I.J.Modern Education and Computer Science, **2011, 4, 25-31**
Published Online July 2011 in MECS (http://www.mecs-press.org/)

Effective Training Data Improved Ensemble Approaches for Urinalysis Model

Ping Wu
School of Information Science and Technology, East China Normal University, Shanghai 200062, China
Email: pwu@cc.ecnu.edu.cn

Min Zhu and Peng Pu
School of Information Science and Technology, East China Normal University, Shanghai 200062, China
Email: mzhu@cc.ecnu.edu.cn, puxiaoming@163.com

Tang Jiang
Clinical Laboratory, The First Affiliated Hospital of Sun Yat-sen University, Guangzhou 510080, China
Email: jennyj@vip.163.com

***Abstract*—Urinalysis remains one of the most commonly performed tests in clinical practice. Laboratory work can be greatly relieved by automated analyzing techniques. However, noisy and imbalanced urine samples make automatically identifying and classifying urine-related diseases become very difficult. This paper proposed hybrid sampling-based ensemble learning strategies by improving training data and classification performance. Having compared the effectiveness of several learning classifiers and data processing techniques, the experiments showed that the suggesting methods provided better classification accuracy than other approaches.**

***Index Terms*—urinalysis, noisy data, imbalanced data, sampling methods, classification, ensembles**

I. INTRODUCTION

Improvements in automated urinalysis are largely requested by laboratory practice. In most laboratories, the general urine-screening procedure consists of chemical test strip, microscopy of urine sediment and automated urinary flow cytometry (commonly known as UF). Different methodologies in screening use their own test parameters and lead to different types of the test report. Traditionally, manual microscopy is regarded as a testing standard. But it is extremely time consuming and given the large volume of requests for routine urinalysis, its precision could be influenced by some misreading of a person. Recent studies with traditional statistical analysis show that information provided by urinary flow cytometry combined with test strip analysis, followed by visual microscopic examination of discordant samples offers the optimal analysis mode for urine screening [1-3]. Nevertheless, the test strip procedure has significant false negative and false positive rates [2], improvements in automated UF predictions are much more needed to eliminate labor-intensive manual microscopy.

In our previous study, a GA-based fuzzy rule learning algorithm [4] was proposed to solve this problem. We determined that an UF predicting model is useful for the selection of urine samples that need microscopic confirmation in different populations. The experiments started comparing the urine data from the test strip and UF on all samples. If the results from the test strip agree with those from UF, they are accepted. Otherwise, GA-based methods are invoked to find the UF features to classify samples and create rules to specify microscopic examined parameters. Although having achieved some improvements, their classification performances were not satisfied.

A problem that affects practical classification accuracy is poor quality of the data. Noisy and imbalanced urine samples in our application are challenges to setup a predicting model. Noise might be in the classification label set by comparing the results of the test strip and UF, or in the features of UF, for automated UF systems provide only scattergram analysis, which limits the number of types of particles that can be differentiated in urine samples.

The aim of the present work was to effectively handle imperfections in the data. One approach is to use ensemble learners which have been demonstrated to perform better than a single classifier and be robust in the presence of those imperfections [5-9]. The other approach is based on improvement of training data by removing class noise and adjusting imbalance.

When creating ensemble learners, it is very important to use the available data to produce classifiers with uncorrelated errors. The general methods include manipulating the input features, using a different random class relabeling, injecting randomness and subsampling the training examples by generating different views of the data to build every individual classifier [5]. Bagging and AdaBoost, the two most popular examples, belong to subsampling method and widely used to enforce weak classifiers. Random Forest [8], a variant of Bagging, is also described as one of the most powerful ensembles. Either of them operates by taking a base learning algorithm and invoking it many times with different training sets. In Bagging, each bootstrap sample, i.e.

training set, is constructed by subsampling the original one with replacement. The results of all the learners are finally integrated by majority voting. The Random Forest consists of many individual trees. Each tree votes on an overall classification for the given set of data and the algorithm chooses the individual classification with the most votes. There are two different sources of randomness in Random Forests: random bootstrap and random selection of attributes. The AdaBoost algorithm takes a different resampling approach by maintaining a set of weights over the original training set and adjusting these weights after each classifier is learned by the base learning algorithm. Bagging tends to reduce the variance and in turn to be robust in the presence of some levels of noise. AdaBoost is showed to bring on larger classification error reductions than Bagging on average in terms of the bias and variance. Bias measures how closely the learning algorithm's average guess matches the target, while variance measures how much the guess fluctuates for the different training sets of the given size [9].

