CHAPTER 4

Results and Discussion

This chapter reports and discusses the experimental results from the proposed algorithms. The results are divided into 2 main parts: data reduction in regression and in classification. For regression problems, support vector regression (SVR) was used as a tool to evaluate the performance of the data reduction and four-fold cross validation was used to select the best model by comparing root mean square errors (RMSE). For classification problems, 1NN was used to evaluate the performance of the data reduction. Besides 1NN, we also used the SVM and ANN in order to show how applicable the proposed algorithm was. The proposed methods were applied to both synthesized and real-world data sets, and also the performances were compared to the existing data reduction algorithms.

4.1 Result of data reduction on regression problems

The performance of the proposed algorithm was evaluated by using synthesized data, the data from the University of California, Irvine [27], and the electrocardiogram data from PhysioNet [28]. After applying the proposed algorithm to each data set to reduce the number of samples, the obtained result was used as a training set for SVR to generate a regression model. We chose SVR over ANN to form the regression model because ANN has too many parameters to control such as the number of hidden layers, the number of neurons on each hidden layer, the activation functions, the number of training epochs, the learning rate, and the momentum term. In contrast, SVR requires only three predefined parameters (σ, C, ε) if the radial basis kernel function is selected. Among various parameters, the best SVR model was selected by 4-fold cross validation whose minimum root means square error (RMSE) was minimal.

The experiment on data reduction for regression problems was conducted on 4 types of data. The first type was data with only one-dimensional input feature representing low complex data. The second type was data with 2-8 dimensional input features to show

the ability of the proposed method to reduce data with higher complexity. Both first and second data types were all synthesized data. Next is the third data type, which was the Automobile miles per gallon (MPG) data, representing a small real-world data set. Finally, the data called "1990 census in California" was used as the forth type to represent a large data set. Both third and fourth data types are examples of real-world data set.

4.1.1 One-dimensional synthesized data set

In the first experiment, we tested our method with a basic function approximation problem with one-dimensional input and output

$$y = 0.6\sin(\pi x) + 0.3\sin(3\pi x) + 0.1\sin(5\pi x) \quad (4.1)$$

where $x \in [-1,1]$. We randomly generated two data sets with two hundred and four hundred samples. Each data set was then equally divided into training and testing samples.

In the initialization of the reduction process, we did not split the training data (m = 1) because the sample size was not large. We used four different quantization levels (q = 10, 50, 100, 200) to observe the change in accuracy and reduction ratio in relation to the increase in the number of quantization levels. All weight values of the cluster validation indices in TOPSIS process was 0.2. Since the result from of each cluster validation indices differed depending on the characteristic of the data, and also the characteristic of the data is random and unpredictable, all weights were assigned to be equal.

The result from the increase in quantization levels corresponding to the decrease in the data reduction ability is shown in Figure 4.1. Since the gap between each quantization level is narrower, the number of data and clusters in each quantized level is also decreased. This leads to the decrease in the performance of the reduction ratio. From the experimental result, the reduction rate is about 40% when the numbers of quantization levels are 100 and 200. The proposed method could not further reduce the data by increasing the number of quantization levels because the data set has only 127 distinct values from 200 samples. Therefore, the reduction ratio was

reduced and close to a certain value when the number of quantization levels was close to the number of data values.

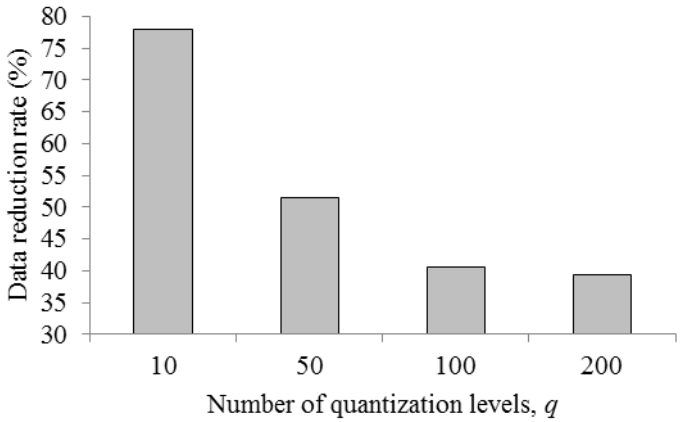


Figure 4.1 Relationship between data reduction rate and the number of quantization levels.

On the other hand, when the number of partitions is smaller, the number of data converges to the original. In Figure 4.2, the RMSE will be close to the RMSE from the original data when the number of quantization levels increases over one hundred. Because the number of data points in each level using the smaller q was larger than the number of data point in each level using the higher q. Therefore, the computation time for the reduction process decreased when the number of quantization levels increased as shown in Figure 4.3.

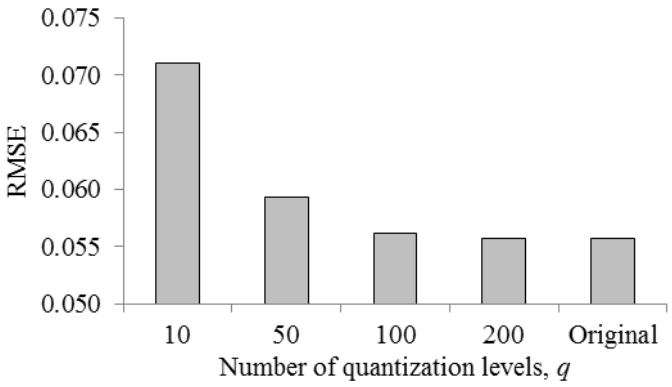


Figure 4.2 Relationship between RMSE and the number of quantization levels.

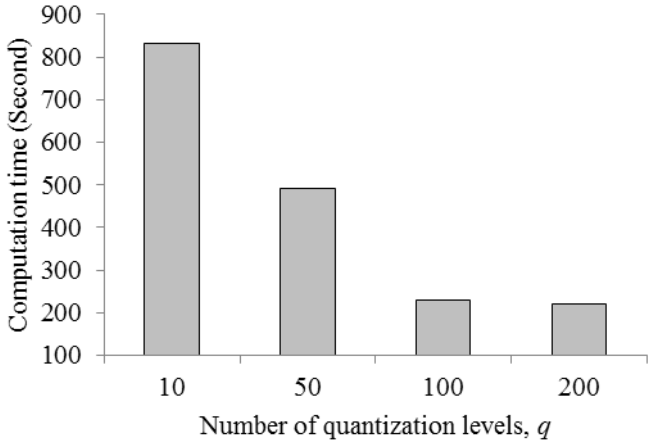
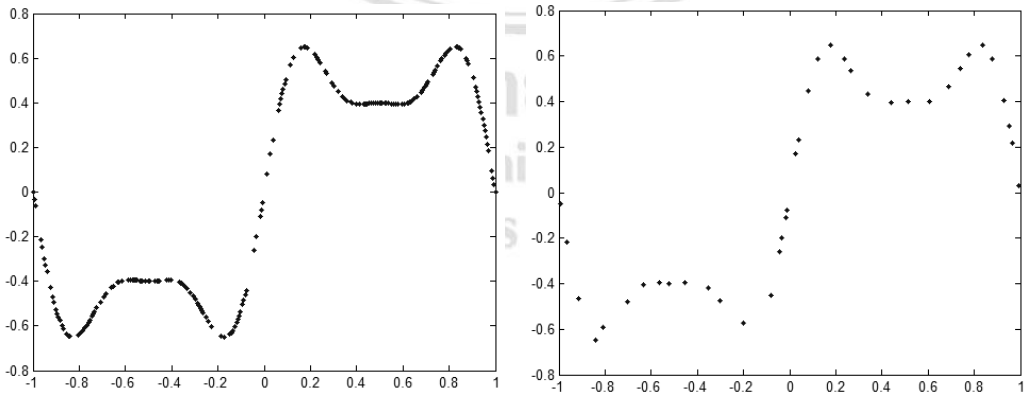


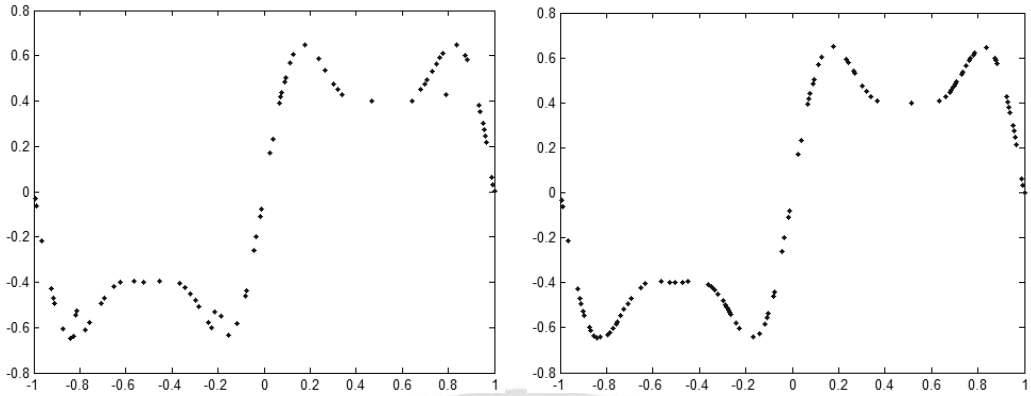
Figure 4.3 Relationship between computation time for the reduction process and the number of quantization levels.

The reduced data from different quantization levels are presented in Figure 4.4 with the original data in Figure 4.4 (a). Figure 4.4 (b) shows that the reduction result had many errors and small number of samples when the number of quantization levels was low at q = 10. If we increased the quantization level q to 50, the number of data increased but the error decreased (Figure 4.4 (c)). When we increased the number of quantization levels, the number of data and accuracy increased and were stable at some values as shown in Figure 4.4 (d-e) (q = 100, 200).

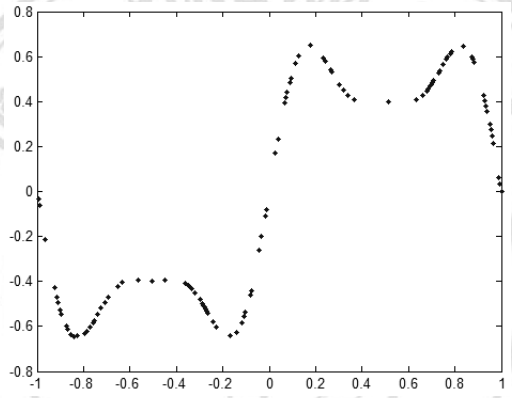


(a) (b)

Figure 4.4 Plot of (a) original data, (b)-(e) reduction results from various quantization levels (q = 10, 50, 100, 200) (cont.).



(c) (d)



(e)

Figure 4.4 Plot of (a) original data, (b)-(e) reduction results from various quantization levels (q = 10, 50, 100, 200).

Table 4.1 shows the accuracy of the proposed method compared to the results from the methods in [23] and [29] which also used the same one-dimensional function in equation (4.1) to evaluate the performance. The training and testing contain 100 samples. We found that our proposed algorithm gave a good result and low missed detection rate comparable to the result obtained from using the whole data set as a training data.

Table 4.1 Comparison between training and testing
RMSE of proposed algorithm and two others.

Models	RMSE of training result	RMSE of testing result
Di Wang [23] (The number of input samples = 100)	0.0809	0.1036
W. Pedrycz [29] (The number of input samples = 100)	0.0610	0.0680
Proposed method (The number of input samples = 100)	0.0561	0.0561
Regression model from all training data (The number of input samples = 100)	0.0557	0.0557

4.1.2 Two-dimensional synthesized data set

The two sets of training and testing data were random sampled (100 and 800 samples) from the two-dimensional input function

$$\boldsymbol{y} = \left(1 + \boldsymbol{x}_1^{-2} + \boldsymbol{x}_2^{-1.5}\right)^2 \tag{4.2}$$

where $x_1 \in [1, 5]$ and $x_2 \in [1, 5]$. The output of the function is shown in Figure 4.5.

