2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)

Cancer Classification Using Microarray Data By DPCAForest

Xiaoheng Deng
School of Computer Science and Engineering
Central South University
Changsha 410083, China
dxh@csu.edu.cn

Yuebin Xu
School of Computer Science and Engineering
Central South University
Changsha 410083, China
csurookie@csu.edu.cn

***Abstract*—Supervised learning is a powerful tool that has shown promise when applied towards bioinformatics data sets. Deep forest, a supervised ensemble model based on decision trees, has been proven to have excellent classification performance and strong generalization ability across different fields. However, when dealing with high-dimensional and small-sample gene expression data, commonly used supervised learning methods including deep forest may not be effective. In this paper, we propose DPCAForest, a deep-forest-based model, which integrates deep forest and dynamic principle component analysis. DPCAForest adaptively generates the minority samples based on sample distribution, then conducts principle component analysis dynamically synchronized with growth of deep forest to reveal the important features with the highest variance. Dynamic PCA enables the model to perform feature extraction in a data-driven way based on cross-validation, and the model can obtain fusion information across layers. In the experimental studies, DPCAForest is verified on Adenocarcinoma, Brain, Colon, Small Round Blue Cell Tumors (SRBCTs) and NCI-60 cancer data sets and demonstrated desirable or better performance than state-of-the-art methods in terms of accuracy and F-Measure.**

***Index Terms*—cancer classification, deep forest, microarray, feature extraction.**

I. Introduction

The classification of cancer sub-types is of great importance to cancer disease diagnosis and therapy. Discrimination between two classes such as normal and cancer samples and between two types of cancers based on gene expression profiles is an important problem which has practical implications as well as the potential to further our understanding of gene expression of various cancer cells [1]. DNA microarrays now permit scientists to screen thousands of genes simultaneously and determine whether those genes are active, hyperactive or silent in normal or cancerous tissue. Because these new microarray devices generate bewildering amounts of raw data, new analytical methods must be developed to sort out whether cancer tissues have distinctive signatures of gene expression over normal tissues or other types of cancer tissues.

In recent years, the development of deep neural networks have achieved great success in various applications, especially in visual and speech recognitions [2]. With the inspiration of deep learning, numerous deep learning-based approaches have been proposed for cancer sub-types classification. However, deep learning may face many deficiencies. On one hand, DNNs are with too many hyper-parameters, and the learning performance depends seriously on careful parameter tuning [2]. In other words, different authors are actually using different learning models even if they are using the same neural networks when the hyperparamenters are not the same such as the convolutional layer structures. One the other hand, most importantly, the learning procedure of DNNs requires tremendous amounts of training data. However, there aren't large enough samples for most cancer genomic data at present. This makes it unruly to get anticipate classification performance using deep neural networks in practice, especially on the small-scale biology data sets [3].

To address the the disadvantages of deep learning, deep forest was proposed [2]. This is a novel decision tree ensemble model, with a cascade structure enabling representation learning by forests. Muti-grained scanning enables the model to be contextual or structual aware. Moreover, the cascade levels can be detemined automatically so that the complexity can be determined in a data-dependent way, which makes it work well on small-scale data with much fewer hyper-parameters.

Neverthless, biology data such as gene expression data are often charaterized by high dimensional and class imbalance. Deep forest does not take into account the either of one. As a result, standard deep forest will generate many noisy trees and features when training the high-dimensional data. Moreover, imbalanced data increases the bias towards the majority and brings out bad classification performance. In summary, it's necessary to exploit the advantage of deep forest and make more effective for gene expression data.

To solve the above-mentioned problem, we propose DPCAForest, based on deep forest and principle component analysis, to follow the mission of cancer classification based on gene expression data. The main idea of PCAForest is to encourage the performance for feature extraction and class imbalance. As a result, our model learns the augmented fusion features from muti-PCA and skip connections. The main contributions of the paper can be summarized as follows.

1) We propose DPCAForest, in which dynamic principle component analysis is intergrated with deep forest. The process of PCA is operated repeatedly synchronized with growth of deep forest to reveal the important features.