In the case of class noise, removing misclassified instances from the training data resulted in a classifier with significantly higher predictive accuracy [16]. In practice, there are two main approaches towards learning from noisy data: data cleansing and robust learning. As for data cleansing methods, the designs behind them are very similar: classifiers trained from a portion of the training data are used to justify the excluded training samples, with misclassified instances identified as noise. As for robust learning methods, they add a new component to the learning algorithms that can handle noisy data and maintain good performance [17].

There have been many approaches to learn from imbalanced data [10-11]. At the data level, sampling learning is the one of most common techniques. The basic idea of sampling is to reduce imbalances by altering the distribution of training instances. In general, this can be fulfilled by over-sampling and under-sampling approaches. The over-sampling method increases the number of minority class instances to reduce the degree of imbalanced distribution. The under-sampling method extracts a smaller set of majority instances while preserving all the minority instances.

This paper investigates the potential of Bagging, Random Forest, AdaBoost ensembles and aims to present a combined ensemble approach to further reduce the predicting errors of urinalysis. At the data level, data processing techniques are studied and we propose using a hybrid approach that consists of rebalancing data and reducing the effect of class-label noise and incomplete UF attributions in order to improve the quality of urine dataset. The experiments have demonstrated that our suggesting approaches yield better classification results.

The remainder of this paper is organized into four sections. Section 2 describes important implementation techniques. The setup of experiments and evaluation methods are presented in Section 3. Section 4 illustrates the experiment results, as well as performance evaluation of related approaches. Finally, conclusions and future works are discussed in the last section.

II. Approaches

A. Bagging Algorithm

The Bagging algorithm votes classifiers generated by different bootstrap samples. Each classifier is trained on the average of 63.2% of training instances. Bagging takes a base classification learning algorithm L and training set S as input, and returns a committee of classifiers C^*. A classifier C_i is built from each bootstrap samples. A final classifier $C^*(x)$ is built from $C_1(x)$, $C_2(x)$,…, $C_N(x)$, where each classifier returns a classification $y \in Y$, whose output is the class that obtains the most votes from the committee members when applied to a new case x. The algorithm is shown as below.

Input: *S*- training dataset; *L*- base classifier; *N*- number of bootstrap samples; *I*- indicator function

1. for $i = 1$ *to* N {
2. B = bootstrap sample from S
3. $C_i = L(B)$}
4. $C^*(x) = \arg\max_{y \in Y} \sum_{i=1}^{N} I(C_i(x) = y)$

Output: classifier C^*

Figure 1. The Bagging Algorithm.

B. Random Forest

The Random Forest consists of many individual trees. It is designed to be diverse and more robust with respect to noise by using randomness: random bootstrap and random selection of attributes. As shown in Fig. 2, each tree forms a dataset “inbag” by sampling with replacement members from the training set. This is referred to as “bootstrapping”. The number of dataset in the “inbag” data set is equal to that of the training data set. Usually one third of the training data set is “out of bag” and used to test the tree. Diversity is obtained by randomly choosing attributes at each node and using the attributes that provide the highest level of learning.

Input: *S*- training dataset; *L*- base classifier; *M*- number of trees; *I*- indicator function

1. for $i = 1$ *to* M {
2. B_i = bootstrap samples from S
3. T_i = a tree derived from B_i, at each node restricted to a randomly selected attributes without pruning the decision tree using L
4. $C_i = T_i(B_i, L)$ }
5. $C^*(x) = \arg\max_{y \in Y} \sum_{i=1}^{M} I(C_i(x) = y)$

Output: classifier C^*

Figure 2. The Random Forest Algorithm.