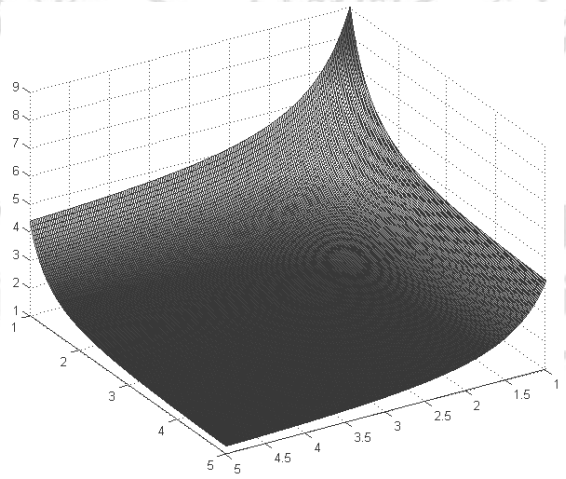


Figure 4.5 Surface area of two-dimensional input function.

We initialized the experimental parameters m = 1. The result from 4.1.1 showed that higher quantization level increases accuracy. Therefore, we set the quantization level to 100 (q = 100), and set all weight values of cluster validation indices in TOPSIS to 0.2.

The support vector regression result is presented in Table 4.2-4.3 and the comparison of the performance with other algorithms is shown in Table 4.4.

Table 4.2 Four-fold cross validation result (The number of input samples = 50).

Training			Testing		
Fold	Number of training samples	RMSE	Number of testing samples	RMSE	MAE
1	37	0.5462	50	0.2838	0.1998
2	37	**0.4596**	50	**0.2941**	**0.1951**
3	38	1.0073	50	0.3935	0.2521
4	38	0.5656	50	0.3039	0.1923

Table 4.3 Four-fold cross validation result (The number of input samples = 400).

Training			Testing		
Fold	Number of training samples	RMSE	Number of testing samples	RMSE	MAE
1	163	0.0925	400	0.0846	0.0693
2	168	0.0946	400	0.0775	0.0652
3	167	0.0925	400	0.0820	0.0697
4	161	**0.0925**	400	**0.0771**	**0.0655**

Table 4.4 Mean absolute errors from SVR for two-dimensional data set using different methods.

Models	MAE
Di Wang [23]	0.0012
Proposed method (The number of input samples = 50)	0.1951
Original data (The number of input samples = 50)	0.1834
Proposed method (The number of input samples = 400)	0.0655
Original data (The number of input samples = 400)	0.0601

Mean absolute errors (MAE) used to evaluate the performance of these methods are shown in Table 4.4. The result from the proposed method is quite less than from the proposed method in [23] because the data used to training was insufficient. The accuracy of the proposed method increased when it received sufficient data or received the whole input-output data space. Therefore, when we increased the number of input samples to cover all output space, the reconstructed accuracy also increased.

Next, we evaluate the proposed method with higher dimensional synthetic function according to Equation 4.3-4.8 to show that the proposed method has an ability to reduce a data set with higher dimension. Figure 4.6 presents the comparison between the average of regression errors from 4-fold cross validation of the reduced data sets and the original data set. In each function, the accuracy of the reduced data set and the original data was slightly different. It was usual that when the dimension of the synthetic function increases, the regression model will be more complicated and the RMSE will also increase. The regression result from each synthetic function is shown in Figure 4.7. The gray thick line is the desired output and the black thin line is the regression output.

$$y_3 = \left(1 + x_1^{-2} + x_2^{-1.5} + x_3^{-1.5}\right)^2 \tag{4.3}$$

$$y_4 = \left(1 + x_1^{-2} + x_2^{-1.5} + x_3^{-1.5} + x_4^{-1.5}\right)^2 \tag{4.4}$$

$$y_5 = \left(1 + x_1^{-2} + x_2^{-1.5} + x_3^{-1.5} + x_4^{-1.5} + x_5^{-1.5}\right)^2 \tag{4.5}$$

$$y_6 = \left(1 + x_1^{-2} + x_2^{-1.5} + x_3^{-1.5} + x_4^{-1.5} + x_5^{-1.5} + x_6^{-1.5}\right)^2 \tag{4.6}$$

$$y_7 = \left(1 + x_1^{-2} + x_2^{-1.5} + x_3^{-1.5} + x_4^{-1.5} + x_5^{-1.5} + x_6^{-1.5} + x_7^{-1.5}\right)^2 \tag{4.7}$$

$$y_8 = \left(1 + x_1^{-2} + x_2^{-1.5} + x_3^{-1.5} + x_4^{-1.5} + x_5^{-1.5} + x_6^{-1.5} + x_7^{-1.5} + x_8^{-1.5}\right)^2 \tag{4.8}$$

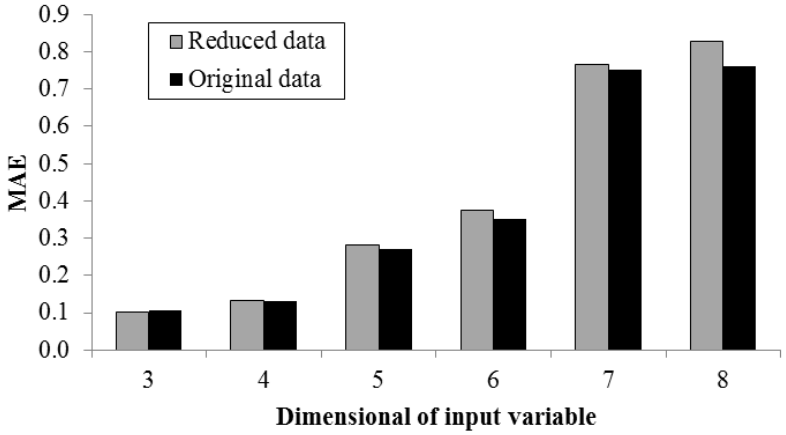


Figure 4.6 Regression result between the reduced and original data.

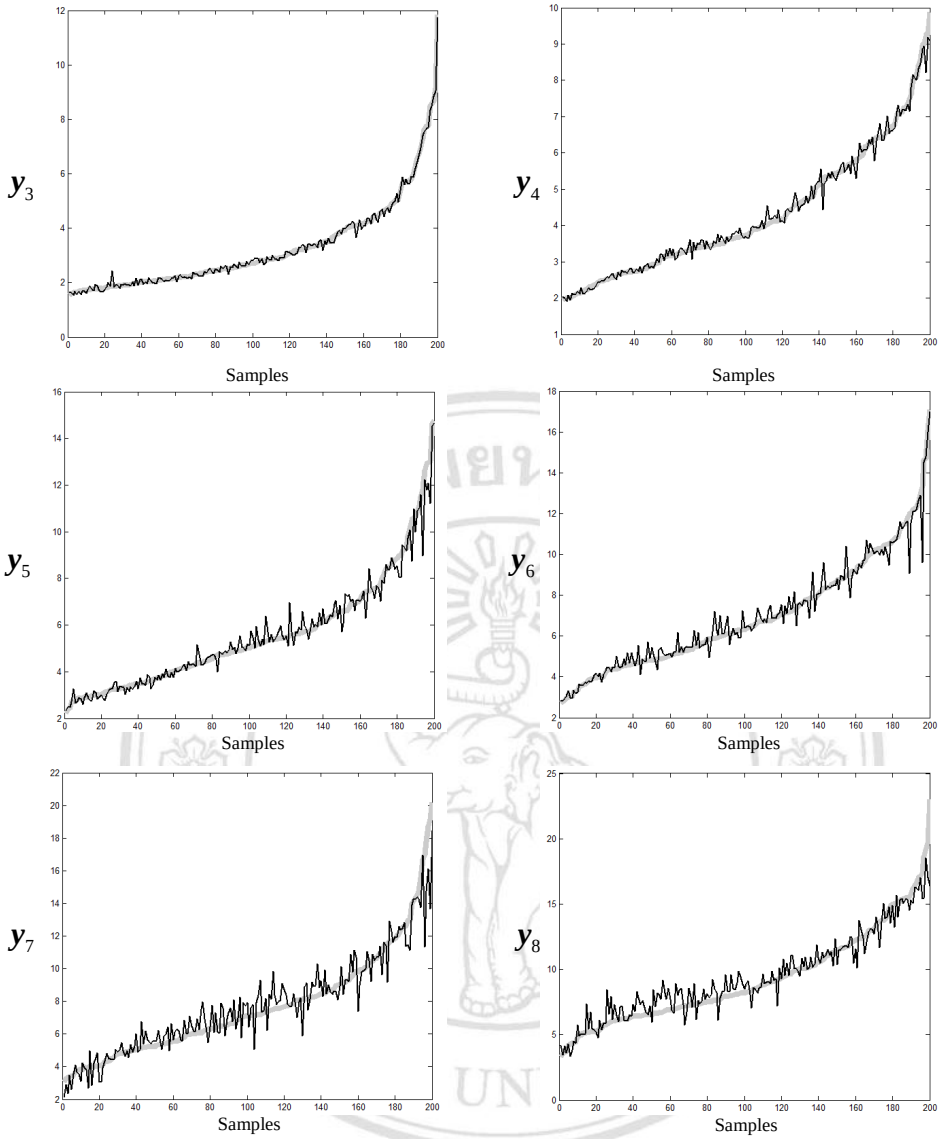


Figure 4.7 Comparison between the desired output and the regression result of each high dimensional function.

4.1.3 Real-world data set 1

The first real-world data set used for the performance evaluation of the proposed method is the Automobile Miles Per Gallon (MPG) data, from ics.uci.edu [27]. The data set contained only 392 samples. The original data set was shuffled and divided 60% into the training set and 40% into the testing set. Because of the small number of data size, we did not divide the data set into smaller part (m = 1). We set all the weight of cluster validation indices to 0.2. In this experiment, the quantization was performed at seven different quantization values, linearly increasing with twenty-five levels per

step (q = 25, 50, 75, 100, 150, 175, 200). The experiments showed that the results from the real-world data set were similar to those from the synthesized data set, although the real-world data set has 8 features while the synthesized data set has only one feature. The data decreasing indices decreased as the number of the quantization levels increased. When the quantization level equals to 100, the data decreasing indices slightly fluctuate at around 45% of the data reduction rate, see Figure 4.8. Considering the output data set, we found that the number of values in the output data set is equal to 127 values that cause data reduction rate steady at 45%, although the quantization level was increased.

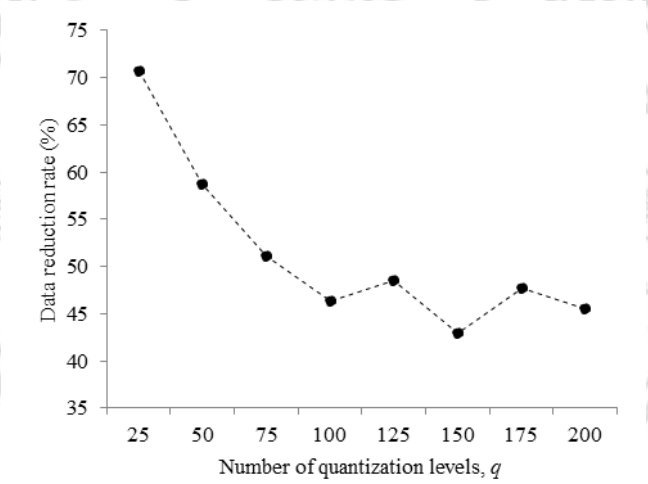


Figure 4.8 Relationship of number of quantization levels, q and data reduction rate.

Since the data set used was a real-world data set and the data in 4-fold cross validation was random, the best training result is not necessary to be derived from the data reduction with high quantization levels. This makes the number of samples close to the original as shown in Figure 4.9 where the minimum RMSE exists at the quantization level of 75.

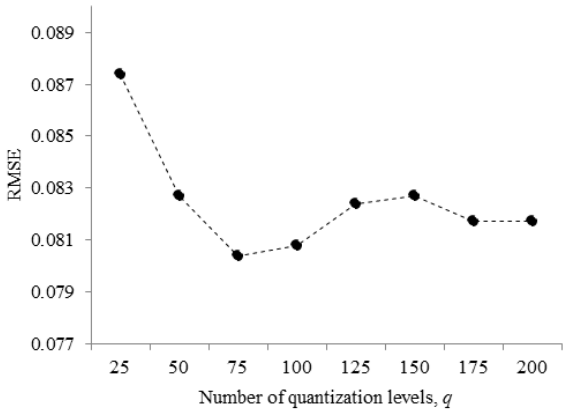


Figure 4.9 Minimum RMSE from four-fold cross validation at various quantization levels.