2375-0197/19/$31.00 ©2019 IEEE
DOI 10.1109/ICTAI.2019.00151

2) We preprocess the input data by adaptively generating the minority samples based on the sample distribution, alleviating the problem caused by class imbalance.
3) Through experimental verification, our proposed DPCAForest is superior to previous methods in Adenocarcinoma, Brain, Colon, SRBRTs and Nci-60 cancer datasets, showing the effectiveness on cancer classification.

The rest of the paper is organized as follows. In Section II, we review related work and existing methods. Section III describes the details of the proposed DPCAForest approach. In Section IV, the experimental results are presented with analysis. Finally, Section V concludes the paper.

II. RELATED WORK

A. Microarray Data

Microarray technology is one of the important biotechnological means that allows to record the expression levels of thousands of genes simultaneously within a number of different samples [4]. DNA microarray is usually used in biological and clinical research, and has been successfully applied to cancer diagnosis [5]. A microarray gene expression data set can be represented by an expression table, where each row corresponds to one particular gene, each column to a sample, and each entry of the matrix is the measured expression level of a particular gene in a sample, respectively [1]. From the perspective of machine learning, microarray data are some structural data and the genes are their features. However, gene expression data is hard to collect. For most gene expression data, the number of training samples is still very small compared to the large number of genes involved in the experiments. For example, a gene expression data about colon usually have less than a hundred samples but thousands of genes. When the number of genes is far greater than data samples, it's hard to use them for cancer diagnosis.

B. Informative Gene Selection

The greatest restraint in analyzing microarray data is that the number of genes is far bigger than the number of samples participating in the experiment [6]. Actually, most genes are noise genes and they have very limited affect to the classification. Informative genes are some key genes containing important information in phenotype identification of specific diseases and the classification methods will only use those genes.The process of informative gene selection is called feature selection.

Recently, various methods have been proposed for feature selection. Feature selection methods can be categorized into three techniques including the filter model [7], wrapper model and embedded model [8]. The filter model considers feature selection and classifier's learning as two separate steps and uses the general characteristics of training data to select important features. The filter model includes both traditional methods which often evaluate genes separately and new methods which consider the correlation between genes. These methods rank the genes and select top ranked genes as input features for the learning step. The gene ranking methods need a threshold for the number of genes to be selected. For example Golub et al. [9] proposed the selection of the top 50 genes.Examples of the filter criterion include Pearson correlation coefficient method [10], t-statistics method [11] and signal-to-noise ratio method [12]. The filter models can be efficient but they can not remove the redundant genes.Principal Component Analysis (PCA), a linear combination method is the one of representative methods in feature selection [13]. PCA reduces the dimension of microarray data by transforming the original data to a new coordinate system corresponding to the K largest eigenvalues.

C. Cancer Classification

In the literature, a variety of classification methods have been experimented to find the best classifier for cancer classification. SVM [14], the K-Nearest Neighbour [15], Random Forest methods are the most representative methods in this field. SVM proceeds by learning the linear decision rules, which are represented by hyper-planes. When SVM is applied to microarray data, the decision hyper-plane is easily affected by the poverty of training data, which makes the model overfitting. The K-Nearest Neighbor is an algorithm that classifies samples by selecting similar ones from the individual training data set of the new sample. This K-NN algorithm has the weakness of not providing good efficiency when granting equal weights to all genes.

Recently, the deep forest model has been proposed and proved to have great performance on many fields [2]. Fig 2 illustrates the basic architecture of the model from Zhou. Deep forest is a decision tree ensemble approach, and robust to hyperparameter settings with few hyperparameters. Guo Y [16] used deep forest model on microarray data for cancer classification. In the deep forest model, each layer of cascade assembles a number of decision tree forests, and receives features processed by its preceding, and inputs its processed results to the next layer. Two types of forests, completely random forest and standard random forest, are employed in the model, which encourage the diversity of the classifiers. Once the original feature vector is input into the model, each forest is trained independently and outputs a class distribution. The class distribution outputs by all forests in the same layer will form a class vector, and then concatenates with the original vector to be a new input vector to the next layer. Guo Y [16] proposed BCDForest based on deep forest on cancer classification. In the paper, Guo Y added top-k most important features in each forest for feature augmentation. However, two main limitations of the model will affect the performance. 1) No feature selection is performed on the original data. Noise trees and features will be generated by the model, which affects classification performance. 2) Class imbalance problem has not been solved thoroughly. The author simply trained several independent weighted binary classifiers to solve to class imbalance problem but this one-vs-all method does not consider the the distribution of the original sample and the

