package technology began to implement the Area Array method, forming package technology such as BGA, Flip Chip and CSP (Chip Scale Package). The use of the Area Array method and the introduction of an organic substrate carrier considerably increased the I/O pin number, meeting the demands of high velocity, high power and super-thin type requirements. The fourth generation packaging types are bare die form factors like Flip Chip, WLCSP, DCA, and the Area Arrayed Flip Chip package type, which are becoming mainstream in new applications of peripheral technologies.

In this study, the IC Package Product Type Database (ICPPTD) was collected from an IC packaging company located in central Taiwan [14]. The IC package product types (ICPPT) were divided into several package body families according to their exterior and function, such as Land Grid Array (LGA) and Flip Chip Ball Grid Array (FCBGA). These families could also be classified according to reliability conditions and different manufacturing processes as well as by their exterior and function. This study first analyzes the information used by ICPPT to classify all kinds of package forms and their characteristic attributes, including the development of products, electrical properties, design of products, development of the manufacturing process, reliability of products and other relevant information used in the IC package industry. The study then summarizes the five IC package product families, which are the TFBGA, LGA, PBGA, FCBGA and QFP Package Families. Each IC family, or package body, has its own applicable IC design scope. In total, there is one category member and thirteen characteristic attributes: Package Size Range, Package Height (mm), Ball Pitch (mm), Lead Count (max.), Wafer Size (in.), Stacked Die Quantity, Substrate Layer, Frequency (GHz) max., MCM, Speed (Gbps) max., Power (W) max., Reliability (Level) and Reliability (IR: °C). The LGA package family for example, was constructed using at most two chip stacks of the SLGA package body; at the same time, the wafer sizes that could be used were 8 in. and 12 in.; the total height of the package bodies were 1.2 mm and 1.4 mm, respectively. The size of the package body ranged from $5 \times 5\ \text{mm}^2$ to $19 \times 19\ \text{mm}^2$, the pitches of the I/O side were 0.5 mm, 0.65 mm, 0.8 mm and 1.0 mm; the highest frequency in electrical property (frequency maximum) was 5 GHz, the fastest transmission speed (transfer speed) was 10 Gbps, the maximum power was 4 W; the reliability condition was Level 3, re-flow temperature was 260 °C, and substrates were either 2 or 4 layers. This study used a total of 2496 data objects, where the number of objects used of the TFBGA Package Family, LGA Package Family, PBGA Package Family, FCBGA Package Family and QFP Package Family were: 632, 424, 352, 816 and 272, respectively. The characteristics of every classification and the data objects are shown in Table 1.

## 4. Experimental design, setup and results

### 4.1. Experimental design

In this study, the entire experiment consisted of three steps (see Fig. 1), data preprocessing, used Excel; the second step was to select of the best attribute (feature selection) based on RSAR and the construction of an IC Package Product Type Classifier using BPNN, also known as ICPPTC.

### 4.2. Experimental setup

The ICPPTD for each connection has 13 attributes plus one class label, and the dataset size of the 2496 objects processed is divided into five product types, comparing the proposed method RSAR_BPNN with the PCA_BPNN method, with FC_BPNN and with

**Table 1**
IC package families and the complete specification scope of their attributes.

| Variable of package family | IC package family | | | | |
|---|---|---|---|---|---|
| | TFBGA<br>"A" | LGA<br>"B" | PBGA<br>"C" | FCBGA<br>"D" | QFP<br>"E" |
| *Attributes* | | | | | |
| Package size (cm$^2$) | 5 × 5–19 × 19 | 5 × 5–19 × 19 | 21 × 21–40 × 40 | 17 × 17–50 × 50 | 32 × 32 |
| Package height (mm) | 0.8–1.6 | 0.5–1.4 | 0.5–3.75 | 0.5–3.75 | 1.0–3.75 |
| Lead pitch (mm) | 0.5–1.0 | 0.5–1.0 | 1.0–1.27 | 1.0–1.27 | 0.5–0.65 |
| Lead count (max.) | 500 | 500 | 1156 | 1156–1521 | 80–256 |
| Wafer size (in.) | 8/12 | 8/12 | 8/12 | 8/12 | 8/12 |
| Stacked die quantity | 1/2/3/4/5/6 | 1/2/4 | 1/2/3 | 1/2/3/4/5/6 | 1/2/3 |
| MCM | Yes/No | Yes/No | Yes/No | Yes/No | Yes/No |
| Substrate layer | 2/4 | 2/4 | 2/4 | 2/4/6/8 | NA |
| Frequency (GHz) (max.) | 2.4–10 | 2.4–10 | 2.4–10 | 2.4–10 | 2.4–10 |
| Speed (Gbps) (max.) | 0.4–10 | 0.4–10 | 0.4–10 | 0.4–10 | 0.4–10 |
| Power (W) (max.) | 2–10 | 2–10 | 4–6 | 4–8 | 2–5 |
| Reliability level | 2–3 | 2–3 | 2–3 | 2–3 | 2–3 |
| IR temp. (°C) | 245/260 | 245/260 | 245/260 | 245/260 | 245/260 |
| No. of attributes | 13 | 13 | 13 | 13 | 12 |
| No. of objects | 632 | 424 | 352 | 816 | 272 |

