

MULTI-TASK ENSEMBLE CREATION FOR ADVANCING PERFORMANCE OF IMAGE SEGMENTATION

HAN LIU¹, SHYI-MING CHEN²

¹ School of Computer Science and Informatics, Cardiff University, Queen's Buildings, 5 The Parade, Cardiff CF24 3AA, United Kingdom

²Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan

E-MAIL: liuh48@cardiff.ac.uk, smchen@mail.ntust.edu.tw

Abstract:

Image classification is a special type of applied machine learning tasks, where each image can be treated as an instance if there is only one target object that belongs to a specific class and needs to be recognized from an image. In the case of recognizing multiple target objects from an image, the image classification task can be formulated as image segmentation, leading to multiple instances being extracted from an image. In the setting of machine learning, each instance newly extracted from an image belongs to a specific class (a special type of target objects to be recognized) and presents specific features. In this context, in order to achieve effective recognition of each target object, it is crucial to undertake effective selection of features relevant to each specific class and appropriate setting of the training of classifiers on the selected features. In this paper, a multi-task approach of ensemble creation is proposed. The proposed approach is designed to first adopt multiple methods of multi-task feature selection for obtaining multiple groups of feature subsets (i.e., multiple subsets of features selected for each class), then to employ the C4.5 algorithm or the KNN algorithm to create an ensemble of classifiers using each group of feature subsets resulting from a specific one of the multi-task feature selection methods, and finally all the ensembles are fused to classify each instance. We compare the performance obtained using our proposed way of ensemble creation with the one obtained using classifiers trained on different feature sets prepared through various ways. The experimental results show some advances achieved in the overall classification performance through using our proposed ensemble creation approach, in comparison with the use of existing feature selection methods and learning algorithms.

Keywords:

Ensemble learning; Multi-task ensemble creation; Diversity creation; Feature selection; Image segmentation

1. Introduction

Image classification is a popular application of machine learning. A special type of image classification is referred to

as image segmentation, where an image involves multiple target objects to be detected. In the context of image segmentation through machine learning, each target object is treated as an instance, which belongs to a specific class.

In order to achieve effective recognition of a target object, it is necessary to make sure that features highly relevant to each specific class are selected for training classifiers. In general, feature selection can be adopted to achieve removal of redundant or irrelevant features for improving the classification performance, but the use of a traditional (single-task) feature selection method cannot guarantee that the performance for each class is improved, since the subset of features selected in a single-task manner may be relevant for some but not all of the classes [1].

This has motivated the development of multi-task feature selection methods (MTFS) [1], i.e., a feature selection task per class. However, the same method of MTFS would usually show different suitability for different data sets [2], even if these data sets are produced by transforming the same original n -class data set into n different binary data sets. In this setting, while the same algorithm is used for training of n binary classifiers on the n binary data sets, the binary classifiers could show varied performance for different classes. In order to address the above issues, we propose a multi-task ensemble creation approach in this paper, through employing different methods of MTFS, leading to multiple subsets of selected features being produced for each class, in order for the same learning algorithm to create multiple ensembles of binary classifiers.

The organization of the rest of this paper is shown as follows. Section 2 provides a review of feature selection techniques. In Section 3, we describe the procedure of our proposed approach of multi-task ensemble creation (MTEC). In Section 4, we describe the details of our experimental setting and discuss the results. The conclusions and further directions are given in Section 5.

2. Related work

In general, the procedure of feature selection involves four main steps, namely, generation, evaluation, stopping criterion and validation, as introduced in [3]. The generation step aims to generate a subset of features as a candidate ready for selection. The evaluation step is designed to employ a heuristic function, e.g., information entropy [4], in order to evaluate the goodness of each candidate feature subset produced at the generation step, i.e., it is to measure how important the features in the candidate subset are to be used for training high performance classifiers. After the evaluation of the features importance, a stopping criterion is used to determine whether the features selected so far have been good enough. If so, we can stop generating and evaluating a further candidate feature subset and thus the current subset of selected features is validated at the last step. Otherwise, it is needed to repeat the feature selection by generating and evaluating another candidate subset of features. Fig. 1 shows the whole procedure of feature selection.