C. AdaBoost Algorithm

The AdaBoost algorithm generates a set of classifiers and votes them. But it takes a different resampling approach: sampling is proportional to an instance's weight and changes the weights of the training instances based on classifiers that were previously built. The goal is to force the inducer to minimize expected error over different input distributions. Finally, the predictions of all classifiers are combined by weighted voting as shown in Fig. 3.

Input: *S*- training dataset of size m; *L*- base classifier; *N*- number of iterations

1. *B* = *S* with instance weights assigned to be 1
2. For *i = 1 to N* {
3. $C_i = L(B)$
4. weighted error $\varepsilon_i = \frac{1}{m} \sum_{x_j \in B: C_i(x_j) \neq y_j} weight(x)$
5. If $\varepsilon_i > 0.5$, set B to a bootstrap sample from S with weight 1 for every instance and goto step 3
6. $\beta_i = \varepsilon_i / (1 - \varepsilon_i)$
7. new weights: if $C_i(x_j) = y_j, weight(x_j) = weight(x_j)\beta_j$
8. Normalize the weights of instances}
9. $C^*(x) = \arg\max_{y \in Y} \sum_{i: C_i(x) = y} \log(1/\beta_i)$

Output: classifier *C**

Figure 3. The AdaBoost Algorithm.

D. Combined Ensemble Algorithm

Considering different mechanisms and effects of the above three algorithms, we try to obtain the benefits of them to retain AdaBoost's bias reduction while adding more Bagging's variance reduction and Random Forest's diversity. The Combined Ensemble Algorithm (CEA) therefore has been presented to combine Bagging and AdaBoost techniques based on Random Forest. Fig. 4 shows the framework of that Bagging-AdaBoost-Random Forest algorithm.

Input: *S*- training dataset of size m; *L- Random Forest classifier; N-* number of iterations; I_t*- specifying the iteration at which each subcommittee t should terminate*

1. *B* = *S* with instance weights assigned to be 1
2. *k=1*
3. For *i = 1 to N* {
4. If I_t *=i, reset B to random weights; increment k*
5. $C_i = L(B)$
6. If $\varepsilon_i > 0.5$, *reset B to random weights; increment k* goto step 5
7. $\beta_i = \varepsilon_i / (1 - \varepsilon_i)$;compute new weights and normalize}
8. $C^*(x) = \arg\max_{y \in Y} \sum_{i: C_i(x) = y} \log(1/\beta_i)$

Output: classifier *C**

Figure 3. The Combined Algorithm.

E. Data Cleansing

The noisy urine data necessitates the use of noise handling techniques. In the experimental model, noise filters were used to identify misclassified instances in a training set. The basic idea is to use a set of learning algorithms to create classifiers that act as a filter for the training data [16]. It assumes that the errors in the class labels are independent of the particular model. We used two approaches to implement filtering. One way is to construct a filter using one algorithm. The other approach uses an ensemble filter to detect misclassified instances by constructing a set of base-level classifiers and then using their classification errors to identify misclassified instances.

In general, collecting information from different models provides a better method for detecting misclassified instances than from a single model. Thus, ensemble filters will make fewer detection errors than a single algorithm filter.

F. Imbalance Handling

Imbalances can occur either between the two classes or within a single class. A between-class imbalance corresponds to the case where the number of examples representing the one class differs from another class (majority class vs. minority class); and a within-class imbalance corresponds to the case where a class is composed of a number of different subclusters and these subclusters do not contain the same number of examples (small disjuncts vs. large disjuncts) [12]. At the data level, one of most common way to handle imbalance is sampling. The basic idea is to minimize rarity by altering the distribution of training set. Typically the class distribution is altered to reduce the problems associated with minority classes, but the distribution of cases can also be altered to deal with small disjuncts [13].

The basic sampling techniques include over-sampling and under-sampling. Both decrease the overall level of imbalance and improve the performance of classifiers. However, random over-sampling might lead to minority instances over-represented and under-sampling could potentially lose some informative instances of majority instances. In practice, getting a trade-off between the over- and under-sampling can minimize total misclassification errors.