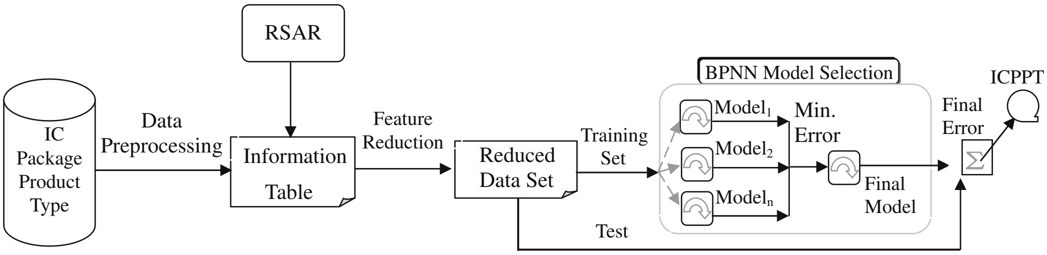

**Fig. 1.** Experiment outline for IC Package Product Type Classifier (ICPPTC).

RSAR. The classification of ICPPTD relative to training and testing of all the experimental methods was found to be 70–30% and 80–20%, respectively. However, before ICPPTD can be reduced, the nominal attributes' data must be preprocessed. This preprocessing is done during the RSAR feature selection stage using ROSETTA, a toolkit that implements rough set methodology. Experiments were designed and then run to reduce the ICPPTD dataset. We then used these reduced datasets (PCA, FC and RSAR) to build a BPNN neural network for ICPPTD classification. The BPNN neural network is a static, feed-forward, neuromorphic system, by which weight values can be determined through supervised learning.

### 4.2.1. Data preprocessing

Data preprocessing involves encoding nominal attributes and setting the dataset into a format that is suitable for the classification task. Before training, it is often useful to encode the condition variables so that they always fall within a specified code. The discrete-valued attributes are then preprocessed with transformation encoding, with one discrete-valued coding scheme applied to each of the condition variables. For example, in Table 1, the classes "5 × 5," "17 × 17," "19 × 19," "21 × 21," "32 × 32," "40 × 40" and "50 × 50" were encoded as "1", "2", "3", "4", "5", "6" and "7", respectively.

### 4.2.2. RSAR-based feature selection for ICPPTD

Lower and upper approximations, and reducts, play important roles in rule induction, where a rough set is a formal approximation of a crisp set in terms of a pair of sets that give the lower and upper approximations of the original set. Lower approximation is a description of the domain objects that are known with certainty to belong to the subset of interest, whereas the upper approximation is a description of the objects that might belong to the subset.

The study of rough set classification involves computing the reduct sets, but in order to create the reducts, which are subset vectors of attributes, classification rules are generated into minimal subsets. The rough set theory-based application ROSETTA [28] and Johnson's reduction algorithm [16], were both used for this purpose. Suppose that the IC package product type dataset is $I = (U, A)$ denotes an approximation space, where $U = \{x_1, x_2, \ldots, x_n\}$ denotes the set of objects representing the universe and $A = \{a_1, a_2, \ldots, a_m\}$ denotes the set of attributes (conditionals) such that $a : U \to V_a,\ a \in A$, where the value set (Va) is the set of values for $a$. When an RST decision table is denoted as $T = (U, A, C, D)$, the attributes in $A$ are further classified into two disjoint subsets of $C$ and $D$, called the condition and decision attributes, respectively: $A = C \cup D, C \cap D = \emptyset$.

For an arbitrary set $X \subseteq U$ and attribute set $R$, a set of samples that satisfy $R$ is denoted by $[x]_R$. Two approximations are then defined: the $R$-lower approximation of $X$ (denoted by $\underline{R}X$) and the $R$-upper approximation of $X$ (denoted by $\bar{R}X$). The boundary set is defined as follows:

$$\underline{R}X = \{x | [x]_R \subseteq X\} \quad (1)$$

$$\bar{R}X = \{x | [x]_R \cap X \neq \emptyset\} \quad (2)$$

$$R\text{-boundary region of } X = \bar{R}X - \underline{R}X \quad (3)$$

RST uses classification accuracy and coverage to measure the degree of sufficiency and necessity, respectively. According to the notations above, classification accuracy and coverage is defined as:

$$\alpha_R(D) = \frac{|[x]_R \cap D|}{|[x]_R|} \quad (4)$$

$$\beta_R(D) = \frac{|[x]_R \cap D|}{|D|} \tag{5}$$

where $\alpha_R(D)$ denotes a classification accuracy of $R$ with respect to classification of $D$ and $\beta_R(D)$ denotes a coverage of $R$ to $D$ .

One of the most important aspects of RST is its indiscernibility relation. The $R$-indiscernibility relation is denoted by $IND(R)$, which is an equivalence relation, and is defined as:

$$IND_I(R) = \{(x,x') \in U^2 | \forall_a \in R, \quad a(x) = a(x')\} \tag{6}$$

where $a(x)$ denotes the value of attribute $a$ of object $x$. If $(x,x') \in IND_I(R)$, then objects $x$ and $x'$ are indistinguishable from each other by attributes of $R$.

The concept of the indiscernibility relation proves that a reduction in the space of attributes is possible, so the idea is to keep only those attributes that preserve the indiscernibility relation. Since attribute reduction techniques eliminate superfluous attributes and create a minimal sufficient subset of attributes of considered knowledge, attribute dependency is considered an important factor in attribute reduction. A reduct is a minimal set of attributes $R \subseteq A$ such that,

$$IND_I(R) = IND_I(A) \tag{7}$$

The problem of finding the minimal reduct of attributes which can describe all of the concepts of the given dataset is an NP-hard problem. However, algorithms to reduce computational intensity have been proposed. Using a discernibility matrix to store the differences between attribute values for each pair of data samples, the need to search through an entire training set to detect redundant attributes can be eliminated. Experiments were then executed using the rough set theory-based application ROSETTA [28], using Johnson's reduction algorithm [16], which invokes a variation of a simple greedy algorithm to compute a single reduct. This algorithm has a natural bias toward finding a single prime deduction of minimal length, and the reduct $R$ was found by executing the following algorithm:

1. Let $R = \emptyset$.
2. Maximizes $\sum w(A)$, where $w(A)$ denotes weighted $A$. The sum is taken over all sets $A$, where $A$ contains $k$. Currently, ties are resolved arbitrarily.
3. $R \leftarrow R \cup k$.
4. Remove all sets $A$ that contain $k$ .
5. If $A = \emptyset$; output $R$. Otherwise, go back to step 2.