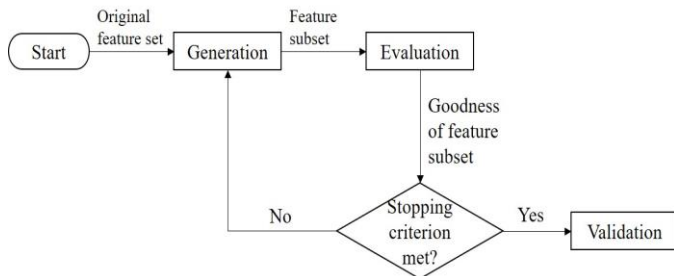


FIGURE 1. Feature selection process [3]

In general, feature selection can be achieved through two main approaches, namely, filter and wrapper. The filter approach is designed to employ a heuristic to evaluate directly the importance of each candidate subset containing one or more features, without considering whether the subset of selected features fits the nature of the learning algorithm that is employed for training a classifier. Some popular heuristics used for evaluating the goodness of features include distance functions [5], entropy [4], information gain [6], correlation coefficient [7] and co-variance [8]. Different from the filter approach, the wrapper approach is designed to employ a learning algorithm for evaluating the goodness of each candidate feature subset by checking the accuracy obtained using the classifier trained on the feature subset.

According to [3], the filter approach generally has low computational complexity, but may result in the case that the selected candidate feature subset is not really suitable for the employed learning algorithm to train a high performance

classifier [9]. In comparison with the filter approach, the wrapper approach can guarantee that the feature subset that results in the production of the best performing classifier is finally selected. In other words, for a feature set, there are n candidate feature subsets, which would finally lead to the production of n classifiers through using the n feature subsets. It is straightforward to find the best performing one out of the n trained classifiers, so the feature subset on which the best performing classifier is trained would finally be selected as the best candidate [3]. However, since it is needed to check all the possible non-empty subsets of the original feature set as candidates for selection, the wrapper approach would really lead to high computational complexity [3].

Furthermore, as introduced in [10], in the setting of traditional machine learning, it is a common practice to undertake feature selection in a single-task manner, which indicates that each candidate feature subset is typically evaluated in general to measure how well the use of the candidate feature subset can lead to the production of a classifier of good overall performance. However, while the overall performance of a classifier is good, the performance for some classes may be much lower. In order to keep more balanced performance over various classes, it has been very necessary to achieve effective selection of relevant features in a class-specific way. This way of feature selection is referred to as multi-task feature selection [1], which aims to undertake n feature selection tasks separately for n classes. In other words, a feature subset is evaluated in terms of its relevance to each specific class, where the n finally selected feature subsets are normally different although they may have some overlaps, by means of having some common features.

Both of the two above-mentioned feature selection approaches (filter and wrapper) can be used in the setting of MTFS, which will be explained in more details in Section 3 as part of the proposed MTEC approach.

3. The proposed multi-task approach of ensemble creation

The proposed MTEC approach involves various ways of feature selection as shown in Fig. 2, which is presented to show the whole procedure in the form of a theoretical framework of ensemble learning. In this paper, the MTEC approach is designed to involve three parts of ensemble creation, i.e., ensemble creation through filter-based MTFS, ensemble creation through wrapper-based MTFS and ensemble creation using the full set of original features.

For each of the above three parts, the created ensemble consists of n binary classifiers, where each binary classifier h_i is trained to identify from a new instance the presence or absence of features relevant to a specific class c_i . In order to