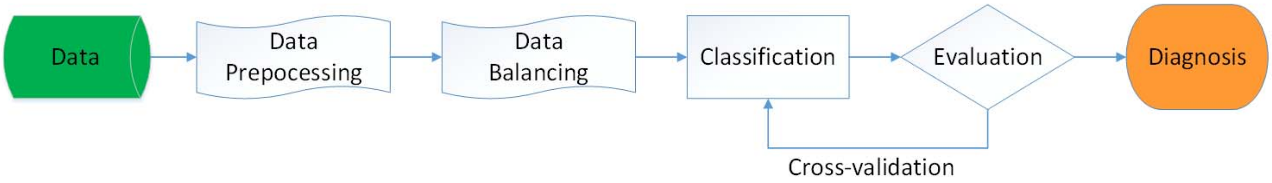

Fig. 1: General Pipeline

imbalance problem will still affect the final classification. To solve the problems mentioned above, our model is proposed.

III. DPCAForest:Novel Deep Forest Model Based on Dynamic PCA

The goal of our study is to build effective model for learning from high-dimensional and imbalanced microarray gene expression data. Fig 1 outlines the general pipeline and Fig 3 shows the framework of DPCADeep Forest. In our work, there are generally four steps. The first step is data preprocessing with normalization. Then the normalized data is processed with ADASYN method [17] to achieve class balance. Next is classification using DPCAForest. The final step is performance evaluation using accuracy and F-Measure.

A. Basic Structure

Our proposed model is based on deep forest. Deep forest employs a cascade structure, as illustrated in Fig. 2, where each level of cascade receives feature information processed by its preceding level, and outputs its processing result to the next level. Each level is an ensemble of decision tree forests, i.e., an ensemble of ensembles. Different types of forests are included to encourage the diversity of classifiers. As shown in the Fig. 2, two random forests and two completely-random forests are set and each forest contains 500 trees. Particularly, completely-random forests randomly select a feature for split at each node of the tree, and growing tree until pure leaf.

Given an instance, assume there are 100 raw features input of 3 classes. The instances will be used to trained two random forests and two completely-random forests independently, and then the class vectors are generated and concatenated as input vector for the next cascade layer. Besides, the raw feature vectors will be concatenated with the estimated class vectors, leading the next level of cascade to receive 12 (3×4) augmented features. Thus, we call it the cross-layer connection. This approach effectively fuses features from different cascade levels and the model will have strong feature extraction capabilities. In this study, we harness the basic structure and continue to enrich these augmented features, as to improve the model performance.

B. Dynamic Feature Extraction based Principle Component Analysis

High dimensionality is the major cause of the practical limitations for cancer classification on microarray data sets. Once the dimension increases, the complexity of data distribution increases rapidly. Therefor, feature extraction is essential

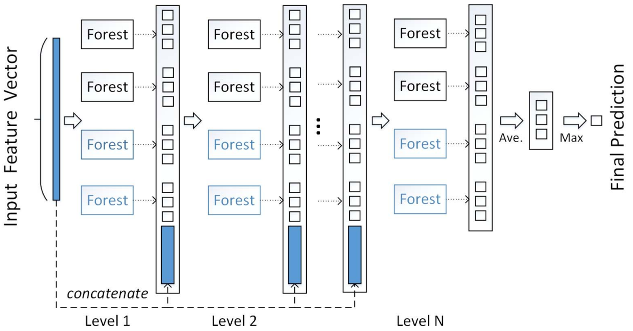

Fig. 2: Illustration of the cascade forest structure. Suppose each level of the cascade consists of two random forests (black) and two completely-random tree forests (blue). Suppose there are three classes to predict; thus, each forest will output a three-dimen-sional class vector, which is then concatenated for re-representation of the original input.