An example of a decision system is shown in Table 2, where the data set $I$ consists of four objects, with attributes each representing a brand of car. The set of attributes is given by $A = \{\{a1, a2, a3\}, \{a1, a4\}, \{a2, a4\}, \{a1, a3\}\}$. To find the reduct for our example, the Johnson's reduction algorithm is employed. The reduction process is shown below.

1. Let $R = \emptyset$.
2. Find the attribute $k = a1$.
3. Add the attribute $k$ to $R$, $R = \{a1\}$.
4. Remove all sets containing $k = a1$ from $A$, $A = \{a2, a4\}$.
5. $A \neq \emptyset$, go to step 2.
6. Find the attribute $k = a2$.
7. Add the attribute $k$ to $R$, $R = \{a1, a2\}$
8. Remove all sets containing $k = a2$ from $A$.
9. $A = \emptyset$, therefore the minimal reduct is $R = \{a1, a2\}$

**Table 2**
An example dataset.

| U | a1 | a2 | a3 | a4 |
|---|---|---|---|---|
| X1 | 1 | 1 | 1 | 0 |
| X2 | 1 | 0 | 0 | 1 |
| X3 | 0 | 1 | 0 | 1 |
| X4 | 1 | 0 | 1 | 0 |

Finding reducts is a method of finding dependencies in the data. The algorithm yielded one reduct that consisted of significant attributes.

ROSETTA is a toolkit application that allows for the analysis of tabular data using rough set methodology. It is a Windows application with a GUI front-end and computational kernel. The ICPPTD dataset was loaded in ROSETTA from an external data source via open database connectivity (ODBC). Using ROSETTA, the entire experimental feature selection process, from data completion to data classification, can be carried out. Once the data is loaded into ROSETTA, it is divided into a training set and a testing set. This step randomly partitions the messages in the ICPPTD into two distinct datasets. The format of the split training and testing sets is identical. The 70–30% and 80–20% partition datasets are divided into standard training and test subsets. On each partition of both datasets, feature selection and classifier design are performed on the training subset, and classification accuracy is evaluated on the test subset. A random set of 1748 (70–30% partition) and 1996 (80–20% partition) objects from the ICPPTD were chosen for the initial training set, and the remaining 748 and 500 objects were used for the testing set. These objects were used during training and for validation during testing. In the case of the RSAR experiment, we employed the above minimal reduct attribute sub-dataset as the input vector to the BPNN network.

### 4.2.3. IC Package Product Type Classifier (ICPPTC)

In this step, ICPPTC training can be made more efficient if certain preprocessing steps are performed on the network targets. The reduced dataset was preprocessed with transformation encoding; one binary coding scheme was applied to each target (nominal variables). For example, in Table 1, the product type classes of "A," "B," "C," "D" and "E" were encoded as $(1, -1, -1)$, $(1, 1, -1)$, $(-1, 1, -1)$, $(1, -1, 1)$ and $(-1, -1, 1)$, respectively. The BPNN neural network was applied for classification use, reduced by RSAR and the training and testing sets. In the training stage, the training set was randomly selected from the overall dataset ($R$) and the remaining samples were used for testing. Next, we created the feed-forward neural network. The network was formed with two layers of neurons: one hidden layer and one output layer of feed-forward BPNN. Initially, we embedded nine neurons in the hidden layer with "tansigmoid" transfer functions and three neurons in the output layer with "satlins" transfer functions. The training stopped when the performance function dropped below the set goals which were associated with the earlier stopping techniques (trainlm algorithm) to improve generalization. In this technique the available data was divided by the training set into two subsets. For example, in the case of an 80–20% training–testing partition, the first subset is the training set (70%), used for computing the gradient and updating the network weights and biases, and the second subset is the validation set (10%). The error of the validation set was monitored during the training process; normally it will decrease during the initial phase of training, along with training set error. At this point, some modification of the default training parameters can be done.

- net.trainParam.lr = 0.3
- net.trainParam.mu = 0.01
- net.trainParam.mu_dec = 0.1
- net.trainParam.mu_inc = 15
- net.trainParam.mc = 0.9
- net.trainParam.epochs = 2000
- net.trainParam.goal = 0.00001

**Table 3**
Dimensionality-reduced results of ICPPD based feature selection techniques.

| Dimensionality | PCA explain variance (%) | Reduced subset | |
|---|---|---|---|
| | | FC method | RSAR method |
| 3 | 89 | {v10, v9, v11} | {v1, v2, v8} |
| 4 | 92 | {v10, v9, v11, v2} | {v1, v2, v8, v9} |
| 5 | 93 | {v10, v9, v11, v2, v8} | {v1, v2, v3, v8, v9} |
| 6 | 94 | {v10, v9, v11, v2, v8, v1} | {v1, v2, v3, v7, v8, v9} |
| 7 | 94.5 | {v10, v9, v11, v2, v8, v1, v7} | {v1, v2, v3, v6, v7, v8, v9} |
| 8 | 95 | {v10, v9, v11, v2, v8, v1, v7, v4} | {v1, v2, v3, v5, v6, v7, v8, v9} |
| 9 | 97 | {v10, v9, v11, v2, v8, v1, v7, v4, v3} | {v1, v2, v3, v5, v6, v7, v8, v9, v10} |
| 10 | 98 | {v10, v9, v11, v2, v8, v1, v7, v4, v3, v6} | – |
| 11 | 99 | {v10, v9, v11, v2, v8, v1, v7, v4, v3, v6, v5} | – |

The ICPPTC simulation was performed using MATLAB® software. The system configuration of the computer used for training both models was as follows: operating system Windows XP; processor, Intel Core 2 Duo T7500; total physical memory was 1.5 GB. Meanwhile, classification knowledge was applied to the training and the testing set to see if it could classify them both correctly. We also wanted to observe its accuracy and coverage. However, these package family subsets contained objects with a different number of inputs and targets; therefore the various binning numbers of the subset were merged into the training set and the testing set. Finally, a well-trained ICPPTC provided a construct with two relatively consistent training and testing Root Mean Squared Errors (RMSEs).