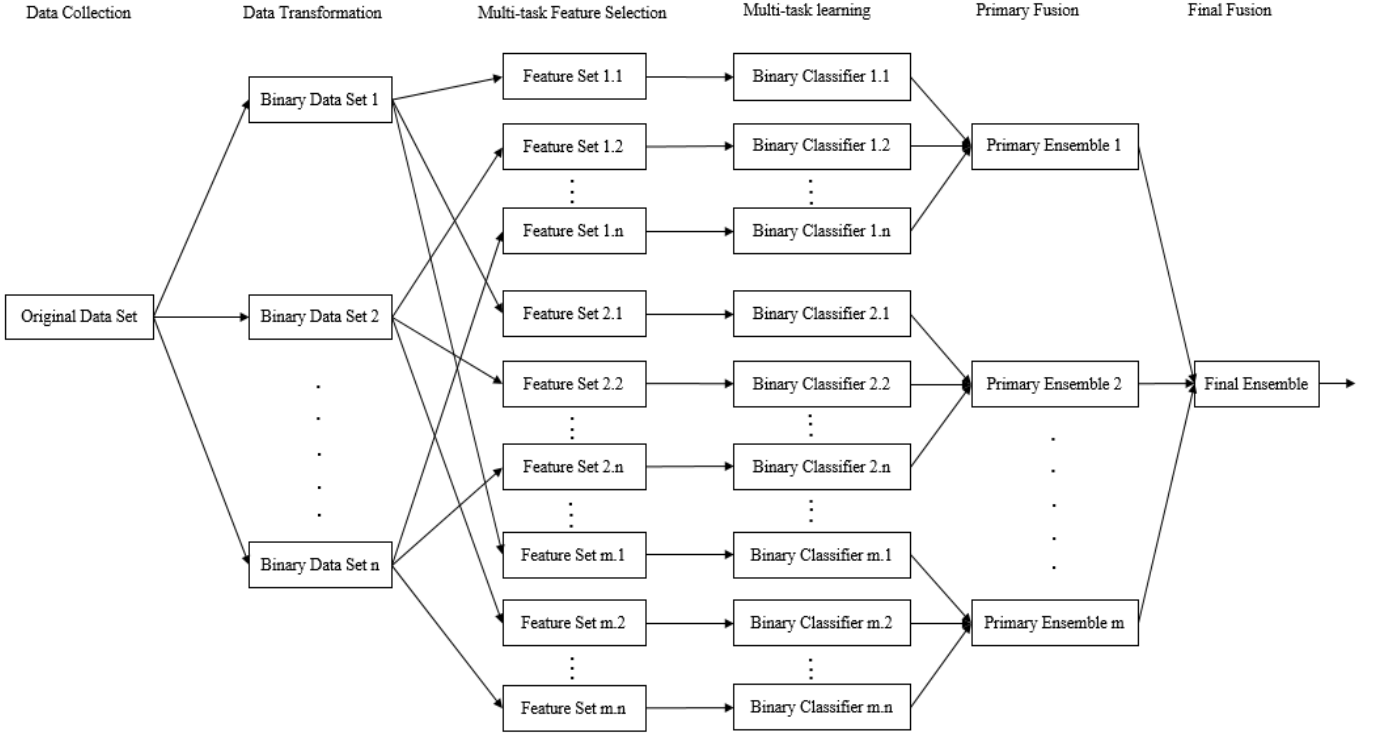


FIGURE 2. Procedure of the proposed multi-task ensemble creation approach

achieve such identification, the original data set that involves n classes needs to be transformed into n binary data sets in a binary relevance manner [2]. For example, while there are three classes (A, B and C) in a data set, it is normally needed to transform the data set into three new data sets that are assigned, respectively, the three pairs of classes: $A/\neg A$, $B/\neg B$ and $C/\neg C$. Through the above data transformation, a binary classifier h_i , which is aimed to identify effectively the presence or absence of features relevant to class c_i from a new instance, can be trained on a manipulated data set D_i . However, before training each binary classifier h_i , feature selection needs to be applied to each manipulated data set D_i , in order to achieve using only the features relevant for class c_i . In this way, there are n binary classifiers trained, respectively, on the n manipulated data sets (containing subsets of selected features), and the n binary classifiers make up an ensemble.

In the setting of filter-based MTFs, a heuristic method, e.g., the correlation-based feature subset selection method (CFS) [11], is applied to each of the n manipulated data sets, separately, for evaluating the goodness of each candidate feature subset involved in the manipulated data set D_i . Therefore, a feature subset f_i that is considered as the best candidate for class c_i can be produced from the manipulated data set D_i , and is then taken for training a binary classifier h_i as a member of the created ensemble.

In the setting of wrapper-based MTFs, it is essential to

employ a learning algorithm to train classifiers on various feature subsets drawn from each of the n manipulated data sets. However, while a rule learning algorithm is used, it is not necessary to take data manipulation in a binary relevance way but instead to learn a set of rules directly from the original data set D for each class c_i . In this way, from each set of rules having class c_i as the rule consequent, we can identify which features have been selected to generate rule antecedents and these selected features are thus considered to be relevant for the class c_i . Each feature subset f_i (containing features extracted from the antecedents of rules trained for class c_i) will finally be used for training a binary classifier h_i as a member of the created ensemble. More details can be found in [1], which shows in particular how the Prism algorithm can be used for MTFs.