III. Experimental Design

A. Data Preparation

The urine data used in this study was provided by the First Affiliated Hospital of Sun Yat-sen University, located in Guangzhou, P.R. China. There are 2395 samples with 50 features grouped in three categories of erythrocyte-, leukocyte- and bacteria-related diseases. These features are believed to be basic information and symptoms required by senior physicians to diagnose urine-related diseases in clinic. Each group has its corresponding features in terms of experts' knowledge. Some of the features are granted to be very important and some combinations of them may influence the diagnosis. Traditionally, manual microscopy is considered to be a standard to help physicians to make a final decision.

In our experimental setup, UF attributes are firstly used to classify a urine sample into four types which are normal, inherited disorder, hematuria, hematuria-risk ones, and then microscopic examination is required to review related features of last three types with different ways. The erythrocytes group has been included in the experiment. Similar experiments can be conducted based on the other two groups.

The samples prepared in the dataset was labeled with 350 inherited disorder reviewed, 67 hematuria reviewed, 1282 normal, and 696 hematuria-risk reviewed cases according to the results of the test strip and UF. By analyzing the relevance between attributes and the class attribute respectively, we added some related attributes of available data to basic features physicians have suggested in order not to lose any useful information. The class distribution is shown in Table Ⅰ.

TABLE I.
CLASS DISTRIBUTION IN THE ORIGINAL DATASET

Class	Number of Instances	Percentage
Inherited disorder	350	14.61
Hematuria	67	2.80
Normal	1282	53.53
Hematuria-risk	696	29.06

B. Data Reprocessing

The work described here focuses on improving the quality of training data.

The classification techniques usually assume that the training samples are uniformly-distributed between different classes. In our application, the imbalance among different classes has been shown in Table Ⅰ and small disjuncts were also found probably within majority classes by applying the classifier C4.5 to the dataset. Therefore, we present a Combined Sampling Method (CSM) that adjusts the number of each class respectively by altering the distribution of cases, getting a trade-off between the over- and under-sampling.

Class noise can occur for subjectivity, data entry error or inadequacy of the information used to classify each instance [13]. Class-noisy urine data may arise from inadequate information. We use noise filters to remove misclassified instances from the training set.

To obtain a more reliable estimate, all experiments were performed using 10-fold cross validation. The urine dataset is separated into ten parts, with the combination of 90% as training and the remaining 10% as test. Repeating ten times with each part acted as test data once, this process allows a relatively complete use of the samples in the data set. In addition, 10 independent runs of 10-fold cross validation can alleviate any biasing that may occur during the random partitioning process.

C. Experimental Model

Experimental models were designed to describe the different strategies and configurations of combined methods. To validate the performance of methods, the experiments were performed based on six different models for the same urine dataset.

The first model is a typical learning approach that contains four well-known learning algorithms: C4.5, Bagging, Random Forests and AdaBoost implemented in Weka[15].

The second model uses the designed sampling method CSM to adjust imbalance and applies four algorithms of the first model again.

The third model is also based on the first model but instead it employs a one algorithm filter to discard the mislabeled instances.

The fourth model is same as the third except that it uses an ensemble filter to detect misclassified instances.

The fifth model combines the third model with the second, removing mislabeled cases by single-algorithm filter and then tuning unbalance; whereas the sixth model combines the fourth and second using an ensemble filter instead of one algorithm filter.

.

D. Evaluation methods

Our study utilized basic evaluation measures, F-Measure and ROC Area to evaluate the performance of all learning models on noisy and imbalanced urine data.

Three commonly used evaluation measurements are precision, sensitivity and specificity.

The classification precision is defined as the correctly classified instances rate provided that a specific class has been predicted. TP and TN denote the number of positive and negative instances correctly classified, while FP and FN refer to the number of misclassified positive and negative instances.

$$\Pr ecision = \frac{TP}{TP+FP}. \qquad (1)$$

Sensitivity measures the proportion of actual positives which are correctly identified.

$$Sensitivity = \frac{TP}{TP+FN}. \qquad (2)$$

Specificity is defined to measure the proportion of negatives which are correctly identified.

$$Specificity = \frac{TN}{TN+FP}. \qquad (3)$$

The F-Measure is a combined measure for precision and sensitivity.

$$F - Measure = \frac{2\times \Pr ecision \times Sensitivity}{\Pr ecision + Sensitivity}. \qquad (4)$$

The receiver operating characteristic curve (ROC) is a fundamental tool for diagnostic evaluation. It plots the true positive rate on the y-axis versus the false positive rate on the x-axis. Each prediction result represents one point in the ROC space. The area under the ROC curve (AUC) is used to measure the model's ability to identify instances of each class.