Algorithm 1 Classification Algorithm

Input: set of raw data X_{raw}, number of classes n_c, number of principle component K, early stop rounds R_{es}, the accuracy tolerance tlc, number of basic classifiers N_{bc}, type of basic classifiers e.g. RF, hyperparameters of basic classifier e.g. number of decision trees, size of sliding windows.
Output: the prediction of class.

```
Standardize the raw data $X_{raw}$.
Use **Algorithm 3** to verify data distribution of $X_{raw}$ and generate new sets of data samples based on the algorithm.
**while** $R_i \leq R_{es}$ **do**
  Build the cascade
  **for** $i = 1 : N_{bc} - 1$ **do**
    Train $Forest^i$ and get the class estimate vector $output^i_j$, $j = 1, 2, ..., n_c$ .
  **end for**
  Run **Algorithm 2** to train $DPCAForest.$
  Concatenate class estimates vectors of each Forest in the same cascade layer.
  Concatenate the concatenated class estimate vector with the original feature vector.
  Verify the current cascade model by cross-validation to determine whether to end the loop.
**end while**
Average the estimates of class distribution, $Y_i$.
Output the prediction : Max($Y_i$), $i = 1, 2, ..., n_c$ .
```

TABLE I: The Key Notations

Notation	Definition
R_{es}	the hyperparameter of early stop rounds
R_i	the round of current cascade layer
X^i	the i-th data sample. $i = 1, 2, 3, ..., M$
x^i_j	the j-th feature of i-th data sample. $j = 1, 2, 3, ..., N$
C	the covariance matrix C
$cov(x_1, x_2)$	the covariance between x_1 and x_2.
λ	the eigenvalues of the covariance matrix
ν	the eigenvectors of the covariance matrix
$I_{m,n}$	the raw input data of $m \times n$
$X_{m,n}$	the standardized data of $m \times n$
tlc	the hyperparameter to determine the accuracy $tolerance$ of whether the cascade layer is growing

for classifier design. Principle component analysis (PCA) is given by a linear transformation matrix minimizing the mean square error criterion [18]. In PCA, the matrix is constituted by the largest eigenvectors (called principal components) of the sample covariance (or correlation) matrix. The purpose of PCA is to keep the information in terms of variance as much as possible.

For feature extraction and dimensionality reduction, the usual practice is to perform PCA only once. We come up with a novel idea of whether to perform PCA more than once to extract deeper features. Coincidentally, the cross-layer connections of deep forest provides the possibility of our idea. It is because the output of each cascade is connected to the original input vector, so the number of features will not reduce by the growth of the cascade. At the same time, with the cascade continues to grow, more new and stronger features will be discovered. In this process, PCA is to mine deeper features and augment the existing features. Specifically, we add a new classifier to the deep forest model, namely dynamic PCA random forest. The feature extraction is operated repeatedly synchronized with growth of deep forest. In other words, the number of PCA runs depends on number of cascade levels in deep forest. After expanding a new level, the performance of the whole cascade will be estimated on validation set, and the training procedure will terminate if there is no significant performance gain. Thus, the proposed model is robust to overfitting and can adaptively decides its model complexity. The key notations are listed in table I and the detailed dynamic feature extraction is described in Algorithm 2.

C. The Framework of DPCAForest

The main framework of DPCAForest is shown in Fig 3. Our contributions could be summarized as follows. (1) We propose PCA-Forest (red), in which the input feature vector is processed by dynamic PCA and classified by standard random forest, and merge it into the model. The feature extraction process is operated dynamically synchronized with growth of cascade layer in the model. (B) We select K features with the highest variance, which are calculated by PCA, and concatenate them with the original output vectors to augment the informative features. (C) We adaptively generate the minority samples on the original input feature vectors based on the data distribution. The specific process is shown in Algorithm 3.

Algorithm 2 Dynamic Feature Extraction Based on PCA

Input: the raw data vector $X_{m,n}$, number of principle components K.