On the basis of the above description, there will be three ensembles of binary classifiers, which are referred to as primary ensembles. Inside each primary ensemble, the n binary classifiers are fused in a selective way. In other words, each binary classifier would provide either a positive output (e.g., A) or a negative output (e.g., $\neg A$). Ideally, it would be expected that only one of the n binary classifiers provides a positive output, such that the positive output would be used as the final output of the ensemble for classifying a new instance. However, in reality, there could be the two cases: (1) multiple binary classifiers provide positive outputs; (2) none

of the binary classifiers provides a positive output. For both cases, the posterior probability of the positive class c_i resulting from each binary classifier h_i needs to be considered in order to measure the confidence of the positive output p_i of the binary classifier h_i and the positive output p_{best} of the most confident classifier h_{best} is selected as the final output of the primary ensemble for classification of a new instance.

The above three primary ensembles (created through three different ways of feature preparation) are finally fused to make up a final ensemble, i.e., the outputs of the three primary ensembles are fused through majority voting (choosing the class that obtains the most votes) to derive a final output for classifying each new instance.

4. Experimental results

Some experiments on multi-task ensemble creation are conducted in this section using an UCI data set on image segmentation [12]. This data set contains 2310 instances and there are totally 19 features extracted from various instances. Each of the 2310 instances belongs to one of the 7 classes, namely, brickface, sky, foliage, cement, window, path and grass, which essentially represent 7 target regions of outdoor images. For the production of the above data set, 7 outdoor images were used leading to the 2310 randomly drawn instances as indicated in [12]. The frequencies of the 7 above-mentioned classes are equally distributed over the data set, i.e., each class involves 330 instances.

The experimental study consists of two parts. In particular, the first part aims to show the performance of various existing methods and find the ones more suitable for learning from the data set, whereas the second part is undertaken to show the influence of using the proposed MTEC approach on the overall classification accuracy and the F-measure score for an individual class that obtains a lower score than all the other classes.

In the first part, the four learning algorithms, namely, Multi-layer Perceptron (MLP), C4.5, Naïve Bayes (NB) and K Nearest Neighbor (KNN), are used separately for building classifiers in a single-task manner using either the full set of features or the subset of selected features resulting from the adoption of the CFS method [11]. The above methods are compared in terms of overall classification accuracy.

In the second part, the C4.5 and KNN algorithms are used to build classifiers in a multi-task manner. In other words, the original full feature set is manipulated to enable multi-task feature preparation and each of the above two learning algorithms (C4.5 and KNN) is used to build classifiers using different feature sets prepared using various methods. The overall classification accuracy obtained using the proposed MTEC approach is compared with the one

obtained through various feature preparation ways, e.g., multi-task classification on the full feature set (MTCF), filter-based multi-task feature selection (FMTFS) and wrapper-based multi-task feature selection (WMTFS).

Based on the results obtained in the above two parts of the experimental study, for each of the two learning algorithms (C4.5 and KNN), the feature preparation method that leads the learning algorithm to produce the best performing classifier is selected for further comparison with the proposed MTEC approach, in terms of the lowest F-measure score among all the scores obtained for the 7 classes. In other words, while a method obtains the best overall performance, the performance for an individual class could be considerably worse than the one for all the other classes and thus we make the worst case comparison between the best performing method and the second best performing one, in order to see whether the best performing method obtains a higher valley than the second best performing method, among the performance obtained for all the classes.

10-fold cross validation is taken for all the experiments on the KNIME platform. In terms of setting the parameters for some of the parametric learning algorithms, the KNN algorithm is set to involve 5 nearest neighbors (weighted by Euclidean distance) for instance-based classification of new instances, i.e., the value of K is set to 5 and the Euclidean distance function is used to calculate the distance between a new instance and each of the training instances; The MLP algorithm is set to involve up to 100 training iterations and 10 units in each of the 2 hidden layers provided for training neural networks as the classifiers.

In the setting of MTFFS, the original data set is transformed into 7 binary data sets that involve the 7 pairs of class labels as follows: the ‘brickface’/‘¬brickface’ labels, the ‘sky’/‘¬sky’ labels, the ‘foliage’/‘¬foliage’ labels, the ‘cement’/‘¬cement’ labels, the ‘window’/‘¬window’ labels, the ‘path’/‘¬path’ labels and the ‘grass’/‘¬grass’ labels, in order to enable multi-task classification in the form of both single-classifier training and ensemble creation, following the procedure of the proposed MTEC approach as described in Section 3. In particular, feature preparation through taking the MTCF approach is designed to simply enable 7 binary classifiers to be trained on the 7 full sets of original features (i.e., each of the 7 full feature sets is identical to the original full feature set but is assigned a specific one of the 7 pairs of class labels shown above), using C4.5 or KNN. Moreover, for selection of relevant features in a class-specific manner, the FMTFS approach is based on the CFS method through data transformation in a binary relevance way [2] and the WMTFS approach is based on the Prism algorithm through learning a set of rules for class c_i towards the extraction of the features (relevant for c_i) from the antecedents of the rules [1]. The combination of MTCF, FMTFS and WMTFS essentially

constitutes MTEC.