Ⅳ. RESULTS AND ANALYSIS

A. *Urine Datasets*

The original urine dataset described in the previous section contained some kind of noise and imbalance. The experiments started from adjusting imbalance using the CSM method and Table Ⅱ shows the result of class distribution of this second dataset. The total number of instances remained the same, while the number of majority classes and minority classes slightly decreased and increased respectively.

TABLE III.
CLASS DISTRIBUTION IN THE SECOND DATASET

Class	Number of Instances	Percentage
Inherited disorder	387	16.16
Hematuria	224	9.35
Normal	1106	46.18
Hematuria-risk	678	28.31

The third dataset was also obtained from the original urine dataset by employing a one algorithm filter to discard the mislabeled instances. C4.5 was chosen to construct the filter in our study. The result shown in Table Ⅲ illustrated that the total number of instances decreased and mislabeled cases were found in the last three classes. The reduced rate of each class ordered in the table is 0.00%, 5.97%, 8.89% and 42.24%, which indicated that there has been much noise in "Hematuria-risk" class.

TABLE V.
CLASS DISTRIBUTION IN THE THIRD DATASET

Class	Number of Instances	Percentage
Inherited disorder	350	17.65
Hematuria	63	3.28
Normal	1168	58.90
Hematuria-risk	402	20.27

By applying an ensemble filter to the original dataset, that we chosen Bagging and C4.5, the total and each class numbers of instances are demonstrated in Table Ⅳ. The reduced rate of each class ordered in the table is 0.00%, 19.4%, 12.71% and 54.45%, which indicated that much more noise in "Hematuria-risk" class and certain amount of noise in "Hematuria" and "Normal"class have been detected.

TABLE II.
CLASS DISTRIBUTION IN THE FORUTH DATASET

Class	Number of Instances	Percentage
Inherited disorder	350	19.02
Hematuria	54	2.93
Normal	1119	60.82
Hematuria-risk	317	17.23

The datasets that the fifth and sixth model used are shown in Table Ⅴ. The number of instances of each class displays that unbalance among four classes has been tuned with the numbers of majority classes decreased and minority classes increased in some measure. Note that the total number of instances in the fifth dataset is the same as in the third dataset, while the sixth and fourth dataset have the same instance numbers as well.

TABLE IV.
NUMBER OF INSTANCES IN THE FIFTH AND SIXTH DATASET

Class	the Fifth Dataset	the Sixth Dataset
Inherited disorder	404	385
Hematuria	191	174
Normal	955	897
Hematuria-risk	433	384

B. *Performance Evaluation*

The performance of each learning method was evaluated using C4.5 as a reference and six experimental models designed in the earlier section were used. In the following tables, M come from M1 to M6 has the meaning "Model". For instance, M1 refers to the first model. C, B, R and A in the first column of tables refer to C4.5, Bagging, Random Forest and AdaBoost respectively.

Table Ⅵ displays the basic performance and F-Measure for each method included in 6 models. Note that we use ensemble CEA just for the original urine dataset and the second dataset rebalanced by CSM to testify its performance. The results show that CEA was better than other learning methods.

When sampling CSM is introduced, the overall performances of all methods in the second model were much better than in the first model, especially for AdaBoost.

For noise handling, when given data from an ensemble filter, all methods in Model 4 performed much better than Model 3 with single-algorithm filtering.