The average RMSE of 4-fold cross validation from the training results was against expectations where the number of quantization levels should increase while the RMSE should decrease. This is because the testing data in each experiment was random. However, the average RMSE of 4-fold cross validation from the testing results was still as expected because the blind test data set was the same data. Even though some of the results slightly increased (at q = 125 and 150), the general trend was still downward as shown in Figure 4.10. The dotted line indicates the average RMSE of 4-fold cross validation from the testing results and the bold line shows the average RMSE of 4-fold cross validation from the training results.

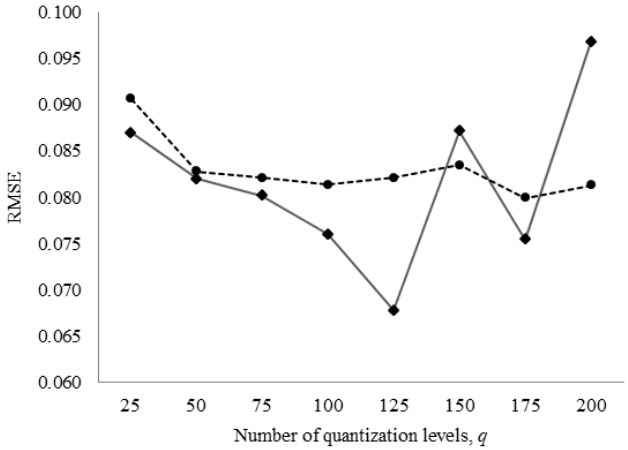


Figure 4.10 Average RMSE from four-fold cross validation at various quantization levels.

Table 4.5 shows that the RMSE from the reduced data by the proposed method is quite similar to the one from the original data. We found that our proposed method gave a little greater RMSE than the one belongs to the original for 0.08.

Table 4.5 Root-mean-square errors obtained by using original data and our proposed method.

Models	RMSE (Training)	RMSE (Testing)
Original data	2.4076	2.9429
Proposed method (q = 75)	2.6966	3.0216

4.1.4 Real world data set 2

The second real-world data set is the 1990 census in California. The data have 9 variables (nine input dimensions) with 20,640 samples. We used 10,000 samples for training and 10,640 for testing. This data set was used to compare the reconstructed result and data decreasing with different number of small parts (m = 1, 4, 10). The quantization level was set to two hundreds (q = 200) and the weight of cluster validation indices was 0.2.

Figure 4.11 provides example histograms of the original data and the small parts. It shows the similarity of the histogram shape between the original and those small parts. The five histograms below illustrates that the amount of data in every figure is divided almost evenly.

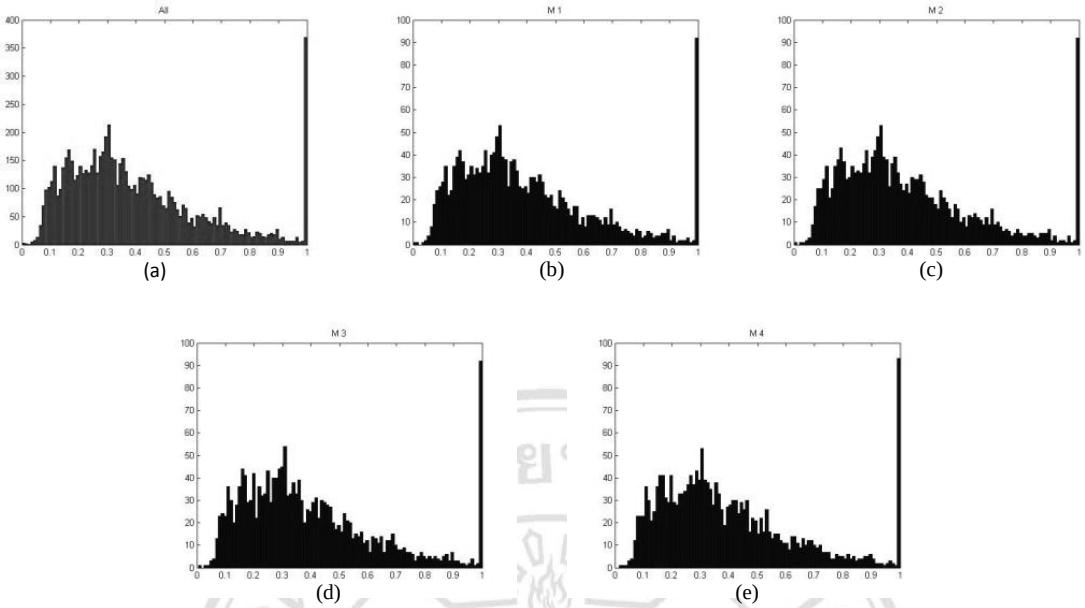


Figure 4.11 Histogram of (a) original data and (b - e) four small parts.

When we increase the number of small parts, the number of training data also increases because the number of data in each quantization level decreased as in Figure 4.12. Figure 4.13 presents the regression accuracy which increased following the number of small parts because the splitting data into small parts affected and increased the fineness of clustering.

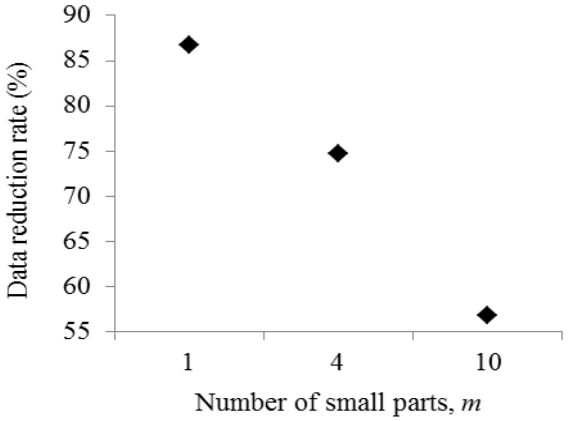


Figure 4.12 Data reduction rate versus the number of small parts.

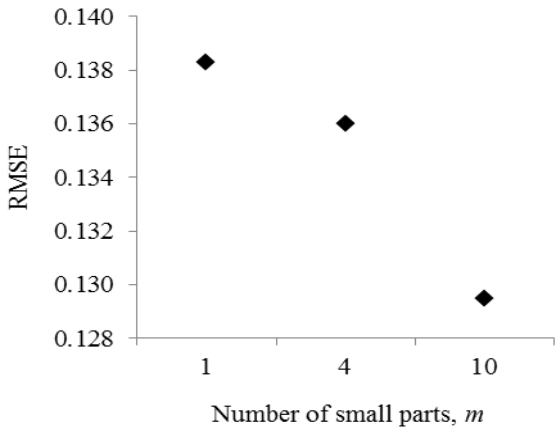


Figure 4.13 Plot between RMSE and the number of small parts.

We present the comparison of the regression result from the proposed method, the other method, and the original data in Table 4.6. It is obvious that the proposed methods give the better result than the other methods. If we increase the number of small parts, the regression accuracy converges to the regression result from the original data. Also from the table, the processing time of the splitting data to ten parts was faster than the processing with the whole data but the reduction ratio was lower. In this case, the user must consider the accuracy, the reduction ratio, the reduction time, and the training time in practice.

Table 4.6 Root-mean-square errors from SVR for real data set 1 using different methods.

Models	RMSE (Training)	RMSE (Testing)	Data reduction rate (%)	Reduction time (Second)
ECSFS [30]	0.3087	0.3760	-	-
Di Wang [23]	0.2860	0.3708	-	-
Proposed method (one part)	0.1372	0.1383	86.70	43,589
Proposed method (four parts)	0.1356	0.1360	74.65	15,193
Proposed method (ten parts)	0.1275	0.1295	56.75	8,837
Original data	0.1203	0.1234	0.00	-

4.1.5 Real world data set 3

This experiment applied the proposed method to reduce the training data set of four ECG signals for training with support vector regression. The 12-lead ECG signals used in this research were obtained from physionet.org. The data set was recorded from 75 patients. However, only 14 patients were used because there were missing data in some lead signals of some patients. We preprocessed by using fifth-order low pass digital Butterworth filter with the cut-off frequency of 0.5 Hz to eliminate the offset voltage in order to bring ECG signals to the baseline voltage all the time. The training data consisted of four heartbeats from 14 patients and the length of data was 10,945 samples. We applied the four-fold cross validation to select the suitable parameter in each support vector regression model with 8,209 training and 2,736 testing samples.

We separated the training data into three different small parts (m = 1, 5, 10), the quantization level was two hundreds (q = 200) and the weight of cluster validation indices was 0.2.

As in the previous experiment, the RMSE of V2 ECG signal was reduced when we increased the number of small parts. The regression accuracy of the proposed method converged to the original accuracy result as in Table 4.7 and Figure 4.14. The reduction ratio when splitting the data set into ten parts was reach to 80% which higher than the real world data set 2 experiment. The reduction ratio of this experiment is better than the experiment from real world data set 2 because the output of ECG signals training data set has more close and redundant data as a periodic signal. The sorting and quantizing process were a good process to cluster the redundancy data or the periodic data.

Table 4.7 V2 ECG signal reconstructed result.

Number of small parts	Number of training samples	Number of testing samples	RMSE	Data reduction rate (%)
1	8,209	2,736	0.1035	95.38
5	8,209	2,736	0.0831	87.19
10	8,209	2,736	0.0717	80.93
Original	8,209	2,736	0.0683	0.00

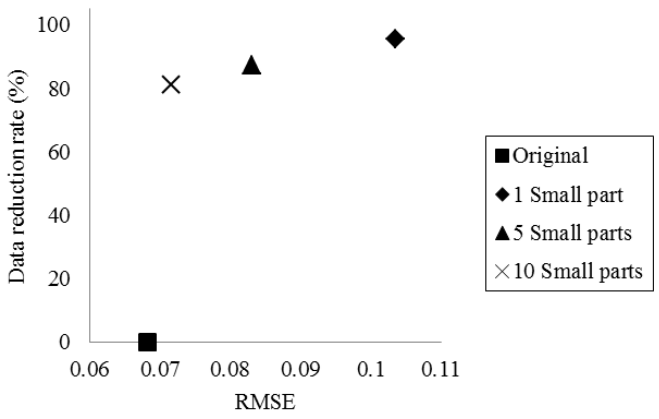


Figure 4.14 V2 ECG signal reconstructed result.

The regression result of V2 ECG signal is presented in Figure 4.15. The shape of the reconstructed signal was similar to the original data.

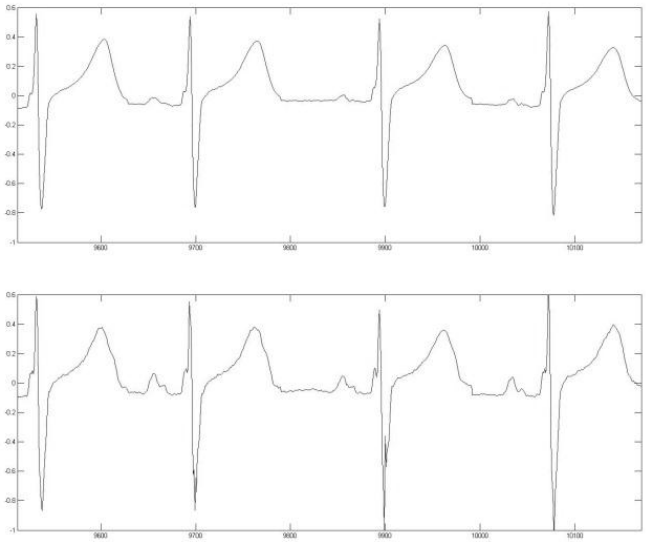


Figure 4.15 (Top) Original V2 chest lead signal,
(Bottom) Reconstructed V2 chest lead signal.

From the regression result of V3 ECG signal, we found that the reduction ratios of the split data into five parts and ten parts were similar because the properties of V3 ECG signal that need to split the data just five parts to get an accuracy nearly close to the original data as shown in Table 4.8 and Figure 4.16. The regression result of V3 ECG signal is presented in Figure 4.17.

Table 4.8 V3 ECG signal reconstructed result.