The experimental results for the first part are shown in Table 1, for comparison of the classification accuracy obtained using various existing learning algorithms alongside different ways of feature preparation (i.e., using the full set of original features or a reduced set of selected features).

TABLE 1. Single-task classification accuracy on segment data

Methods	Full feature set	Reduced feature set obtained using CFS [11]
MLP [13]	0.831	0.788
C4.5 [14]	0.959	0.965
NB [15]	0.763	0.816
KNN [16]	0.960	0.927

The results shown in Table 1 indicate that the adoption of the CFS method [11] in a single-task manner may fail to achieve advances in the performance, i.e., the performance is improved when using the C4.5 and NB algorithms for training classifiers, but the performance is dropped when using the MLP and KNN algorithms. This phenomenon indicates that the nature of the CFS method leads to the production of a feature subset that is more suitable for C4.5 and NB but less suitable for MLP and KNN. However, the performance change resulting from the adoption of feature selection would indicate the likelihood of the creation of diversity between two classifiers trained on two different feature sets using the same algorithm, which shows the necessity to adopt both the full set of original features and different subsets of selected features produced using various methods of feature selection. Through the various ways of feature preparation, the final fusion of the primary ensembles created using various feature sets is likely to achieve further advances in the overall performance.

TABLE 2. Multi-task classification accuracy on segment data

Methods	Using C4.5 for training classifiers	Using KNN for training classifiers
MTCF [2]	0.960	0.961
FMTFS [2]	0.955	0.940
WMTFS [1]	0.958	0.964
MTEC	0.968	0.965

Since the use of the KNN algorithm leads to the best performing classifier resulting from the full set of original features and the use of the C4.5 algorithm leads to the best performing classifier resulting from the reduced set of selected features, in the second part of the experimental study, we investigate further the influence on the performance of C4.5-driven classification and KNN-driven classification, by using our proposed MTEC approach through combining various ways of feature preparation. The results on the overall performance for the second part are shown in Table 2.

Table 2 shows that the adoption of the MTEC approach

leads to slight advances in the performance on the overall classification accuracy, for both C4.5 and KNN that are used separately for training classifiers, in comparison with the other ways of feature preparation. Also, in comparison to the results shown in Table 1, the MTEC approach also achieves to advance slightly the performance of both C4.5 and KNN.

According to the results shown in Table 1 and Table 2, when C4.5 is used for building classifiers, the second best performing classifier results from the reduced set of selected features prepared using the CFS method. In particular, the overall classification accuracy obtained using the second best performing classifier is 0.965, while the lowest F-measure score among all the scores obtained for the 7 classes is 0.906, i.e., the F-measure score for the ‘window’ class is 0.906, which is lower than the F-measure scores obtained for all the other 6 classes. However, the adoption of the proposed MTEC approach results in a higher score (0.921) of F-measure for the ‘window’ class.

When using the KNN algorithm, the second best performing classifier results from using the WMTFS method for feature preparation, i.e., the overall classification accuracy obtained using the second best performing classifier is 0.964, while the lowest F-measure score among all the scores obtained for the 7 classes is 0.899 and the class corresponding to the lowest F-measure score is again ‘window’. However, the adoption of the proposed MTEC approach also achieves to obtain a higher score (0.905) of F-measure for the ‘window’ class.

Overall, the adoption of the proposed approach leads to effective creation of multiple ensembles that are diverse and complementary to each other, while C4.5 and KNN classifiers are trained on multiple feature sets obtained through various ways of feature preparation. Moreover, through using the MTEC approach, the results show that the overall performance is improved, in comparison with taking a single way of feature preparation, and the valley among the performance scores obtained for all the classes is also higher, in comparison with the adoption of another method that results in the second best overall performance.

5. Conclusions

In this paper, the MTEC approach has been proposed to create multiple ensembles on various feature sets for advancing the performance of image segmentation. Also, some experiments have been conducted to compare the performance obtained using our proposed MTEC approach with the performance obtained using C4.5 or KNN alongside various feature preparation methods in both single-task and multi-task manners. The experimental results show that the adoption of the proposed MTEC approach leads to an

improvement of the overall classification accuracy and also achieves a higher valley among the F-measure scores obtained for all the classes in comparison with the method achieving the second best overall performance. Therefore, the results can indicate the effectiveness of MTEC in dealing with image segmentation.