When filtering combined with sampling CSM, all methods in Model 5 were able to surpass Model 3 and Model 6 went beyond Model 4, whereas Model 4 had slightly higher performance than Model 5

TABLE VI.
EVALUATION MEASURES OF METHODS

Method	Performance Evaluation			
	Precision	Sensitivity	Specificity	F-Measure
M1_C	0.718	0.730	0.760	0.718
M1_B	0.730	0.738	0.774	0.731
M1_R	0.718	0.732	0.758	0.718
M1_A	0.686	0.696	0.747	0.690
M1_CEA	0.738	0.749	0.771	0.736
M2_C	0.869	0.87	0.917	0.869
M2_B	0.887	0.888	0.928	0.887
M2_R	0.889	0.889	0.924	0.888
M2_A	0.885	0.885	0.926	0.885
M2_CEA	0.898	0.899	0.933	0.898
M3_C	0.922	0.922	0.933	0.922
M3_B	0.913	0.914	0.924	0.913
M3_R	0.877	0.880	0.867	0.874
M3_A	0.916	0.920	0.924	0.916
M4_C	0.982	0.982	0.985	0.982
M4_B	0.984	0.984	0.987	0.984
M4_R	0.954	0.955	0.950	0.954
M4_A	0.979	0.979	0.982	0.979
M5_C	0.948	0.948	0.969	0.948
M5_B	0.953	0.953	0.972	0.953
M5_R	0.949	0.949	0.963	0.948
M5_A	0.952	0.952	0.969	0.952
M6_C	0.986	0.986	0.992	0.986
M6_B	0.988	0.988	0.994	0.988
M6_R	0.980	0.980	0.986	0.980
M6_A	0.987	0.987	0.993	0.987

In Table Ⅶ, we show the performance of the above methods using ROC Area. The last column in Ⅶ gives the average values of area under the ROC curve, while the other 4 columns correspond to class-label ranging from C1 to C4.

As shown in Table Ⅶ, CEA and three ensemble algorithms, Bagging, Random Forest and AdaBoost, perform significantly better when implemented with CSM. This is due to the reductions of imbalance among training classes, as well as rare cases in class 3 and class 4. Another important fact is that ensemble learners seem to be impacted more by using CSM when area under the ROC curve is used to measure performance. Comparing Model 3 to Model 1, it is shown that Random Forest and AdaBoost made no significant differences as Model 3 did not apply sampling CSM. All of methods in Model 4,

TABLE Ⅶ
AUC MEASURE OF METHODS

Method	ROC Area				
	C1	C2	C3	C4	AVG
M1_C	1.000	0.948	0.762	0.714	0.788
M1_B	1.000	0.964	0.822	0.787	0.842
M1_R	1.000	1.000	0.807	0.764	0.828
M1_A	1.000	1.000	0.789	0.740	0.811
M1_CEA	1.000	1.000	0.837	0.803	0.856
M2_C	1.000	1.000	0.896	0.875	0.917
M2_B	1.000	1.000	0.953	0.944	0.962
M2_R	1.000	1.000	0.963	0.954	0.970
M2_A	1.000	1.000	0.957	0.948	0.965
M2_CEA	1.000	1.000	0.969	0.962	0.975
M3_C	1.000	0.984	0.943	0.914	0.948
M3_B	1.000	0.980	0.975	0.963	0.977
M3_R	1.000	0.959	0.903	0.669	0.874
M3_A	0.999	0.976	0.932	0.789	0.916
M4_C	0.999	1.000	0.985	0.975	0.986
M4_B	0.999	1.000	0.995	0.991	0.995
M4_R	1.000	1.000	0.992	0.986	0.993
M4_A	0.999	1.000	0.996	0.993	0.996
M5_C	1.000	1.000	0.958	0.939	0.966
M5_B	1.000	1.000	0.985	0.978	0.988
M5_R	1.000	1.000	0.989	0.981	0.991
M5_A	1.000	1.000	0.989	0.985	0.992
M6_C	1.000	1.000	0.988	0.982	0.990
M6_B	1.000	1.000	0.995	0.993	0.996
M6_R	1.000	1.000	0.998	0.996	0.998
M6_A	1.000	1.000	0.996	0.995	0.997

Model 5 and Model 6 achieved comparable higher accuracies.

Ⅴ. DISCUSSION AND CONCLUSIONS

The aim of this paper was to do research whether the suggesting urinalysis predicting models can be efficiently incorporated into laboratory investigation of urine-related diseases. Several approaches including ensemble CEA, sampling CSM and noise filtering methods are presented to improve the performance of these urinalysis models.