Number of small parts	Number of training samples	Number of testing samples	RMSE	Data reduction rate (%)
1	8,209	2,736	0.1762	96.59
5	8,209	2,736	0.1132	81.00
10	8,209	2,736	0.1108	80.80
Original	8,209	2,736	0.0940	0.00

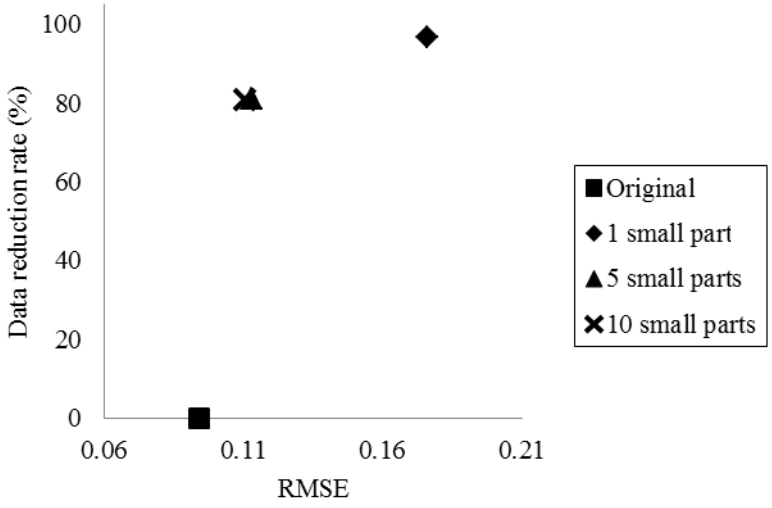


Figure 4.16 V3 ECG signal reconstructed result.

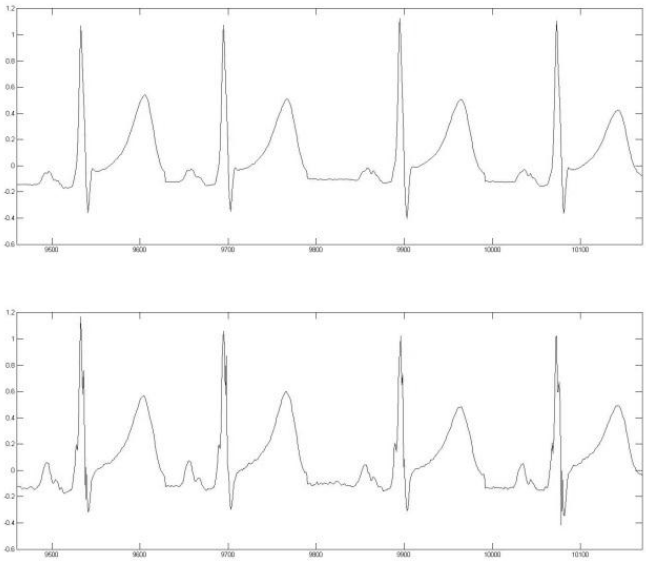


Figure 4.17 (Top) Original V3 chest lead signal,
(Bottom) Reconstructed V3 chest lead signal.

From the regression result of V4 ECG signal, the RMSE of the split data into ten parts was lower than the RMSE from the original data result because the data in 4-fold cross validation was randomly selected data. In this case, the split data into ten parts can form the regression model that gave a better RMSE than the original data set as shown in Table 4.9 and Figure 4.18. The regression result of V4 ECG signal is presented in Figure 4.19.

Table 4.9 V4 ECG signal reconstructed result.

Number of small parts	Number of training samples	Number of testing samples	RMSE	Data reduction rate (%)
1	8,209	2,736	0.1244	96.29
5	8,209	2,736	0.1116	89.14
10	8,209	2,736	0.0960	81.95
Original	8,209	2,736	0.1029	0.00

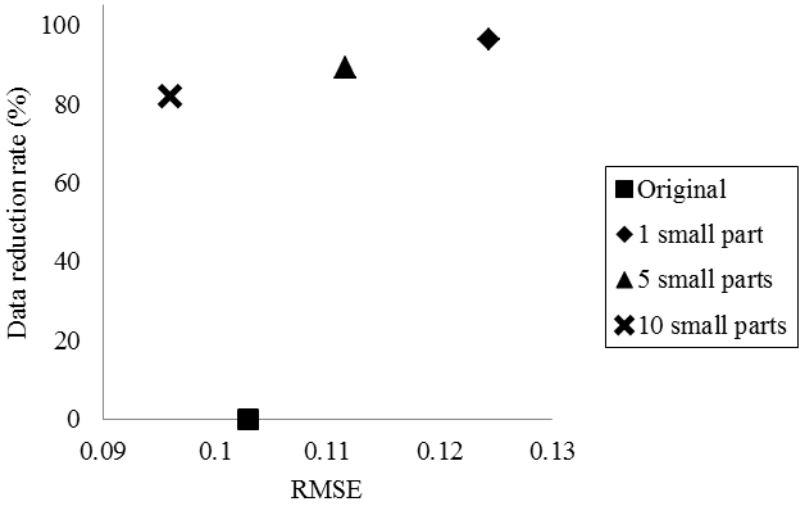


Figure 4.18 V4 ECG signal reconstructed result.

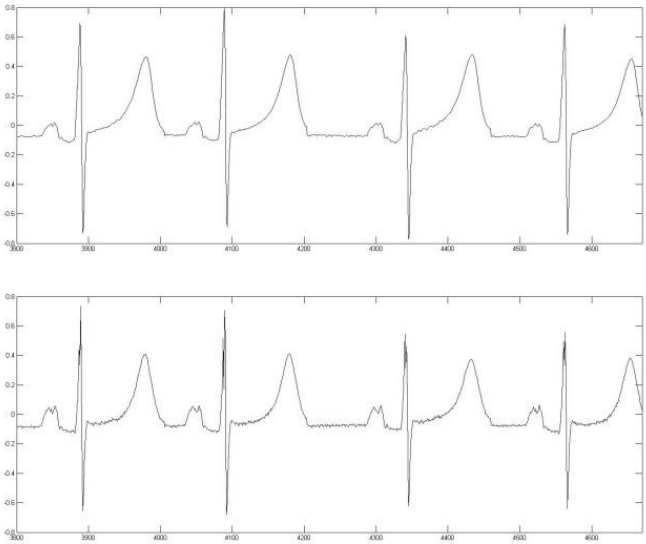


Figure 4.19 (Top) Original V4 chest lead signal,
(Bottom) Reconstructed V4 chest lead signal.

From Table 4.10 and Figure 4.20, the RMSE of the split data into five parts was better than the RMSE of the split data into ten parts due ot the same reason as the regression result of V4 ECG signal, i.e. the data in 4-fold cross validation was randomly selected data. The regression result of V5 ECG signal is presented in Figure 4.21.

Table 4.10 V5 ECG signal reconstructed result.

Number of small parts	Number of training samples	Number of testing samples	RMSE	Data reduction rate (%)
1	8,209	2,736	0.1007	96.29
5	8,209	2,736	0.0735	89.48
10	8,209	2,736	0.0750	83.57
Original	8,209	2,736	0.0613	0.00

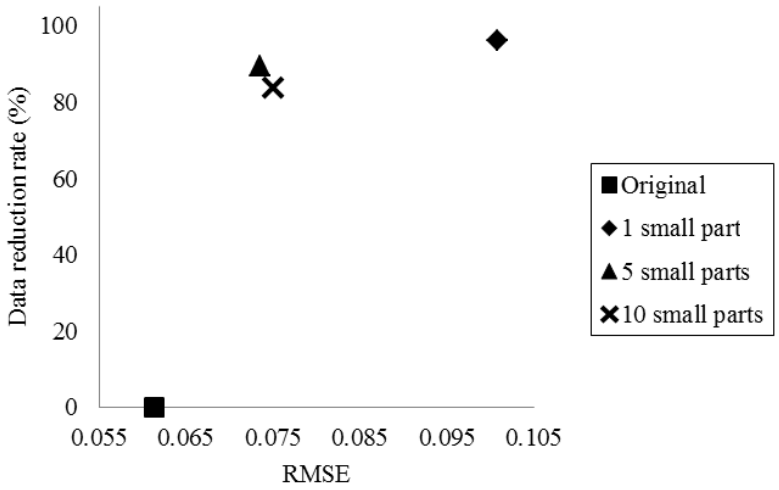


Figure 4.20 V5 ECG signal reconstructed result.

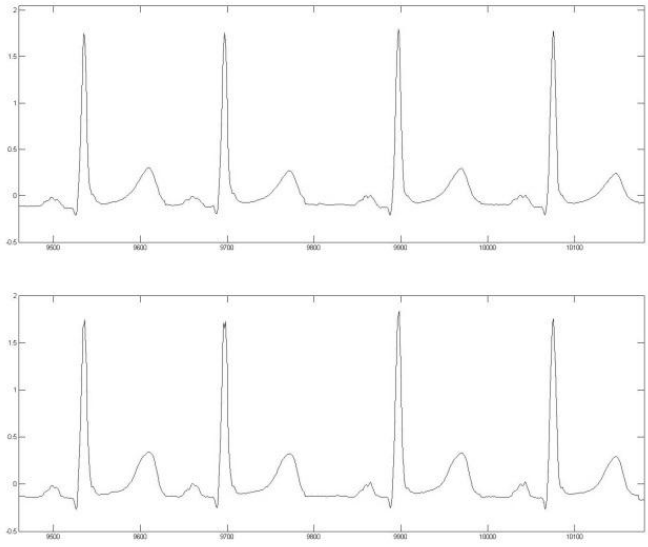


Figure 4.21 (Top) Original V5 chest lead signal,
(Bottom) Reconstructed V5 chest lead signal.

4.1.6 Concerning issues of the proposed method for regression problems

1) The proposed method cannot reduce the data if the outputs of the data are not redundant or close to each other.

2) The reduction ratio is low if the input data in each quantization levels are not close to each other.

3) If the input data in each quantization level are spread, the reduction ratio is lower.

4) The reduction ratio depends on the properties of training data set. The more data redundancy, the more reduction ratio becomes.

4.1.7 Setting the number of small parts and quantization levels

The computational complexity of the proposed method is $O((\ell/m)^3)$. Therefore, the higher m reduces the computing time and the accuracy of regression result reach to the original data set but the reduction ratio is decreasing. Besides, the number of quantization levels should be selected by considering the value and the range of the output data. The round-off error of the output data is also to be considered because after achieving input data clustering, the output data in each quantization level will be approximated by the mean. Therefore, before selecting parameter q, we should specify an acceptable error.

4.2 Result of instance reduction of classification problem

In this section, the performance of the instance reduction for classification with synthesized data and the public data set were evaluated. The synthesized data was used to approve the reduction method and the public data set was used for performance evaluation.

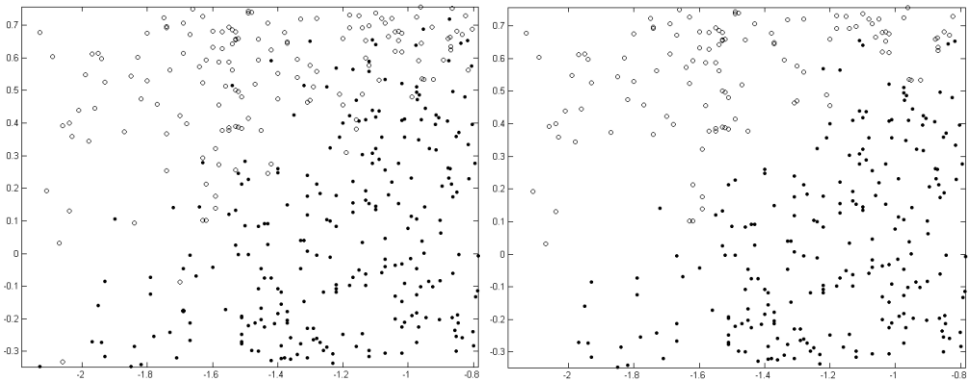
4.2.1 Two-dimensional problem: two-class synthesized data set

In the first experiment, a two-dimensional given two-class synthesized data set was used for presenting the procedure of the proposed method. The visualization of the 2D output results proved that the proposed method maintained the sample points near the decision boundary and reduced the

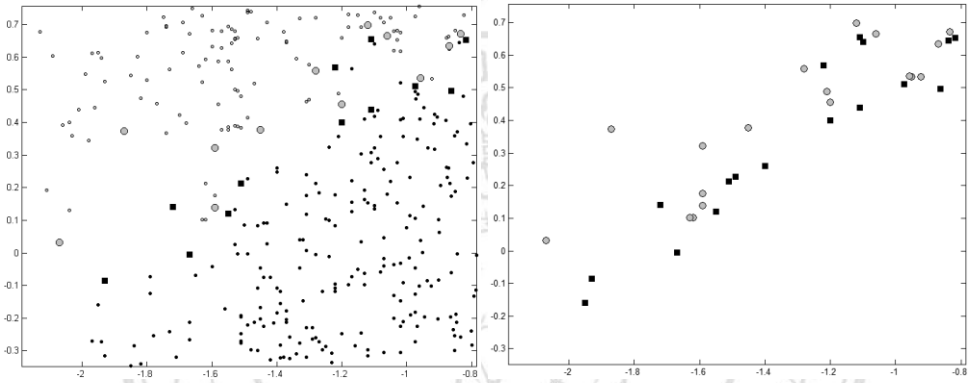
redundant sample points that were far from the decision boundary. The result of each procedure is shown in Figure 4.22 (a) – (e).