In the future, it is worth to investigate possible ways that can be taken to achieve effective extraction of image features of multiple levels [17]. This way of feature extraction can enable the operations of both feature selection and ensemble creation in multiple levels of granularity towards deep processing of image data. We will also explore the use of fuzzy approaches [18], [19] based on fuzzy sets [20], such that fuzzy image segmentation can be achieved to deal with ambiguous cases.

Acknowledgements

This work is supported by the Social Data Science Lab and the School of Computer Science and Informatics at the Cardiff University in the UK.

References

- [1] H. Liu and H. Zhao, "Multi-task feature selection for advancing performance of image segmentation", in Proceedings of the International Conference on Wavelet Analysis and Pattern Recognition, Chengdu, China, 2018, pp. 244-249.
- [2] H. Liu, M. Cocea, and W. Ding, "Multi-task learning for intelligent data processing in granular computing context", *Granular Computing*, Vol. 3, No. 3, pp. 257-273, 2018.
- [3] M. Dash, and H. Liu, "Feature selection for classification", *Intelligent Data Analysis*, Vol. 1, pp. 131-156, 1997.
- [4] C. E. Shannon, "A mathematical theory of communication", *Bell Systems Technical Journal*, Vol. 27, No. 3, pp. 379-423, 1948.
- [5] C. Cassisi, P. Montalto, M. Aliotta, A. Cannata, and A. Pulvirenti, "Similarity measures and dimensionality reduction techniques for time series data mining", in *Advances in Data Mining Knowledge Discovery and Applications*, A. Karahoca, Ed. InTech, 2012, pp. 71-96.
- [6] S. Kullback and R. A. Leibler, "On information and sufficiency", *The Annals of Mathematical Statistics*, Vol. 22, No. 1, pp. 79-86, 1951.
- [7] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution", in Proceedings of the 20th International Conference on Machine Learning, Washington, D.C., U.S.A., 2003, pp. 856-863.
- [8] D. Barber, *Bayesian Reasoning and Machine Learning*. Cambridge: Cambridge University Press, 2012.
- [9] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection", *Journal of Machine Learning Research*, Vol. 3, pp. 1157-1182, 2003.
- [10] H. Liu, M. Cocea, and W. Ding, "Decision tree learning based feature evaluation and selection for image classification", in Proceedings of the 2017 International Conference on Machine Learning and Cybernetics, Ningbo, China, 2017, pp. 569-574.
- [11] M. A. Hall and L. A. Smith, "Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper", in Proceedings of the Twelfth International Florida Artificial Intelligence Research Society Conference, Orlando, Florida, 1999, pp. 235-239.
- [12] M. Lichman, *UCI Machine Learning Repository*, <http://archive.ics.uci.edu/ml>, 2013.
- [13] R. Collobert, and S. Bengio, "Links between perceptrons, MLPs, and SVMs", in Proceedings of the 21st International Conference on Machine learning, Banff, Alberta, Canada, 2004.
- [14] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, 1993.
- [15] I. Rish, "An empirical study of the Naive Bayes classifier", in Proceedings of IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, Seattle, USA, 2001, Vol. 3, No. 22, pp. 41-46.
- [16] J. Zhang, "Selecting typical instances in instance-based learning", in Proceedings of the Ninth International Workshop on Machine Learning, Aberdeen, Scotland, United Kingdom, 1992, pp. 470-479.
- [17] X. Wang, W. Ding, H. Liu, and X. Huang, "Shape recognition through multi-level fusion of features and classifiers", *Granular Computing*, Vol. 5, No. 3, 2020.
- [18] H. Liu and L. Zhang, "Fuzzy rule-based systems for recognition intensive classification in granular computing context", *Granular Computing*, Vol. 3, No. 4, pp. 355-365, 2018.
- [19] S. M. Chen and Y. C. Chang, "Weighted fuzzy rule interpolation based on GA-based weight-learning techniques," *IEEE Transactions on Fuzzy Systems*, Vol. 19, No. 4, pp. 729-744, 2011.
- [20] L. Zadeh, "Fuzzy sets", *Information and Control*, Vol. 8, No. 3, pp. 338-353, 1965.