Output: K-dimension transformed feature vector: $Y_{m,k}$.

```
while $R_i \le R_{es}$ do
    Standardize all features of $X^i$. $i = 1, 2, 3, ..., M$.
        $x_j = x_j - \overline{x_j}$, $j = 1, 2, 3, ..., N$
    Compute the covariance matrix $C$.
        $C = \begin{bmatrix} cov(x_1, x_1) & cov(x_1, x_2) \\ cov(x_2, x_1) & cov(x_2, x_2) \end{bmatrix}$
    , where
        $cov(x_1, x_1) = \frac{\sum_{i=1}^{M}(x^i_1 - \overline{x_1}) \times (x^i_2 - \overline{x_2})}{M - 1}$
    Calculate $\lambda$ and $\nu$ of the covariance matrix $C$. Sort the eigenvalues $\lambda$ in descending order and select the largest top $K$.
        $C\nu = \lambda\nu$
    Project the original features $X$ to the new $K$-dimensional feature $Y$. For sample $X_i$, the original feature is $(x^i_1, x^i_2, ..., x^i_n)^T$, the projected new feature is $(y^i_1, y^i_2, ..., y^i_k)^T$. The formular is as follows :
        $\begin{bmatrix} y^i_1 \\ y^i_2 \\ \cdot \\ \cdot \\ y^i_k \end{bmatrix} = \begin{bmatrix} \nu_1^T \cdot (x^i_1, x^i_2, ..., x^i_n)^T \\ \nu_2^T \cdot (x^i_1, x^i_2, ..., x^i_n)^T \\ \cdot \\ \cdot \\ \nu_k^T \cdot (x^i_1, x^i_2, ..., x^i_n)^T \end{bmatrix}$
    Train the classifier and verify the performance using cross-validation.
    if $\Delta$result< $tlc$ then
        End the loop and stop growing the cascade layer.
    else
        $R_i = R_i + 1$;
    end if
end while
```

Specifically, we set up two complete-random forests, two random forests and one PCA-forest at each layer of cascade, and 500 decision trees in each forest. To ease the risk of overfitting, we use the built-in 5-fold cross validation in each cascade layer to separate the data training and testing. The propagation of cascade will be automatically terminated once the performance turns to decrease. After training, the model could be used for estimating classes of new instances. The final pseudo-code of framework of DPCAForest is shown in Algorithm 1.

Algorithm 3 Adaptive Synthetic Sampling Algorithm

```
Require: Training data set $X$ with $m$ samples and $n$ dimensions. Define $m_s$ and $m_l$ as the number of minority and majority class samples. There, $m_s \leq m_l$ and $m_s + m_l = m$.
Ensure: Synthetic data sample $S$
Calculate the degree of class imbalance:
    $$d = m_s/m_l \quad (1)$$
    where $d \in$ (0,1].
if $d < d_{th}$ then
    Calculate the number of minority data samples that need to be synthesized :
        $$G = (m_l - m_s \times \beta) \quad (2)$$
        where $\beta \in$ [0,1] is a parameter used to specify the desired balance level after generation of the synthetic data. $\beta = 1$ means the full balance.
    For each example $x_i \in$ minority class, find their $K$ nearest neighbors based on the Euclidean distance in $n$ dimensional space, and calculate the ratio $r_i$ :
        $$r_i = \Delta_i/K, \, i = 1, ..., m_s \quad (3)$$
        where $\Delta_i$ is the number of minority samples in the $K$ nearest neighbours of $x_i$. Therefor, $r_i \in$ [0,1].
    Normalize $r_i$ according to $\widehat{r_i} = r_i / \sum_{i=1}^{m_s} r_i$, so that $\widehat{r_i}$ is a density distribution.
    Calculate the number of synthetic data examples to be generated for each minority example $x_i$:
        $$g_i = \widehat{r_i} \times G \quad (4)$$
        $G$ is the total number of synthetic data examples to be generated, defined in Equation 2.
    for $i = 1 : g_i$ do
        Randomly choose one minority data sample $x_{zi}$ in $K$ nearest neighbors from data $x_i$;
        Generate the synthetic data sample based on :
            $$s_i = x_i + (x_{zi} - x_i) \times \lambda \quad (5)$$
            where $(x_{zi} - x_i)$ is the difference vector in $n$ dimensional spaces, and $\lambda$ is a random number. $\lambda \in$ [0,1].
    end for
end if
```