From the original data in Figure 4.22 (a), we found that the synthesized data present overlapping area between two classes that may cause by noises or real data so that make the difficulty to define the data boundaries. After noises and overlapping data were removed using 1NN clustering, the data boundaries of two classes are obviously observed in Figure 4.22 (b). Then, the reduced data set ($\mathcal{RS}$) was initialized and represented by black squares and big white dots, see Figure 4.22 (c). However, due to the limitation of 1NN, some overlapping areas still remain. After extract data procedure, the number of data points near the decision boundary increases as shown in Figure 4.22 (d). The data pruning was performed in order to remove unnecessary data points. The outcome of pruning step was the necessary data points which were used for classification system training (Figure 4.22 (e)).

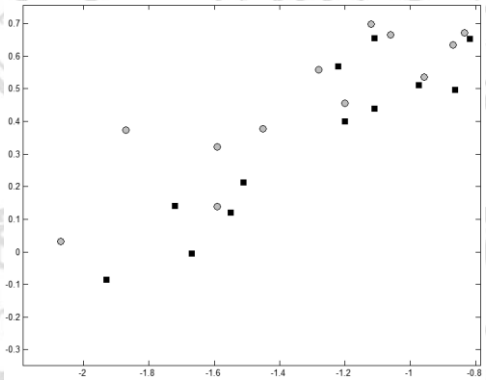
Figure 4.23 presented the voronoi diagram of original data set and reduced data set. The overlapping area in the orignal data was removed and the boundary of these classes was clearly separated. Therefore, the proposed method can reduce the classification data set and maintain the important data at the boundary of each class.



(a) (b)



(c) (d)



(e)

Figure 4.22 Scatter plots of two-dimensional data for two class problem. (a) The original data, (b) Preprocessed with removal of noise and overlap data, (c) Initialized the reduced data set ($\mathcal{RS}$), (d) Extracted the instance data near the decision boundary, (e) The results from pruning the $\mathcal{RS}$.

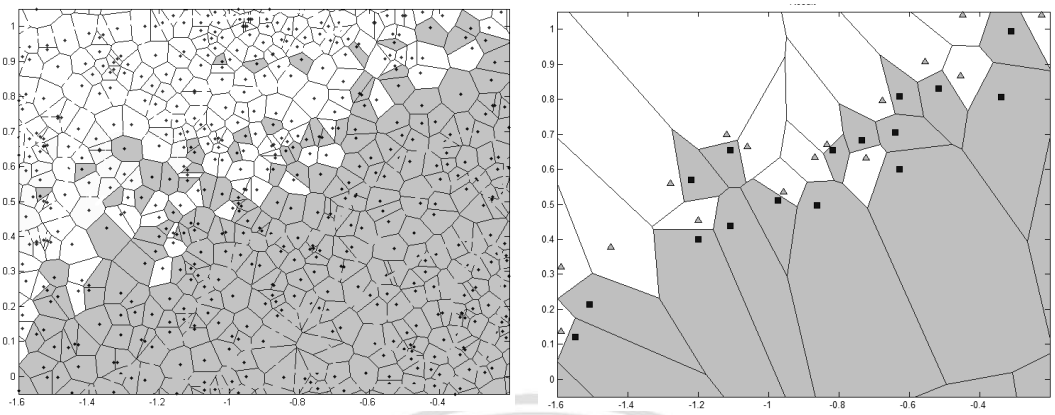
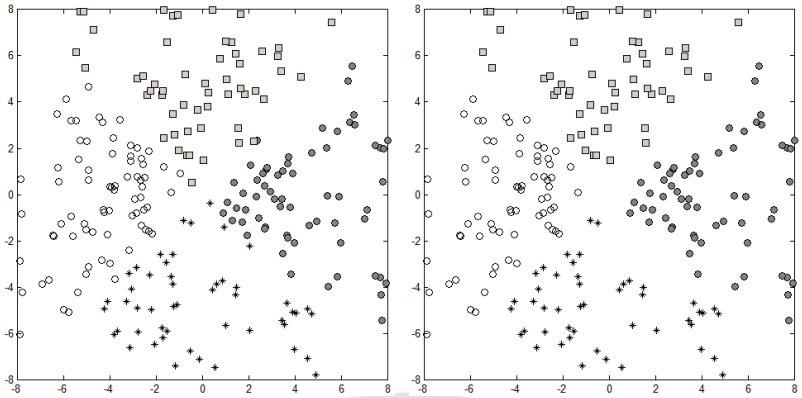


Figure 4.23 Voronoi diagram of the original data set (Left),
Voronoi diagram of the reduced data set (Right).

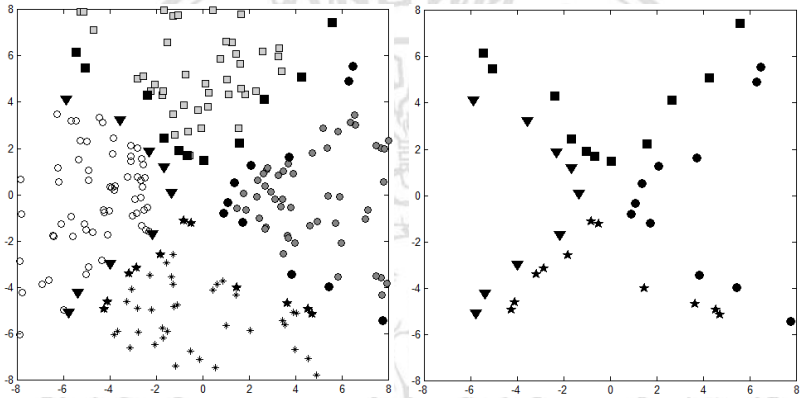
4.2.2 Two-dimensional problem: multiclass synthesized data set

The experimental results of different procedures are shown in Figure 4.24. The original data consisted of 4-class data sets which were represented by white circle, gray square, black circle, and star, respectively, as in Figure 4.24 (a). After the operation with 1NN classification, some noises and the overlapping data were removed (Figure 4.24 (b)). The result from extracted samples in Figure 4.24 (d) is similar to initialized $\mathcal{RS}$ in Figure 4.24 (c). Because the synthesized data are well-separated, the data from initialized $\mathcal{RS}$ is sufficient to be the boundary data points. At last, the pruning step was applied to remove similar data points and maintained essential data points according to Figure 4.24 (e).

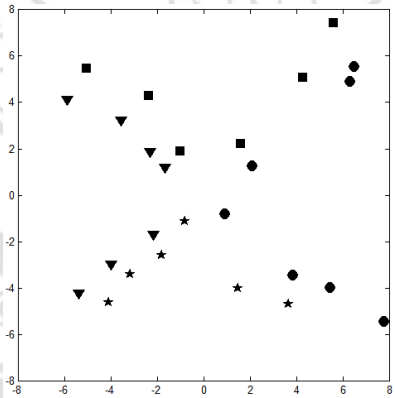
The voronoi diagram of the reduced data set was similar to the voronoi diagram of the original data set as shown in Figure 4.25. This experimental result approved that the proposed method could be applied to multiclass data sets.



(a) (b)



(c) (d)



(e)

Figure 4.24 Scatter plots of two-dimensional data for 4-class problem. (a) The original data, (b) Preprocessed with removal of noise and overlapping data, (c) Initialized the reduced data set ($\mathcal{RS}$), (d) Extracted the instance data near the decision boundary, (e) The results from pruning the $\mathcal{RS}$.

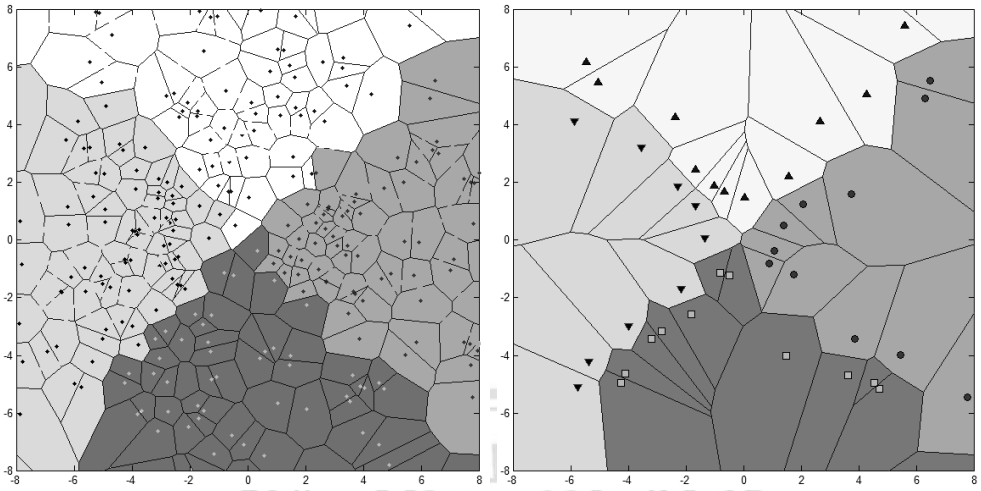


Figure 4.25 Voronoi diagram of the original data set : four-classes (Left), Voronoi diagram of the reduced data set : four-classes (Right).

4.2.3 Public data sets from sci2s.ugr.es : small data sets

We compared accuracy which was measured by 1NN as well as the data reduction ratio of our proposed method and the proposed method in [31]. The other prototype selection methods that were used to compare the performance with the proposed method are shown in Table 4.11.

In addition, the product of classification accuracy and reduction ratio parameter, the multi-criteria result from TOPSIS and the percentage improvement were also used in the performance comparison. All parameter values were received from the average of 10-fold cross validation.

In Table 4.12, we categorized the prototype selection method into three types (e.g., condensation method, edited method, and hybrid method followed by the type of selection). Each type can be separated into three subgroups (e.g., filter, batch, and wrapper followed by the evaluation of search). According to the procedure of the proposed method in Chapter 3, we can assign the proposed method in Hybrid–filter group as shown in Table 4.12.

Table 4.11 List of prototype selection methods.