### 4.3. Results of experiments

Four types of base classifiers were used in this study: PCA_BPNN, FC_BPNN, RSAR_BNN and RSAR. The classification performance was evaluated by using the reduced dataset with 70–30% training–testing partition and 80–20% training–testing partition, while the average classification accuracy (over 10 independent runs of the experiments) of the test data was preferred. Firstly, the results of RSAR were compared systematically to those obtained via the use of PCA and FC, as summarized in Table 3. Next, the classification performance of classifier BPNN using these reduced datasets was evaluated, based on the early stopping strategy. Using this technique, the available data was divided into

**Table 4**
Average classification accuracy and standard deviation of the feature selection techniques for two proportions of training samples.

| Dimensionality | 80–20% partition | | | | 70–30% partition | | | |
|---|---|---|---|---|---|---|---|---|
| | PCA_BPNN | FC_BPNN | RSAR_BPNN | RSAR | PCA_BPNN | FC_BPNN | RSAR_BPNN | RSAR |
| 3 | 0.8622 ± 0.1055 | 0.8280 ± 0.0337 | 0.8052 ± 0.0318 | 0.8016 | 0.8792 ± 0.0389 | 0.8226 ± 0.0177 | 0.7786 ± 0.0148 | 0.7864 |
| 4 | 0.8768 ± 0.0641 | 0.8548 ± 0.1233 | **0.9256 ± 0.0079** | **0.9516** | **0.9152 ± 0.0500** | 0.8912 ± 0.0610 | **0.9194 ± 0.0537** | **0.9399** |
| 5 | 0.8812 ± 0.0638 | 0.9060 ± 0.0645 | **0.9428 ± 0.0399** | **0.9459** | **0.9158 ± 0.0084** | **0.9206 ± 0.0052** | **0.9264 ± 0.0541** | **0.9439** |
| 6 | 0.8924 ± 0.0633 | 0.8828 ± 0.1606 | **0.9496 ± 0.0463** | **0.9198** | **0.9318 ± 0.0057** | **0.9272 ± 0.0501** | **0.9524 ± 0.0042** | **0.9332** |
| 7 | **0.9352 ± 0.0437** | 0.8876 ± 0.1445 | **0.9500 ± 0.0277** | 0.8898 | **0.9492 ± 0.0158** | **0.9324 ± 0.0427** | **0.9588 ± 0.0081** | 0.8705 |
| 8 | **0.9548 ± 0.0128** | **0.9052 ± 0.0664** | **0.9716 ± 0.0053** | 0.6473 | **0.9624 ± 0.0101** | **0.9568 ± 0.0075** | **0.9688 ± 0.0085** | 0.6235 |
| 9 | **0.9544 ± 0.0143** | **0.8940 ± 0.1189** | **0.9672 ± 0.0111** | 0.6513 | **0.9642 ± 0.0049** | **0.9528 ± 0.0206** | **0.9605 ± 0.0091** | 0.6275 |
| 10 | **0.9581 ± 0.0213** | **0.9644 ± 0.0097** | – | – | **0.9636 ± 0.0067** | **0.9632 ± 0.0061** | – | – |
| 11 | **0.9352 ± 0.0391** | **0.9476 ± 0.0393** | – | – | **0.9584 ± 0.0055** | **0.9612 ± 0.0065** | – | – |

The classification accuracies of bold values are above 90%.

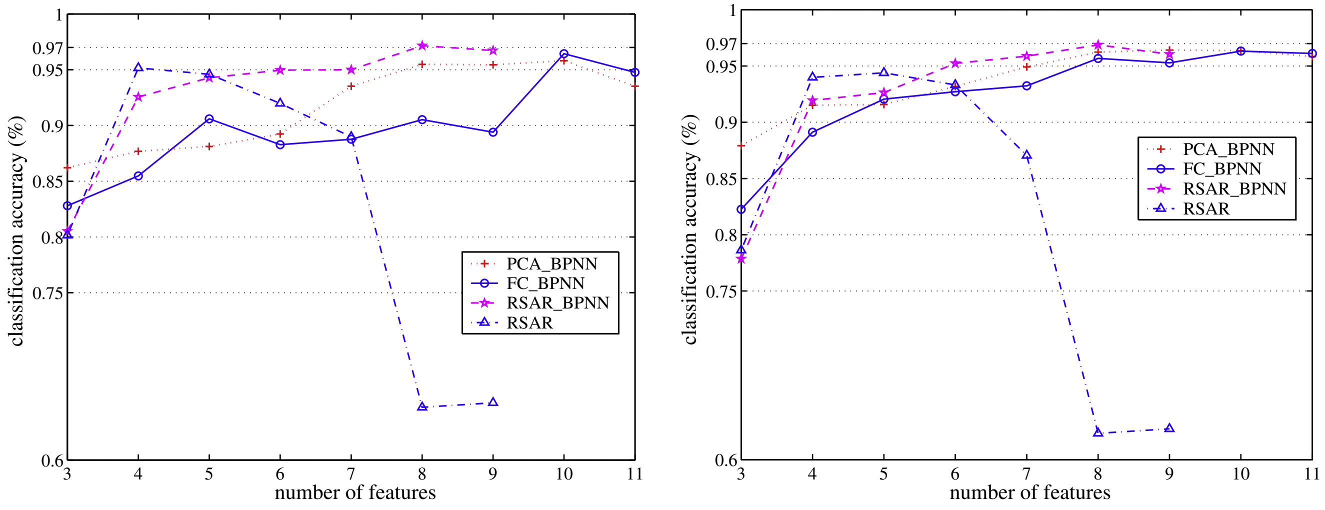

**Fig. 2.** Classification accuracy with the training rates of (a) 80–20% training-test partition, (b) 70–30% training-test partition.