IV. Experiment

A. Experiment Setup

1) Data sets: In this paper, the data sets are gathered from a small identified data set of a real-world microarray gene expression data, collected by [19]. Colon tumor, Adenocarcinoma, SRBRTs (Small Round Blue Cell Tumors) and Nci-60 cancer data sets are used, shown in Table II. From the perspective of machine learning, a gene can be seen as a feature of a data sample. Consequently, there are type of data sets whose feature numbers are much larger than the numbers of samples. The $Ratios$ represents the ratio of the number of samples in the data sets and we can see that there is a serious imbalance problem especially on the Adenocarcinoma data set.

TABLE II: Characteristics of data sets

Data sets	Samples	Genes	Classes	Ratios
Adenocarcinoma	76	9868	2	5.3:1
Colon	62	2000	2	1.82:1
Brain	42	5597	5	2.5:2.5:2.5:2:1
SRBRTs	63	2308	4	2.875:2.5:1.5:1
Nci-60	61	5244	8	-

2) Evaluating Metrics: We adopt two metrics to test the performance of our algorithm: **Accuracy**, and **F-Measure**. The Accuracy is defined as $accuracy = \frac{TP+TN}{TP+TN+FP+FN}$. The F-Measure is defined as $\text{F}-\text{measure} = \frac{2 \times Recall \times Precision}{Recall + Precision}$, where $Precision = \frac{TP}{TP+FP}$, $Recall = \frac{TP}{TP+FN}$, TP/FP/TN/FN means true positive/false positive/true negative/false negative.

Accuracy is the most commonly used metric to measure classifier performance. Besides, *precision* measures the accuracy of a classifier, while the metric of *recall* measures the completeness of a classifier. Hence, F-Measure is a comprehensive performance assessments to evaluate the robustness considering both precision and recall.

3) Compared Algorithms: We compare the performance of our model with multiple machine learning classifiers. The detailed algorithms and hyperparameter setting are as follows.

1) LR: Logistic Regression with $l1$ norm. Use $liblinear$ as solver and set parameter class_weight to $balanced.$
2) SVM: Support Vector Machine with $linear$ kernel.
3) RF: Random Forest with $CART$ decision tree. 500 decision trees and $sqrt$ for max_features are set.
4) gcForest: Standard Deep Forest with Multi-Grained Scanning. Two random forests and two completely-random forests are set. For each forest, 500 trees and $sqrt$ for max_features are set. 100-dimension sliding window size is used.
5) BCDForest: Boosting Cascade Deep Forest. The number of binary classifiers trained for class imbalance depends on number of classes. The k, which denotes the number of selected highest variance is 10.
6) the proposed **DPCAForest**. The setting of our algorithm is similar to gcForest. The k is automatically determined by dynamic PCA.

4) Training tricks: We apply a regularization strategy on data labels, called $label\ smoothing$. This technique softens the one-hotted labels that effectively suppresses over-fitting that