Complete name	Abbreviate Name
All-*k* NN	AllKNN [32]
Class Conditional Instance Selection	CCIS [33]
CHC Evolutionary Algorithm	CHC [34]
Condensed Nearest Neighbor	CNN [35]
Cooperative Coevolutionary Instance Selection	CoCoIS [36]
C-Pruner	Cpruner [37]
Decremental Reduction Optimization Procedure 3	DROP3 [21]
Edited Nearest Neighbor	ENN [38]
Edited Nearest Neighbor Estimating Class Probabilistic and Threshold	ENNTh [39]
Edited Normalized Radial Basis Function	ENRBF [40]
Explore	Explore
Fast Condensed Nearest Neighbor	FCNN [41]
Generalized Condensed Nearest Neighbor	GCNN [42]
Generational Genetic Algorithm	GGA [43,44]
Hit Miss Network Edition Iterative	HMNEI [45]
Instance Based 3	IB3 [46]
Iterative Case Filtering	ICF [47]
Intelligent Genetic Algorithm	IGA [48]
Improved KNN	IKNN [49]
Modified Condensed Nearest Neighbor	MCNN [50]
Minimal Consistent Set	MCS [51]
Modified Edited Nearest Neighbor	MENN [52]
Mutual Neighborhood Value	MNV [53]
Model Class Selection	MoCS [54]
Modified Selective Subset	MSS [55]
Multiedit	Multiedit [56]
Nearest Centroid Neighbor Edition	NCNEdit [57]
Noise Removing based on Minimal Consistent Set	NRMCS [58]
Patterns by Ordered Projections	POP [59]
Prototype Selection Based on Clustering	PSC [15]
Prototype Selection using Relative Certainty Gain	PSRCG [60]
Reconsistent	Reconsistent [61]
Random Mutation Hill Climbing	RMHC [62]
Relative Neighborhood Graph Editing	RNG [63]
Reduced Nearest Neighbor	RNN [64]
Shrink	Shrink [65]
Selective Nearest Neighbor	SNN [66]
Steady-State Memetic Algorithm	SSMA [67]
Support Vector based Prototype Selection	SVBPS [68]
Tomek Condensed Nearest Neighbor	TCNN [69]
Template Reduction for KNN	TRKNN [70]
Variable Similarity Metric	VSM [71]

Table 4.12 Groups of prototype selection methods.

PS Method					
Condensation		Edition		Hybrid	
Filter	Wrapper	Filter	Wrapper	Filter	Wrapper
CNN	-	ENN	-	CCIS	Explore
GCNN		ENRBF		Cpruner	GGA
MCS		ENNTh		DROP3	IGA
MSS		MENN		HMNEI	CHC
MNV		Multiedit		IB3	SSMA
MCNN		NCNEdit		ICF	CoCoIS
PSC		RNGE		NRMCS	RMHC
RNN		AllKNN		PSRCG	
SNN		MoCS		SVBPS	
Shrink				VSM	
TCNN				**Proposed**	
IKNN					
POP					
Reconsistent					
TRKNN					

We divided the data used in this research into 3 types: small, medium, and large in order to enable the comparison between the results presented in [22] and ours. The first group of data sets used to evaluate the performance is the small data sets. These data sets contained 24 data sets, each of which has the length less than 2,000 samples as shown in Table 4.13. The table also presents the detail of the data, i.e., the name, the number of instances, the number of features, and the number of classes. The variation of features and classes can evaluate the generalization of the proposed method.

Table 4.13 List of small data sets.

Data sets	Number of instances	Number of features	Number of classes
appendicitis	106	7	2
bupa	345	6	2
cleveland	303	13	5
contraceptive	1,473	9	3
dermatology	366	34	6
ecoli	336	7	8
glass	214	9	7
haberman	306	3	2
hayes-roth	160	4	3
heart	270	13	2
iris	150	4	3
led7digit	500	7	10
monks	432	6	2
newthyroid	215	5	3
pima	768	8	2
sonar	208	60	2
spectfheart	267	44	2
tae	151	5	3
vehicle	846	18	4
vowel	990	13	11
wine	178	13	3
wisconsin	699	9	2
yeast	1,484	8	10
zoo	101	16	7

The performances of all prototype selection methods from Table 4.12 are presented in Table 4.14, including the proposed method with the average of accuracy, reduction ratio, the product of accuracy and reduction ratio, and the result from TOPSIS. The result from the methods that is based on hybrid-wrapper methods was better than the result from condensation and edition methods because the performance of the reduced data set from condensation and edition methods cannot be better than the original data. The hybrid-wrapper method applied the optimization algorithm to select the important prototype and reduce the unnecessary data so that the performance of this method is better than using the original data. Thus, we consider only the prototype selection methods that are based on condensation and edition method (omit the hybrid-wrapper method) as implemented by our proposed method. The result of our proposed method

using only condensation and edition compared to other methods is shown in Table 4.15.

Table 4.14 Result of small data sets from different methods.

Accuracy		Reduction ratio		Accuracy× Reduction ratio		TOPSIS	
CHC	0.7541	Explore	0.9741	CHC	0.7286	CHC	0.8316
SSMA	0.7516	CHC	0.9662	Explore	0.7189	SSMA	0.8272
GGA	0.7505	NRMCS	0.9637	SSMA	0.7149	GGA	0.8234
ENN	0.7476	SSMA	0.9511	GGA	0.7028	Explore	0.8228
RNG	0.7444	GGA	0.9365	RMHC	0.6692	RMHC	0.8089
NCNEdit	0.7424	CPruner	0.9201	RNN	0.6588	RNN	0.7993
RMHC	0.7421	RNN	0.9143	NRMCS	0.6400	IGA	0.7775
AllKNN	0.7411	IGA	0.9080	IGA	0.6241	MCNN	0.7771
ModelCS	0.7396	RMHC	0.9018	MCNN	0.6207	NRMCS	0.7760
Explore	0.7380	MCNN	0.8971	**Proposed**	**0.6027**	**Proposed**	**0.7723**
ENNTh	0.7330	CCIS	0.8949	CoCoIS	0.5953	CoCoIS	0.7686
HMNEI	0.7277	**Proposed**	**0.8449**	CPruner	0.5852	DROP3	0.7520
POP	0.7249	CoCoIS	0.8327	CCIS	0.5820	CCIS	0.7518
MENN	0.7234	DROP3	0.8207	DROP3	0.5697	CPruner	0.7501
RNN	0.7205	SNN	0.7501	IB3	0.4832	IB3	0.6951
CoCoIS	0.7150	ICF	0.7118	ICF	0.4683	ICF	0.6842
Proposed	**0.7133**	IB3	0.6965	PSC	0.4552	PSC	0.6746
MSS	0.7125	PSC	0.6917	TCNN	0.4535	TCNN	0.6734
Multiedit	0.7086	SVBPS	0.6742	FCNN	0.4498	FCNN	0.6706
TCNN	0.7032	Shrink	0.6551	SNN	0.4482	SNN	0.6693
FCNN	0.7029	TCNN	0.6449	SVBPS	0.4371	SVBPS	0.6611
CNN	0.7011	FCNN	0.6399	MNV	0.4165	MNV	0.6457
MCS	0.6990	MNV	0.5998	HMNEI	0.4017	HMNEI	0.6353
MNV	0.6944	CNN	0.5704	CNN	0.3999	CNN	0.6333
DROP3	0.6942	Reconsistent	0.5623	MCS	0.3909	MCS	0.6265
IB3	0.6937	MCS	0.5593	Reconsistent	0.3812	Reconsistent	0.6184
MCNN	0.6918	HMNEI	0.5520	MENN	0.3539	MENN	0.6005
IKNN	0.6914	VSM	0.5340	PSRCG	0.3433	ENNTh	0.5937
IGA	0.6873	PSRCG	0.5118	ENNTh	0.3432	TRKNN	0.5893
TRKNN	0.6800	TRKNN	0.5043	VSM	0.3430	PSRCG	0.5890
Reconsistent	0.6779	MENN	0.4892	TRKNN	0.3429	VSM	0.5871
GCNN	0.6751	ENNTh	0.4681	MSS	0.3300	MSS	0.5826
PSRCG	0.6709	GCNN	0.4668	GCNN	0.3151	GCNN	0.5679
NRMCS	0.6641	MSS	0.4632	Shrink	0.3062	Shrink	0.5591
PSC	0.6580	IKNN	0.3696	AllKNN	0.2677	AllKNN	0.5446
ICF	0.6579	Multiedit	0.3649	Multiedit	0.2586	Multiedit	0.5329
ENRBF	0.6504	AllKNN	0.3612	IKNN	0.2555	IKNN	0.5276
CCIS	0.6503	ENRBF	0.3394	ENRBF	0.2208	ENN	0.5044
SVBPS	0.6483	ENN	0.2633	ENN	0.1968	RNG	0.4961
VSM	0.6423	RNG	0.2458	RNG	0.1830	ENRBF	0.4953
CPruner	0.6360	NCNEdit	0.2239	NCNEdit	0.1662	NCNEdit	0.4867
SNN	0.5975	ModelCS	0.1296	ModelCS	0.0959	ModelCS	0.4525
Shrink	0.4674	POP	0.0967	POP	0.0701	POP	0.4365

Table 4.15 presents the result of the proposed methods is little less than the result obtained from the best method around 0.05 for the product of accuracy and reduction ratio and 0.03 for TOPSIS.

Table 4.15 Result of small data sets from different methods without hybrid-wrapper algorithm.

Accuracy		Reduction		Accuracy× Reduction ratio		TOPSIS	
ENN	0.7476	NRMCS	0.9637	RNN	0.6588	RNN	0.7993
RNG	0.7444	CPruner	0.9201	NRMCS	0.6400	MCNN	0.7771
NCNEdit	0.7424	RNN	0.9143	MCNN	0.6207	NRMCS	0.7760
AllKNN	0.7411	MCNN	0.8971	**Proposed**	**0.6027**	**Proposed**	**0.7723**
ModelCS	0.7396	CCIS	0.8949	CPruner	0.5852	DROP3	0.7520
ENNTh	0.7330	**Proposed**	**0.8449**	CCIS	0.5820	CCIS	0.7461
HMNEI	0.7277	DROP3	0.8207	DROP3	0.5697	Cpruner	0.7378
POP	0.7249	SNN	0.7501	IB3	0.4832	IB3	0.6951
MENN	0.7234	ICF	0.7118	ICF	0.4683	ICF	0.6843
RNN	0.7205	IB3	0.6965	PSC	0.4552	PSC	0.6746
Proposed	**0.7133**	PSC	0.6917	TCNN	0.4535	TCNN	0.6734
MSS	0.7125	SVBPS	0.6742	FCNN	0.4498	FCNN	0.6706
Multiedit	0.7086	Shrink	0.6551	SNN	0.4482	SNN	0.6693
TCNN	0.7032	TCNN	0.6449	SVBPS	0.4371	SVBPS	0.6611
FCNN	0.7029	FCNN	0.6399	MNV	0.4165	MNV	0.6457
CNN	0.7011	MNV	0.5998	HMNEI	0.4017	HMNEI	0.6353
MCS	0.6990	CNN	0.5704	CNN	0.3999	CNN	0.6333
MNV	0.6944	Reconsistent	0.5623	MCS	0.3909	MCS	0.6265
DROP3	0.6942	MCS	0.5593	Reconsistent	0.3812	Reconsistent	0.6184
IB3	0.6937	HMNEI	0.5520	MENN	0.3539	MENN	0.6006
MCNN	0.6918	VSM	0.5340	PSRCG	0.3433	ENNTh	0.5937
IKNN	0.6914	PSRCG	0.5118	ENNTh	0.3432	TRKNN	0.5893
TRKNN	0.6800	TRKNN	0.5043	VSM	0.3430	PSRCG	0.5890
Reconsistent	0.6779	MENN	0.4892	TRKNN	0.3429	VSM	0.5871
GCNN	0.6751	ENNTh	0.4681	MSS	0.3300	MSS	0.5826
PSRCG	0.6709	GCNN	0.4668	GCNN	0.3151	GCNN	0.5679
NRMCS	0.6641	MSS	0.4632	Shrink	0.3062	Shrink	0.5592
PSC	0.6580	IKNN	0.3696	AllKNN	0.2677	AllKNN	0.5447
ICF	0.6579	Multiedit	0.3649	Multiedit	0.2586	Multiedit	0.5329
ENRBF	0.6504	AllKNN	0.3612	IKNN	0.2555	IKNN	0.5276
CCIS	0.6503	ENRBF	0.3394	ENRBF	0.2208	ENN	0.5044
SVBPS	0.6483	ENN	0.2633	ENN	0.1968	RNG	0.4961
VSM	0.6423	RNG	0.2458	RNG	0.1830	ENRBF	0.4954
CPruner	0.6360	NCNEdit	0.2239	NCNEdit	0.1662	NCNEdit	0.4867
SNN	0.5975	ModelCS	0.1296	ModelCS	0.0959	ModelCS	0.4525
Shrink	0.4674	POP	0.0967	POP	0.0701	POP	0.4365

The percentage improvement (PI) applied to evaluate the performance between the proposed method and other prototype selection methods in Table 4.15. We report the PI of accuracy and reduction ratio in Table 4.16. The boldfaced characters note that the proposed method has a better result than those methods.