**Table 5**
Experimental results of ICPPTC for reduced set based on RSAR.

| Class | A | | B | | C | | D | | E | | Accuracy | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Trial | Train set | Test set | Train set | Test set | Train set | Test set | Train set | Test set | Train set | Test set | Train_80% (1996) | Test_20% (500) |
| 1 | 508 | 124 | 333 | 91 | 284 | 68 | 658 | 158 | 213 | 59 | 0.96707 | 0.97 |
| 2 | 513 | 119 | 339 | 85 | 276 | 76 | 658 | 158 | 210 | 62 | 0.96796 | 0.972 |
| 3 | 504 | 128 | 344 | 80 | 292 | 60 | 649 | 167 | 207 | 65 | 0.96885 | 0.974 |
| 4 | 512 | 120 | 358 | 66 | 281 | 71 | 631 | 185 | 214 | 58 | **0.96484** | 0.972 |
| 5 | 499 | 133 | 345 | 79 | 275 | 77 | 659 | 157 | 218 | 54 | 0.96929 | **0.966** |
| 6 | 510 | 122 | 336 | 88 | 279 | 73 | 658 | 158 | 213 | 59 | 0.96885 | 0.97 |
| 7 | 487 | 145 | 339 | 85 | 273 | 79 | 680 | 136 | 217 | 55 | 0.97018 | 0.974 |
| 8 | 513 | 119 | 322 | 102 | 279 | 73 | 660 | 156 | 222 | 50 | **0.97063** | 0.974 |
| 9 | 500 | 132 | 346 | 78 | 275 | 77 | 661 | 155 | 214 | 58 | 0.96996 | **0.982** |
| 10 | 508 | 124 | 335 | 89 | 273 | 79 | 652 | 164 | 228 | 44 | 0.96929 | 0.962 |
| Max. | 513 | 145 | 358 | 102 | 292 | 79 | 680 | 185 | 228 | 65 | **0.97063** | **0.982** |
| Min. | 487 | 119 | 322 | 66 | 273 | 60 | 631 | 136 | 207 | 44 | 0.96484 | 0.966 |
| Avg. | 505.4 | 126.6 | 339.7 | 84.3 | 278.7 | 73.3 | 656.6 | 159.4 | 215.6 | 56.4 | **0.96869** | **0.9716** |
| Avg. of Accuracy | 0.93750 | 0.94307 | 0.90892 | 0.91358 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.96869 | 0.9716 |
| Avg. of Coverage | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |

The maximum, minimum and average classification accuracies for training and testing results have been shown in bold.

**Table 6**
Experimental results of ICPPTC for complete dataset (without feature selection).

| Class | A | | B | | C | | D | | E | | Accuracy | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Trial | Train set | Test set | Train set | Test set | Train set | Test set | Train set | Test set | Train Set | Test set | Train_80% (1996) | Test_20% (500) |
| 1 | 498 | 134 | 342 | 82 | 285 | 67 | 649 | 167 | 222 | 50 | 0.94023 | 0.92992 |
| 2 | 504 | 128 | 335 | 89 | 282 | 70 | 661 | 155 | 214 | 58 | **0.95263** | **0.95984** |
| 3 | 496 | 136 | 327 | 97 | 286 | 66 | 660 | 156 | 227 | 45 | 0.9321 | 0.90112 |
| 4 | 497 | 135 | 354 | 70 | 285 | 67 | 647 | 169 | 213 | 59 | 0.94585 | 0.93490 |
| 5 | 514 | 118 | 338 | 86 | 280 | 72 | 650 | 166 | 214 | 58 | 0.92131 | 0.90260 |
| 6 | 503 | 129 | 349 | 75 | 278 | 74 | 654 | 162 | 212 | 60 | 0.90581 | 0.91052 |
| 7 | 503 | 129 | 336 | 88 | 275 | 77 | 653 | 163 | 229 | 43 | **0.90148** | **0.89839** |
| 8 | 501 | 131 | 340 | 84 | 278 | 74 | 665 | 151 | 212 | 60 | 0.95176 | 0.91218 |
| 9 | 523 | 109 | 349 | 75 | 274 | 78 | 637 | 179 | 213 | 59 | 0.93283 | 0.92581 |
| 10 | 501 | 131 | 329 | 95 | 276 | 76 | 669 | 147 | 221 | 51 | 0.91897 | 0.90175 |
| Max. | 523 | 136 | 354 | 97 | 286 | 78 | 669 | 179 | 229 | 60 | **0.95263** | **0.95984** |
| Min. | 496 | 109 | 327 | 70 | 274 | 66 | 637 | 147 | 212 | 43 | 0.90148 | 0.89839 |
| Avg. | 504 | 128 | 339.9 | 84.1 | 279.9 | 72.1 | 654.5 | 161.5 | 217.7 | 54.3 | **0.9303** | **0.91770** |
| Avg. of accuracy | 0.94174 | 0.92152 | 0.91150 | 0.90454 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1 | 0.9303 | 0.91770 |
| Avg. of coverage | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |

Bold values are significant.