The ensemble CEA was compared to four learning methods, which showed that it generally outperformed those existing algorithms.

For CSM approach, the experiments demonstrated that it can solve imbalance problems by adjusting distribution of instances and the average of precision, sensitivity, specificity, F-Measure, ROC Area of learners have been significantly improved after applying the sampling CSM.

Filtering was designed to identify class noise, i.e. mislabeled instances in our urine dataset. The results of an empirical evaluation indicated that filtering greatly improved classification accuracy. A comparison of ensemble filtering to single algorithm filtering illustrated that the ensemble filter performed better than the individual filter.

We finish by indicating some possible directions for future research. One of future work will include an investigation of urine data by checking class-label using microscopy of urine sediment. Using domain knowledge to estimate the amount of class-label noise in a dataset would be the best choice. Another future work will be to extend the filtering approach by labeling errors in training data. Errors can be cataloged into different levels. Some types of errors can be corrected automatically, while other mislabeled data should be confirmed by domain experts. Finally, using a larger urinary test database for further improving the performance of the classifier.

ACKNOWLEDGMENT

The authors would like to thank all the reviewers for their valuable comments and suggestions.

REFERENCES

[1] Josiane Steinmetz, Joseph Henny and Rene Gueguen, Stepwise strategies in analysing haematuria and leukocyturia in screening Clin Chem Lab Med 44(4), 2006, pp. 464-470.

[2] Elin RJ, Hosseini JM, Kestner J, et al., Comparison of automated and manual methods for urinalysis. Am J ClinPathol. ,1986, pp. 731-737.

[3] Tang Jiang, et al., Development of a new mode of urinary test and its software Chin J Lab Med 29(7), 2006.

[4] Ping Wu, et al., A Hybrid GA-based Fuzzy Classifying Approach to Urinary Analysis Modeling. Genetic and Evolutionary Computation Conference 2009, pp. 2671-2677.

[5] Dietterich T.G., Machine learning research: Four current directions. AI Magazine 18(4), 1997, pp.97–136.

[6] BREIMAN Leo, Bagging Predictors. Machine Learning, 24(2), 1996, pp. 123–140.

[7] T. G. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization, Mach. Learn. 40(2), 2000, pp. 139–157.

[8] Breiman L., Random forests. Machine Learning 45(1), 2001,pp.5–32

[9] Bauer E. and Kohavi R., An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learning 36,1999, pp. 105-139

[10] Nitesh V. Chawla , Nathalie Japkowicz , and Aleksander Ko lcz, Editorial: Special Issue on Learning from Imbalanced Data, ACM SIGKDD Explorations Newsletter 6(1), 2004, pp.1-6.

[11] Haibo He, Garcia, and E.A., Learning from Imbalanced Data, IEEE Transactions On Knowledge And Data Engineering, 21(9), 2009,pp. 1263 – 1284

[12] Nathalie Japkowicz, Concept-Learning in the Presence of Between-Class and Within-Class Imbalances, Advances in Artificial Intelligence, 2056, 2001, pp.67-77

[13] Gary M. Weiss, Mining with rarity: a unifying framework, ACM SIGKDD Explorations Newsletter 6(1), 2004, pp.7-19.

[14] M. Kubat, S. Matwin, Addressing the curse of imbalanced training sets: one-sided selection. International Conference on Machine Learning 1997, pp. 179-186.

[15] I. H. Witten and E. Frank, Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, San Francisco, California, 2nd edition, 2005.

[16] Brodley, C. E. and Friedl, M. A., Identifying mislabeled training data. Journal of Artificial Intelligence Research, 11, 1999, pp.131–167.

[17] Xingquan Zhu, et al., “Cleansing Noisy Data Streams,” ICDM '08 Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, 2008, pp. 1139-1144.

Ping Wu received the B.S. degree in Computer Science from Xi’an Jiaotong University in 1986 and the M.S. degree in Computational Mathematics from Xi’an Jiaotong University in 1989.

She is currently an Associate Professor in Computing Center, School of Information Science and Technology, East China Normal University. Her research interests include learning classifier systems, genetic algorithms and machine learning.