Table 4.16 Percentage improvement between the proposed method and other methods using small data sets.

Data sets	Accuracy	Reduction	PI (%)	
			Accuracy	Reduction
Proposed	0.7133	0.8449	-	-
ENN	0.7476	0.2633	-4.59	**220.88**
RNG	0.7444	0.2458	-4.18	**243.69**
NCNEdit	0.7424	0.2239	-3.92	**277.30**
AllKNN	0.7411	0.3612	-3.76	**133.91**
ModelCS	0.7396	0.1296	-3.56	**551.74**
ENNTh	0.733	0.4681	-2.70	**80.49**
HMNEI	0.7277	0.552	-1.98	**53.06**
POP	0.7249	0.0967	-1.60	**773.35**
MENN	0.7234	0.4892	-1.40	**72.70**
RNN	0.7205	0.9143	-1.01	-7.59
MSS	0.7125	0.4632	**0.11**	**82.39**
Multiedit	0.7086	0.3649	**0.66**	**131.52**
TCNN	0.7032	0.6449	**1.43**	**31.03**
FCNN	0.7029	0.6399	**1.47**	**32.05**
CNN	0.7011	0.5704	**1.74**	**48.12**
MCS	0.699	0.5593	**2.04**	**51.07**
MNV	0.6944	0.5998	**2.71**	**40.86**
DROP3	0.6942	0.8207	**2.75**	**2.95**
IB3	0.6937	0.6965	**2.81**	**21.31**
MCNN	0.6918	0.8971	**3.10**	-5.82
IKNN	0.6914	0.3696	**3.17**	**128.62**
TRKNN	0.68	0.5043	**4.90**	**67.54**
Reconsistent	0.6779	0.5623	**5.22**	**50.25**
GCNN	0.6751	0.4668	**5.66**	**81.01**
PSRCG	0.6709	0.5118	**6.32**	**65.09**
NRMCS	0.6641	0.9637	**7.41**	-12.33
CCIS	0.6628	0.8532	**7.61**	-0.97
PSC	0.658	0.6917	**8.39**	**22.15**
ICF	0.6579	0.7118	**8.41**	**18.70**
ENRBF	0.6504	0.3394	**9.67**	**148.93**
SVBPS	0.6483	0.6742	**10.01**	**25.33**
VSM	0.6423	0.534	**11.05**	**58.23**
Cpruner	0.6345	0.8784	**12.41**	-3.81
SNN	0.5975	0.7501	**19.37**	**12.65**
Shrink	0.4674	0.6551	**52.59**	**28.98**

We sort the methods by the PI of accuracy values. The PI of accuracy values in Table 4.16 of our proposed methods is lower than ENN, RNG, NCNEdit, AllKNN, ModelCS, ENNTh, HMNEI, POP, and MENN methods (the maximum is 4.59%). However, the PI of reduction is better than all of these methods. The MCNN and NRMCS methods gave a better reduction ratio result but the proposed method gave a better accuracy result. Even though, the RNN method showed both accuracy and reduction ratio being better than the proposed method 1 and 7% but it consumed too much cost of computation in both time and memory than the other methods. The proposed method provided a better performance in both accuracy and reduction ratio than 21 methods.

Figure 4.26 shows the scatter plot of the average of normalized accuracy and reduction ratio. We present the top ten TOPSIS results from Table 4.17. The position of the best method must be close to (1,1) and the worst method must be close to (0,0). In this case, the proposed method is the second from all of 36 methods. It is the second in accuracy performance and the fourth in reduction performance.

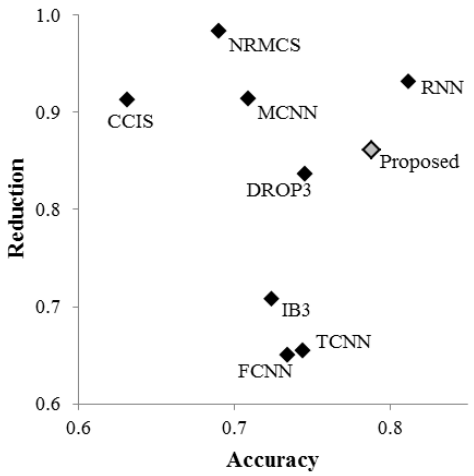


Figure 4.26 Average of normalized accuracy and reduction ratio of small data sets.

The accuracy of each data set is different in range, thus we want to present the position of the proposed method accuracy between the maximum and minimum of the other prototype selection method accuracy from each small data set. The mark '-' is the proposed method's accuracy and the vertical

line is the range between the maximum and minimum accuracy from other prototype selection methods as show in Figure 4.27. All of the accuracy results are nearly close to the maximum value except for Cleveland, heart, tae, and wine data sets that have accuracy lower than half of the range.

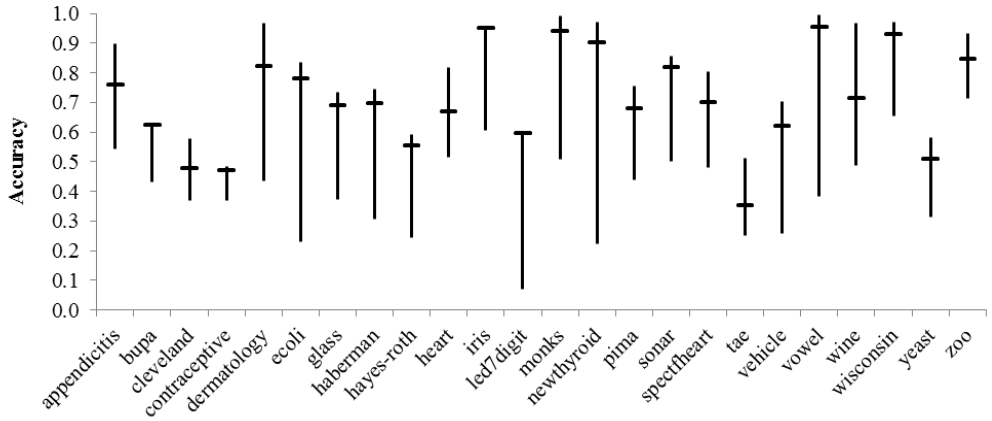


Figure 4.27 Range of average accuracy of each data in small data sets.

4.2.4 Public data sets from sci2s.ugr.es : medium data sets

Medium-sized data sets are a data set which contains sample points between 2,000 and 20,000 samples, the number of features between 2 to 85, and the number of classes from 2 to 11. The detail of the medium data sets is shown in Table 4.17.

Table 4.17 List of medium data sets.

Data sets	Number of instances	Number of features	Number of Classes
banana	5,300	2	2
coil2000	9,822	85	2
magic	19,020	10	2
marketing	8,993	13	9
page-blocks	5,472	10	5
pen based	10,992	16	10
phoneme	5,404	5	2
ring	7,400	20	2
sat image	6,435	36	7
segment	2,310	19	7
spam base	4,597	57	2
texture	5,500	40	11
thyroid	7,200	21	3
titanic	2,201	3	2
two norm	7,400	20	2

In this experiment, we did not select all prototype selection method from Table 4.12 but we used some of them that showed a good result from the small data set experiment to evaluate the performance with the proposed method. The experimental result is presented in Table 4.18.

Table 4.18 Result of medium data sets from different methods.

Accuracy		Reduction ratio		Accuracy× Reduction ratio		TOPSIS	
RMHC	0.8712	CHC	0.9949	SSMA	0.8547	SSMA	0.9075
SSMA	0.8670	MCNN	0.9909	CHC	0.8490	CHC	0.8994
RNG	0.8638	SSMA	0.9859	RNN	0.8124	RNN	0.8910
ModelCS	0.8612	CCIS	0.9580	RMHC	0.7842	RMHC	0.8849
HMNEI	0.8571	RNN	0.9560	GGA	0.7736	GGA	0.8769
CHC	0.8534	NRMCS	0.9550	**Proposed**	**0.7728**	**Proposed**	0.8744
GGA	0.8503	**Proposed**	**0.9201**	NRMCS	0.7693	NRMCS	0.8622
RNN	0.8498	GGA	0.9097	CPruner	0.7276	CPruner	0.8505
AllKNN	0.8483	DROP3	0.9075	DROP3	0.7206	DROP3	0.8423
POP	0.8433	RMHC	0.9001	MCNN	0.7168	CCIS	0.8224
MSS	0.8413	CPruner	0.8875	CCIS	0.7106	MCNN	0.8159
Proposed	**0.8399**	ICF	0.8399	IB3	0.6428	IB3	0.8000
IB3	0.8397	IB3	0.7655	FCNN	0.6325	FCNN	0.7939
FCNN	0.8300	FCNN	0.7620	TCNN	0.6176	TCNN	0.7847
CNN	0.8241	TCNN	0.7529	CNN	0.6152	CNN	0.7828
TCNN	0.8203	CNN	0.7466	ICF	0.5830	ICF	0.7594
MENN	0.8198	Reconsistent	0.6927	Reconsistent	0.5389	Reconsistent	0.7331
Cpruner	0.8198	MSS	0.6042	MSS	0.5083	MSS	0.7084
NREMCS	0.8055	HMNEI	0.5290	HMNEI	0.4534	HMNEI	0.6717
DROP3	0.7941	MENN	0.2880	MENN	0.2361	MENN	0.5419
Reconsistent	0.7779	AllKNN	0.1925	AllKNN	0.1633	AllKNN	0.5143
CCIS	0.7417	RNG	0.1263	RNG	0.1091	RNG	0.4968
MCNN	0.7234	POP	0.0984	POP	0.0830	POP	0.4813
ICF	0.6941	ModelCS	0.0644	ModelCS	0.0555	ModelCS	0.4773

Table 4.19 presents the experimental result except for the prototype selection method based on the hybrid–wrapper method. The proposed method had the average accuracy less than the best result from RNG method for about 2%, and the reduction ratio less than MCNN for about 7%, but the Accuracy×reduction ratio and the TOPSIS were better than these methods. The proposed method was the second from twenty methods when compared with TOPSIS.

Table 4.19 Result of medium data sets from different methods without hybrid-wrapper algorithm.

Accuracy		Reduction ratio		Accuracy× Reduction ratio		TOPSIS	
RNG	0.8638	MCNN	0.9909	RNN	0.8124	RNN	0.8910
ModelCS	0.8612	CCIS	0.9580	**Proposed**	**0.7728**	**Proposed**	**0.8744**
HMNEI	0.8571	RNN	0.9560	NRMCS	0.7693	NRMCS	0.8622
RNN	0.8498	NRMCS	0.9550	CPruner	0.7276	CPruner	0.8505
AllKNN	0.8483	**Proposed**	**0.9201**	DROP3	0.7206	DROP3	0.8423
POP	0.8433	DROP3	0.9075	MCNN	0.7168	CCIS	0.8224
MSS	0.8413	CPruner	0.8875	CCIS	0.7106	MCNN	0.8159
Proposed	**0.8399**	ICF	0.8399	IB3	0.6428	IB3	0.8000
IB3	0.8397	IB3	0.7655	FCNN	0.6325	FCNN	0.7939
FCNN	0.8300	FCNN	0.7620	TCNN	0.6176	TCNN	0.7847
CNN	0.8241	TCNN	0.7529	CNN	0.6152	CNN	0.7828
TCNN	0.8203	CNN	0.7466	ICF	0.5830	ICF	0.7594
MENN	0.8198	Reconsistent	0.6927	Reconsistent	0.5389	Reconsistent	0.7331
CPruner	0.8198	MSS	0.6042	MSS	0.5083	MSS	0.7084
NRMCS	0.8055	HMNEI	0.5290	HMNEI	0.4534	HMNEI	0.6717
DROP3	0.7941	MENN	0.2880	MENN	0.2361	MENN	0.5419
Reconsistent	0.7779	AllKNN	0.1925	AllKNN	0.1633	AllKNN	0.5143
CCIS	0.7417	RNG	0.1263	RNG	0.1091	RNG	0.4968
MCNN	0.7234	POP	0.0984	POP	0.0830	POP	0.4813
ICF	0.6941	ModelCS	0.0644	ModelCS	0.0555	ModelCS	0.4773

The ranges of PI were between -2.77% to 21% for the accuracy and the PI of reduction ratio were -7.14% to 1,328%. The RNN method was the only one that had a better value than the proposed method but the reduction time of RNN that reported in [22] was longest (The summary was about 400,000 second). The PI values are presented in Table 4.20.