**Table 7**
Nineth trial classification results of reduced set based on RSAR.

| Package family | A | B | C | D | E | No. of. Obj | Accuracy | Coverage |
|---|---|---|---|---|---|---|---|---|
| *Summary of experimental results (training data)* | | | | | | | | |
| A | 473 | 27 | 0 | 0 | 0 | 500 | 0.946000 | 1 |
| B | 31 | 313 | 0 | 2 | 0 | 346 | 0.904624 | 1 |
| C | 0 | 0 | 275 | 0 | 0 | 275 | 1 | 1 |
| D | 0 | 0 | 0 | 661 | 0 | 661 | 1 | 1 |
| E | 0 | 0 | 0 | 0 | 214 | 214 | 1 | 1 |
| True positive rate | 0.946 | 0.9017 | 1.00 | 1.00 | 1.00 | | 0.9696994 | 1 |
| Total number of tested objects: 1996 | | | | | | | | |
| Total accuracy: 0.969439 | | | | | | | | |
| Total coverage: 1.00 | | | | | | | | |
| *Summary of experimental results (testing data)* | | | | | | | | |
| A | 129 | 3 | 0 | 0 | 0 | 132 | 0.977273 | 1 |
| B | 5 | 72 | 0 | 1 | 0 | 78 | 0.923077 | 1 |
| C | 0 | 0 | 77 | 0 | 0 | 77 | 1 | 1 |
| D | 0 | 0 | 0 | 155 | 0 | 155 | 1 | 1 |
| E | 0 | 0 | 0 | 0 | 58 | 58 | 1 | 1 |
| True positive rate | 0.9773 | 0.923 | 1.00 | 1.00 | 1.00 | | 0.982 | 1 |
| Total number of tested objects: 500 | | | | | | | | |
| Total accuracy: 0.982 | | | | | | | | |
| Total coverage: 1.00 | | | | | | | | |

three subsets. The first subset was the training set, used for computing the gradient and updating the network weights and biases. The second subset was the validation set—its error was monitored during the training process. When the validation error increases after a specified number of iterations, the training is stopped, and the weights and biases at the minimum of the validation error are returned. Table 4 shows the average classification accuracy and standard deviation of the test data and dimensionality of features for the classifiers RSAR_BPNN, PCA_BPNN, FC_BPNN and RSAR, respectively. It is noted that the performance of the RSAR-based (RSAR and RSAR_BPNN) method dramatically increases with the removal of features. When comparing the average accuracies of these classifiers for dimensionality 8 and 9, the RSAR_BPNN classifiers are found to provide the highest accuracy. Comparisons of the RSAR-based methods and the corresponding hybrid methods are shown in Fig. 2. This study has shown that the performance of RSAR_BPNN method is improved, and that in general, RSAR_BPNN benefits more than BPNN when combines with the PCA and FC methods.

The results from the previous experiment were then used to obtain a measure of the performance of the generated classifiers. Ten classification trial results for the complete ICPPTD dataset (without feature selection) and reduct sets based on RSAR were used: their corresponding accuracies are shown in Tables 5 and 6. In Table 5, the total coverage is 100%, and the average classification accuracies of the training set and the testing set are 96.869% and 97.16%, respectively. The minimum classification accuracy of the testing set is the 5th trial (96.66%) and the maximum classification accuracy of the testing set is the 9th trial (98.2%). Table 6, however, shows that although the total coverage is 100%, the average

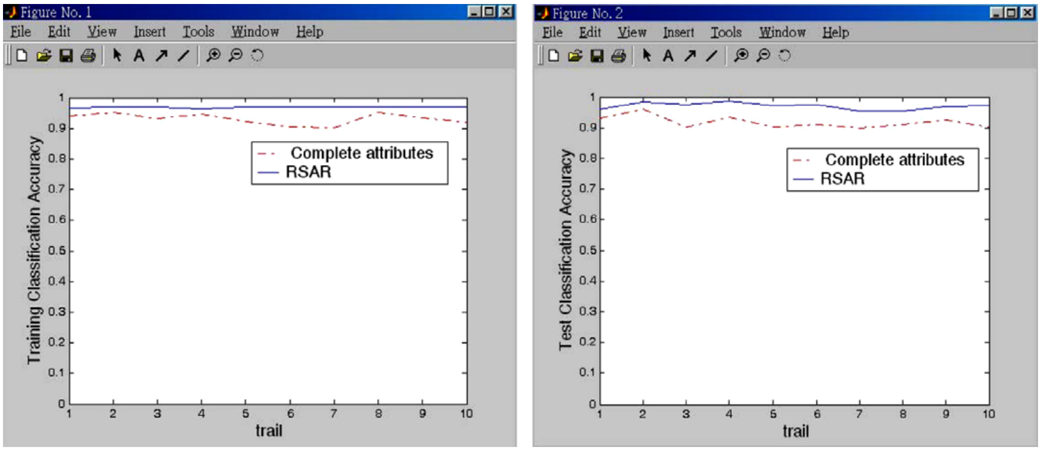

**Fig. 3.** Comparison of the classification results of the 10 trials of RSAR and the complete dataset.

**Table 8**
SPSS printout of mean and descriptive statistics for classification accuracy of samples.

| Statistical descriptive | Complete ICPTD dataset | | Reduct set based on RSAR | |
|---|---|---|---|---|
| | Statistic | Std. error | Statistic | Std. error |
| *Training accuracy* | | | | |
| Mean | 0.93029548 | 0.005723843 | 0.96866934 | 0.000535691 |
| 95% Confidence interval for mean | | | | |
| Lower bound | 0.91734725 | | 0.96745752 | |
| Upper bound | 0.94324371 | | 0.96988115 | |
| 5% Trimmed mean | 0.93065595 | | 0.96877318 | |
| Median | 0.93246096 | | 0.96906987 | |
| Variance | 0.000 | | 0.000 | |
| Std. deviation | 0.01810038 | | 0.001694004 | |
| Minimum | 0.901475 | | 0.964842 | |
| Maximum | 0.952628 | | 0.970628 | |
| Range | 0.051152 | | 0.005785 | |
| Interquartile range | 0.031645 | | 0.002114 | |
| Skewness | −0.351 | 0.687 | −1.367 | 0.687 |
| Kurtosis | −1.083 | 1.334 | 2.100 | 1.334 |
| *Testing accuracy* | | | | |
| Mean | 0.91770412 | 0.006234998 | 0.97068273 | 0.003493437 |
| 95% Confidence interval for mean | | | | |
| Lower bound | 0.90359958 | | 0.96278003 | |
| Upper bound | 0.93180867 | | 0.97858543 | |
| 5% Trimmed mean | 0.91643609 | | 0.97054886 | |
| Median | 0.91135020 | | 0.97188755 | |
| Variance | 0.000 | | 0.000 | |
| Std. deviation | 0.019716795 | | 0.011047219 | |
| Minimum | 0.898394 | | 0.955823 | |
| Maximum | 0.959839 | | 0.987952 | |
| Range | 0.061446 | | 0.032129 | |
| Interquartile range | 0.029569 | | 0.019076 | |
| Skewness | 1.131 | 0.687 | −0.010 | 0.687 |
| Kurtosis | 0.861 | 1.334 | −0.930 | 1.334 |