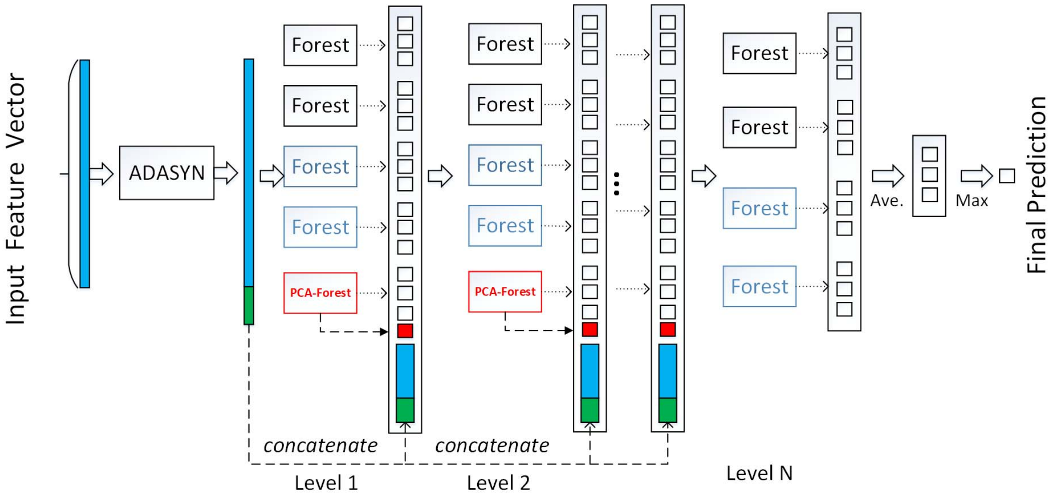

Fig. 3: Main framework of DPCAForest. (A) We propose PCA-Forest (red), in which the input feature vector is processed by dynamic PCA and classified by standard random forest, and merge it into the model. The feature extraction process is operated dynamically synchronized with growth of cascade layer in the model. (B) We additionally select K features with the highest variance, which are calculated by PCA, and concatenate them with the original output vectors to augment the informative features. (C) We adaptively generate the minority samples (green) on the original input feature vectors based on the data distribution. The generated samples are labeled in green.

may occur due to lack of data. The hyperparameter $epsilon$ is set to 0.1.

B. Results and analysis

Table III shows the overall accuracy of each data set. The best performance in each data set is highlighted as bold. As one can perceive from the results, our proposed method shows a consistently better performance than others. On Adenocarcinoma and SRBCTs data set, our model achieves 98.6% and 99.2% on accuracy.

TABLE III: COMPARISON OF OVERALL ACCURACY

Data sets	LR	RF	SVM	gcForest	BCDForest	DPCAForest
Adenoca-rcinoma	0.729	0.861	0.853	0.868	0.931	**0.986**
Colon	0.673	0.851	0.872	0.921	0.918	**0.938**
Brain	0.846	0.803	0.701	0.907	0.958	**0.963**
SRBCTs	0.756	0.927	0.883	0.924	0.938	**0.992**
Nci-60	0.513	0.468	0.775	0.542	0.771	**0.776**

In order to evaluate the robustness of DPCAForest for the imbalance problem on gene expression data sets, we run algorithms and retrieve F-Measures. F-Measure is a comprehensive performance assessment in imbalanced learning considering both precision and recall. As we can see, except for the Brain data set, our model gets the best performance. The results are listed in Tablel IV.

TABLE IV: Comparison of F-Measures

Data sets	F-Measure					
	LR	RF	SVM	gcForest	BCDForest	DPCAForest
Adenoca-rcinoma	0.797	0.807	0.765	0.886	0.917	**0.944**
Colon	0.893	0.886	0.844	0.906	0.914	**0.925**
Brain	0.853	0.803	0.710	0.885	**0.971**	0.968
SRBCTs	0.853	0.929	0.847	0.948	0.956	**0.992**
Nci-60	0.562	0.646	0.583	0.716	0.772	**0.801**

The results illustrate that our method is effective in the mission of cancer classification on small-sample gene expression data. We can also see that deep-forest-based method including gcForest, BCDForest and DPCAForest outperform other conventional supervised learning methods in this field. In other words, the experiments show that our model have better performance on data dimension reduction and informative genes selection than others on small-sample and high-dimensional gene expression data sets.

V. Conclusion

In this paper, we proposed a deep forest based model, so-called DPCAForest, to follow the mission of cancer classification on small-sample gene expression data sets. DPCAForest adaptively generates the minority samples based on sample distribution, then conducts principle component analysis, which is dynamically synchronized with growth of deep forest to reveal the important features with the highest variance. Dynamic PCA enables the model to perform feature extraction in a data-driven way based on cross-validation, and the model can obtain fusion information across layers. We compared DPCAForest with original gcForest, BCDForest and several supervised learning methods on five microarray cancer datasets. Experimental results show that the DPCAForest consistently outperformed state-of-the-art methods in terms of accuracy and F-Measure on most of cancer data sets. In conclusion, our method provides an option to investigate cancer diagnosis by using deep forest based methods on small-scale biology data sets.