Figure 4.28 shows the scatter plot of the average of normalized accuracy and reduction ratio. From this graph, the accuracy of the proposed method is the second. The reduction ratio of the proposed method is the third if we focus only on the method which had accuracy better than 0.7.

From Figure 4.29, all of the accuracy results from each data set are nearly close to the maximum value except for the magic, spambase, and twonorm data sets whose accuracy are lower than half of the range.

Table 4.20 Percentage improvement between the proposed method and other methods using medium data sets.

Data sets	Accuracy	Reduction	PI (%)	
			Accuracy	Reduction
Proposed	0.8399	0.9201	-	-
RNG	0.8638	0.1263	-2.77	**628.39**
ModelCS	0.8612	0.0644	-2.48	**1,328.87**
HMNEI	0.8571	0.5290	-2.01	**73.94**
RNN	0.8498	0.9560	-1.17	-3.75
AllKNN	0.8483	0.1925	-0.99	**377.87**
POP	0.8433	0.0984	-0.40	**834.74**
MSS	0.8413	0.6042	-0.17	**52.28**
IB3	0.8397	0.7655	**0.02**	**20.2**
FCNN	0.8300	0.7620	**1.19**	**20.75**
CNN	0.8241	0.7466	**1.92**	**23.24**
TCNN	0.8203	0.7529	**2.39**	**22.21**
MENN	0.8198	0.2880	**2.44**	**219.47**
CPruner	0.8198	0.8875	**2.45**	**3.67**
NRMCS	0.8055	0.9550	**4.26**	-3.66
DROP3	0.7941	0.9075	**5.77**	**1.39**
Reconsistent	0.7779	0.6927	**7.96**	**32.82**
CCIS	0.7417	0.9580	**13.23**	-3.96
MCNN	0.7234	0.9909	**16.10**	-7.14
ICF	0.6941	0.8399	**21.00**	**9.54**

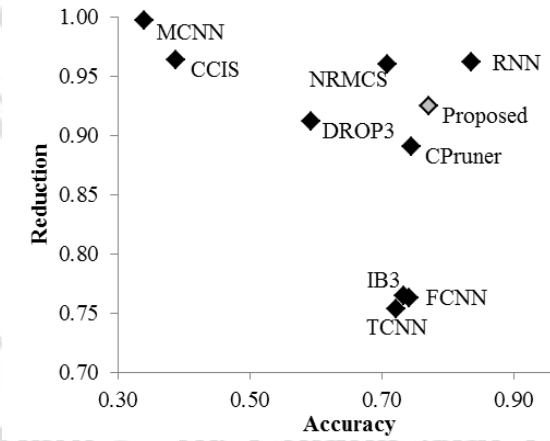


Figure 4.28 Average of normalized accuracy and reduction ratio of medium data set.

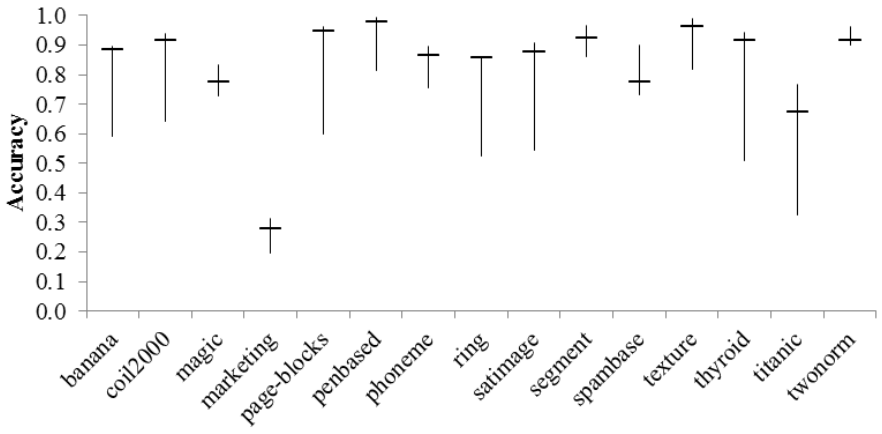


Figure 4.29 Range of average accuracy of each data in medium data sets.

4.2.5 Public data sets from sci2s.ugr.es : large data sets

A large-sized data set is the data set which has sample points between 40,000 and 300,000 samples, the number of features between 9 and 42, and the number of classes from 2 to 8. The details of large data sets are shown in Table 4.21.

Table 4.21 List of large data sets.

Data sets	Number of instances	Number of features	Number of Classes
Adult	48,842	14	2
census	299,285	41	2
connect-4	67,557	42	3
fars	100,968	29	8
shuttle	58,000	9	7

Due to the limitation of the computational resources as well as the experimental result from the proposed method being better than some prototype selection methods for small and medium data sets, we did not evaluate the performance with all the methods. The prototype selection methods that we were able to compare with the proposed method are CCIS, DROP3, FCNN, HMNEI, MCNN, RMHC, RNG, and SSMA, i.e. we compare the performance of the proposed method with all of these methods.

From Table 4.22, the proposed method could operate with the large data sets and it had average accuracy less than the best result from RNG method for

about 4%, and the reduction ratio less than MCNN for about 8%; however the Accuracy×reduction ratio and the TOPSIS had better than these methods. The best result is SSMA, which is based on hybrid–wrapper method. The experimental result from the proposed method is lower than the result from SSMA for about 0.09 and 0.04.

Table 4.22 Result of large data sets from different methods.

Accuracy		Reduction		Accuracy×Reduction		TOPSIS	
RNG	0.8421	MCNN	0.9992	SSMA	0.8296	SSMA	0.8899
SSMA	0.8401	SSMA	0.9875	RMHC	0.7527	RMHC	0.8649
RMHC	0.8358	CCIS	0.9225	CCIS	0.7491	CCIS	0.8580
HMNEI	0.8187	**Proposed**	**0.9156**	**Proposed**	**0.7358**	**Proposed**	**0.8508**
CCIS	0.8120	RMHC	0.9006	DROP3	0.7242	DROP3	0.8475
DROP3	0.8113	DROP3	0.8926	MCNN	0.6396	MCNN	0.7673
Proposed	**0.8037**	FCNN	0.7178	FCNN	0.5683	FCNN	0.7529
FCNN	0.7918	HMNEI	0.6489	HMNEI	0.5313	HMNEI	0.7256
MCNN	0.6401	RNG	0.1871	RNG	0.1575	RNG	0.5102

The PI of accuracy has a range between -4.6% and 25%, and the PI of reduction ratio are from -8.4% to 389%. The SSMA and CCIS methods are the only two methods that have better PI values than the proposed method. The PI values are presented in Table 4.23.

Table 4.23 Percentage improvement between the proposed method and other methods using large data sets.

Data sets	Accuracy	Reduction	PI (%)	
			Accuracy	Reduction
Proposed	0.8037	0.9156	-	-
RNG	0.8421	0.1871	-4.57	**389.45**
SSMA	0.8401	0.9875	-4.33	-7.28
RMHC	0.8358	0.9006	-3.85	**1.66**
HMNEI	0.8187	0.6489	-1.83	**41.09**
CCIS	0.8120	0.9225	-1.02	-0.76
DROP3	0.8113	0.8926	-0.94	**2.57**
FCNN	0.7918	0.7178	**1.5**	**27.56**
MCNN	0.6401	0.9992	**25.55**	-8.37

Due to the limitation of the cost of computation, the data set used to evaluate the performance was not enough to indicate the robustness of the proposed and others methods. Then, the performance was not similar to the experimental result from small and medium data sets. However, the

performance of the proposed method is still in the group that gave the best result as shown in Figure 4.30. Finally, the accuracy of three large data sets was close to the maximum accuracy (Figure 4.31).

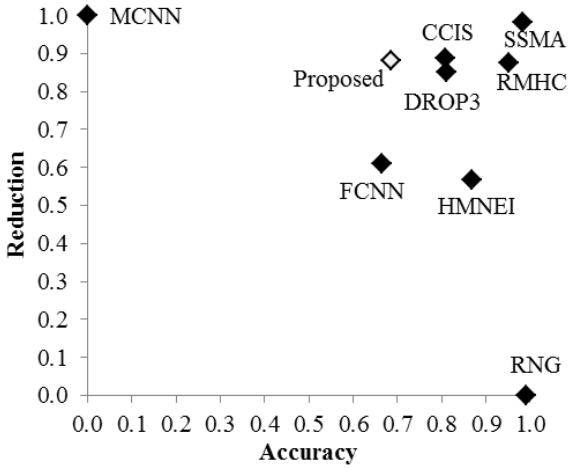


Figure 4.30 Average of normalized accuracy and reduction ratio of large data set.

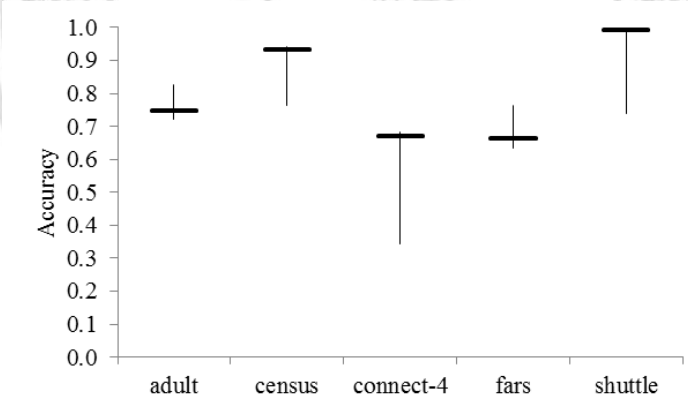


Figure 4.31 Range of average accuracy of each data in large data sets.

4.2.6 Evaluation of reduced data sets in model-based classifiers

In this section, the comparison of the average accuracy between 3 super vised learning of 10-fold cross validation classified by supervised learning algorithms, i.e., SVM, ANN, and 1-NN classifications, is shown in Table 4.24. The real-world data sets used in this experiment were obtained from sci2s.ugr.es. The data set consists of two classes with various features between 2 are 85, and the number of sample points was between 106 and 19,020 samples. From the average accuracy result, we found that 1NN clustering provided the highest accuracy because the proposed method applied 1NN to process the training data set. However, the average accuracy of ANN and SVM were also close to 1NN so the proposed method could, therefore, provide the reduced data sets to the other supervised learning algorithms.

Table 4.24 Testing accuracy result from supervised learning and 1NN rule.

Data sets	SVM	ANN	1NN
coil2000	**0.9394**	0.8916	0.921
Haberman	**0.7320**	0.6439	0.700
Monks	**0.9773**	0.9727	0.9416
Spectfheart	**0.7905**	0.7001	0.7019
Banana	0.8874	**0.8966**	0.8877
Bupa	0.5789	**0.6408**	0.6272
Heart	0.563	**0.7185**	0.6704
Magic	0.6496	**0.8503**	0.7772
Spambase	0.6419	**0.8908**	0.7775
Twonorm	0.7258	**0.9641**	0.9211
Appendicitis	0.7273	0.7464	**0.7625**
Phoneme	0.8294	0.7948	**0.8677**
Pima	0.6511	0.6627	**0.6819**
Ring	0.4951	0.7766	**0.8593**
Sonar	0.7876	0.7362	**0.8203**
Titanic	0.5670	0.5403	**0.6770**
Wisconsin	0.7313	0.8952	**0.9311**
Average	0.7220	0.7836	**0.7956**

4.2.7 Concerning issues of the proposed method for classification problems

1) The proposed method has low data reduction efficiency when the data set has only boundary data.

2) The values in the data set must be numerical or reasonably labeled because the propose method reduces data based on the distance.

3) The computational complexity of the proposed method is $o(3\ell^2 + c\ell^3)$ where c is the ratio of the sample points located in the class to the total number of sample points in each class boundary with $0 < c \leq 1$. The proposed method will process even faster if the number of sample points located in the class boundary is smaller.