classification accuracies of the training set and the testing set are as low as 93.03% and 91.77%, respectively, and that the minimum and maximum classification accuracies of the testing set in the 7th trial are 90.148% and 89.839%, respectively. These results are shown in bold.

Detailed information on the 9th trial is also given in Table 7. Through ICPPTC classification, we obtain a total of 132 objects of right class "A" data, 78 objects of right class "B" data, 77 objects of right class "C" data, 155 objects of right class "D" data and 58 objects of right class "E" data. Referring to A in Table 7's test experiment results as an example, the sensitivity, or the ratio between the predicted and real values of predicting classification "A" is 129/132 = 0.977273, while those for "B," "C," "D" and "E" are 0.923077, 1.00, 1.00 and 1.00, respectively.

The experimental results of the complete ICPPTD datasets and reduced sets are shown in Fig. 3. Table 8 shows that the mean and standard deviations of the reduced sets are superior to those of the complete ICPPTD sets in the testing set and that the features produced by RSAR improve the generalization accuracy. Training and testing set classification accuracies are 96.9% and 98.2%, respectively, and from the experimental results, we can conclude that the reduct gained by RSAR exhibits higher classification accuracy than those without feature selection.

## 5. Conclusion

In this paper we have tested the use of the RSAR_BPNN method to assess the classification accuracy of IC package type selection from a real database. The IC's design specifications are also classified by the RSAR_BPNN model, using rules derived inductively from the data to overcome the shortcomings of methods traditionally applied in the semiconductor industry. IC package product classification technology, based on rough sets and neural networks is also presented, and we have demonstrated that rough sets theory is able to be applied successfully to feature reduction for larger datasets. We also compared the performance of neural networks, with and without rough set preprocessing, and discovered that when the number of features is low, the ICPPTC classifier offers better performance. For example, RSAR can remove redundant ineffective attributes for the ICPPTC classifier, and as a result, those feature values that have very little effect on classification accuracy, are reduced. The accuracy of TFBGA and LGA classifications, on the other hand, are found to be the worst, at 97.7273% and 92.3077%, respectively. In IC package factories, TFBGA and LGA are the most difficult to distinguish (the most similar results), so future research could further explore these two methods to improve the accuracy rate of the whole classification. In the end, we found that the overall performance of ICPPTC based on RSAR was better than the performance when rough set preprocessing was not used. We have also demonstrated an effective RSAR_BPNN technique for dealing with a large amount of information and multiclass IC package product classification activity. It should also be noted here that the results here presented, this research demonstrates an advance on previous research because it was used to predict a class of performance that the IC design type should belong to, rather than an actual product classification. Beyond the attractive accuracy results, these models could also be adapted for other IC product designs, where the particular parameters within the testing datasets of the particular package types could be further altered for experimental purposes.

## Acknowledgements

The authors thank the anonymous referees for their careful reading of the paper and for making several suggestions that improved it. The authors also thank the National Science Council of the Republic of China for financially supporting this research under Contract No. 96-2221-E-167-022.

## Appendix A. Supplementary data

Supplementary data associated with this article can be found in the online version, at doi:10.1016/j.aei.2009.04.001.

## References

[1] G. Acciani, G. Brunetti, G. Fornarelli, A multiple neural network system to classify solder joints on integrated circuits, International Journal of Computational Intelligence Research 2 (4) (2006) 337–348.

[2] A.L. Blum, P. Langley, Selection of relevant features and examples in machine learning, Artificial Intelligence 97 (1997) 245–271.

[3] L.H. Chen, T.Y. Wang, Artificial neural networks to classify mean shifts from multivariate $\chi^2$ chart signals, Computers & Industrial Engineering 47 (2–3) (2004) 195–205.

[4] T. Chen, An intelligent hybrid system for wafer lot output time prediction, Advanced Engineering Informatics 21 (2007) 55–65.

[5] Z. Chen, S. Lu, S. Lam, A hybrid system for SPC concurrent pattern recognition, Advanced Engineering Informatics 21 (2007) 303–310.

[6] A. Chouchoulas, Incremental Feature Selection Based on Rough Set Theory, Dissertation Thesis, Division of Informatics, University of Edinburgh, 2001.

[7] M. Christopher, Neural Networks for Pattern Recognition, Bishop Oxford University Press, 1995.

[8] K.J. Cios, W. Pedrycz, R.W. Swiniarski, Data Mining Methods in Knowledge Discovery, Kluwer, Boston, 1998.

[9] R. Crowley, 1999. Available from: <http://www.www.chipscalereview.com/issues/0799/columns2.htm>.

[10] Z. Cui, B. Xu, W. Zhang, D. Jiang, J. Xu, CLDA: feature selection for text categorization based on constrained LDA. Semantic computing, in: ICSC 2007, 2007, pp. 702–712.

[11] M. Dash, H. Liu, Feature selection for classifications, Intelligent Data Analysis: An International Journal 1 (1997) 131–156.

[12] S. Haykin, Neural Networks: A Comprehensive Foundation, Wiley & Sons, NJ, 1994.