Acknowledgment

This work was supported by the Fundamental Research Funds for the Central Universities of Central South University grant No.2019zzts588 and the National Natural Science Foundation of China under grant No.61772553, 61379058.

References

[1] D. V. Nguyen and D. M. Rocke, “Multi-class cancer classification via partial least squares with gene expression profiles,” *Bioinformatics*, vol. 18, no. 9, pp. 1216–1226, 2002.

[2] Z. H. Zhou and J. Feng, “Deep forest,” 2017. [Online]. Available: https://arxiv.org/pdf/1702.08835.pdf

[3] G. Yang, S. Liu, Z. Li, and X. Shang, “Towards the classification of cancer subtypes by using cascade deep forest model in gene expression data,” in *IEEE International Conference on Bioinformatics & Biomedicine*, 2017.

[4] P. Maji, *Mutual Information-Based Supervised Attribute Clustering for Microarray Sample Classification*, 2012.

[15] P. Meesad and K. Hengpraprohm, “Combination of knn-based feature selection and knnbased missing-value imputation of microarray data,” in *International Conference on Innovative Computing Information & Control*, 2008.

[5] T. Z and J. W, “Multiclass microarray data classification based on saecoc,” 2017.

[6] S. Dudoit, J. Fridlyand, and T. P. Speed, “Comparison of discrimination methods for the classification of tumors using gene expression data,” *Publications of the American Statistical Association*, vol. 97, no. 457, pp. 77–87, 2002.

[7] L. Y. Sun X and X. M, “Feature selection using dynamic weights for classification,” *Knowledge-Based Systems*, vol. 37, no. 457, pp. 541–549, 2013.

[8] S. Ghorai, A. Mukherjee, and P. K. Dutta, “Gene expression data classification by vvrkfa,” *Procedia Technology*, vol. 4, pp. 330–335, 2012.

[9] R. Alexandridis, S. Lin, and M. Irwin, “Class discovery and classification of tumor samples using mixture modeling of gene expression data–a unified approach.” *Bioinformatics*, vol. 20, no. 16, pp. 2545–52, 2004.

[10] . Xiong, M., . Fang, X., and . Zhao, J., “Biomarker identification by feature wrappers,” *Genome Research*, vol. 11, no. 11, pp. 1878–1887, 2001.

[11] P. Baldi and A. D. Long, “A bayesian framework for the analysis of microarray expression data: regularized t -test and statistical inferences of gene changes,” *Bioinformatics*, vol. 2012, no. 2, pp. 132–148, 2001.

[12] D. K. Slonim, P. Tamayo, J. P. Mesirov, T. R. Golub, and E. S. Lander, “Class prediction and discovery using gene expression data,” in *Proceedings of the Fourth Annual International Conference on Computational Molecular Biology*, ser. RECOMB ’00, 2000.

[13] W. Astuti and Adiwijaya, “Support vector machine and principal component analysis for microarray data classification,” *Journal of Physics: Conference Series*, vol. 971, p. 012003, mar 2018.

[14] S. Maldonado and J. Lpez, “Dealing with high-dimensional class-imbalanced datasets,” *Appl. Soft Comput.*

[16] Y. Guo, S. Liu, Z. Li, and X. Shang, “Bcdforest: a boosting cascade deep forest model towards the classification of cancer subtypes based on gene expression data,” *Bmc Bioinformatics*, vol. 19, no. Suppl 5, p. 118, 2018.

[17] H. He, B. Yang, E. A. Garcia, and S. Li, “Adasyn: Adaptive synthetic sampling approach for imbalanced learning,” in *IEEE International Joint Conference on Neural Networks*, 2008.

[18] I. T. Jolliffe, “Principal component analysis,” *Journal of Marketing Research*, vol. 87, no. 100, p. 513, 2002.

[19] K. Moorthy and M. S. Mohamad, “Random forest for gene selection and microarray data classification,” *Bioinformation*, vol. 7, no. 3, p. 142, 2011.