[13] M.L. Huang, Y.H. Hung, Combining radial basis function neural network and genetic algorithm to improve HDD driver IC chip scale package assembly yield, Expert Systems with Applications 34 (1) (2008) 588–595.

[14] Y.H. Hung, The IC package product type dataset, http://blog.ncut.edu.tw/meworksv2a/meworks/page1.aspx?no=2763, National Chin-Yi University of Technology, Department of Department of Industrial Engineering and Management, 2008.

[15] R. Jensen, Q. Shen, Fuzzy-rough attribute reduction with application to web categorization, Fuzzy Sets and Systems 141 (3) (2004) 469–485.

[16] D.S. Johnson, Approximation algorithms for combinatorial problems, Journal of Computer and System Sciences 9 (1974) 256–278.

[17] B.S. Joo, N.J. Choi, Y.S. Lee, J.W. Lim, B.H. Kang, D.D. Lee, Pattern recognition of gas sensor array using characteristics of impedance, Sensors and Actuators B Chemical 77 (1–2) (2001) 209–214.

[18] J. Kittler, Feature selection and extraction, in: T.Y. Young, K.S. Fu (Eds.), Handbook of Pattern Recognition and Image Processing, Academic Press, San Diego, 1986, pp. 59–83.

[19] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1–2) (1997) 273–324.

[20] I. Kononenko, Estimating attributes: analysis and extension of RELIEF, in: Proceeding of European Conference on Machine Learning, Berlin, 1994, pp. 171–182.

[21] R.J. Kuo, L.M. Ho, C.M. Hu, Cluster analysis in industrial market segmentation through artificial neural network, Computers & Industrial Engineering 42 (2–4) (2002) 391–399.

[22] N. Kwak, C.H. Choi, Input feature selection by mutual information based on Parzen window, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (12) (2002) 1667–1671.

[23] D. Langoni, M.H. Weatherspoon, S.Y. Foo, H.A. Martinez, A speed and accuracy test of backpropagation and RBF neural networks for small-signal models of active devices, Engineering Applications of Artificial Intelligence 19 (2006) 883–890.

[24] M. Last, A. Kandel, O. Maimon, Information theoretic algorithm for feature selection, Pattern Recognition Letters 22 (6) (2001) 799–811.

[25] R. Li, Z. Wong, Mining classification rules using rough sets and neural networks, European Journal of Operational Research 157 (2004) 439–448.

[26] X.L. Li, Z.L. Du, T. Wang, D.M. Yu, Audio feature selection based on rough set, International Journal of Information Technology 11 (6) (2005) 117–123.

[27] T. Miyamoto, S. Uchimura, Y. Hamamoto, N. Iizuka, M. Oka, H. Yamada-Okabe, Comparative study of feature selection methods on microarray data, Biomedical Engineering, 2003, in: IEEE EMBS Asian-Pacific Conference on 20–22 October 2003, pp. 82–83.

[28] A. Øhrn, The ROSETTA homepage (http://rosetta.lcb.uu.se/), Norwegian University of Science and Technology, Department of Computer and Information Science, 2000.

[29] N.M. Parthalain, Q. Shen, R. Jensen, Distance measure assisted rough set feature selection, in: Fuzzy Systems Conference, FUZZ-IEEE 2007, IEEE International, July 2007, pp. 1–6.
[30] Z. Pawlak, J. Grzymala-Busse, R. Slowinski, W. Ziarko, Rough sets, Communications of the ACM 38 (11) (1995) 89–95.
[31] Z. Pawlak, Rough set theory for intelligent industrial applications, in: Proceedings of the Second International Conference on Intelligent Processing and Manufacturing of Materials IPMM '99, 1999, pp. 37–44.
[32] Z. Pawlak, Rough sets, International Journal of Computer and Information Science 1 (5) (1982) 341–356.
[33] Z. Pawlak, Rough Sets: Theoretical Aspects of Reasoning about Data, Kluwer, Boston, 1991.
[34] J.F. Peters, S. Ramanna, Z. Suraj, M. Borkowski, Rough neurons: Petri net models and applications, in: L. Polkowski, S.K. Pal, A. Skowron (Eds.), Rough-Neuro-Computing: Techniques for Computing with Words, Springer-Verlag New York, Secaucus, NJ, 2002, pp. 474–493.
[35] G. Phipps, Selecting the best package for your design (http://www.ecnasiamag.com/article-12997-selectingthebestpackageforyourdesign-Asia.html), Advanced Interconnect Technologies, 2007.
[36] L. Polkowski, S.K. Pal, A. Skowron, Rough-Neuro-Computing: Techniques for Computing with Words, Springer-Verlag New York, Secaucus, NJ, 2002.
[37] Q. Shen, A. Chouchoulas, A modular approaches to generating fuzzy rules with reduced attributes for the monitoring of complex system, Engineering Applications of Artificial Intelligence 13 (3) (2000) 263–278.
[38] C.T. Su, T.L. Chiang, Optimizing the IC wire bonding process using a neural networks/genetic algorithms approach, Journal of Intelligent Manufacturing 14 (2003) 229–238.
[39] K. Thangavel, Q. Shen, A. Pethalakshmi, Application of clustering for feature selection based on rough set theory approach, AIML Journal 6 (1) (2006) 19–27.
[40] G.V. Trunk, A problem of dimensionality: a simple example, IEEE Transactions on Pattern Analysis and Machine Intelligence 1 (3) (1979) 306–307.
[41] N. Zhong, J. Don, S. Ohsuga, Using rough sets with heuristics for feature selection, Journal of Intelligent Information Systems 16 (2001) 199–214.

**Yung-Hsiang Hung** is an associate professor in Industrial Engineering and Management at National Chin-Yi University of Technology. He received his Ph.D. in Industrial Engineering and Management at National Chiao-Tung University in 2002. His main research interest is in the area of statistical process control and service quality management. His research areas include process capability analysis, semiconductor manufacturing management and process parameters optimization design.