IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 9, 2018, accepted May 6, 2018, date of publication May 21, 2018, date of current version June 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2837654

Integrating Feature Selection and Feature Extraction Methods With Deep Learning to Predict Clinical Outcome of Breast Cancer

DEJUN ZHANG[1], LU ZOU[1], XIONGHUI ZHOU[2], AND FAZHI HE[3, 4]

[1]College of Information and Engineering, Sichuan Agricultural University, Yaan 0086-625014, China
[2]College of Informatics, Huazhong Agricultural University, Wuhan 0086-430070, China
[3]School of Computer, Wuhan University, Wuhan 0086-430072, China
[4]State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 0086-430074, China

Corresponding author: Dejun Zhang (djz@sicau.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61702350 and Grant 61472289 and in part by the Open Project Program of the State Key Laboratory of Digital Manufacturing Equipment and Technology, HUST, under Grant DMETKF2017016.

ABSTRACT In many microarray studies, classifiers have been constructed based on gene signatures to predict clinical outcomes for various cancer sufferers. However, signatures originating from different studies often suffer from poor robustness when used in the classification of data sets independent from which they were generated from. In this paper, we present an unsupervised feature learning framework by integrating a principal component analysis algorithm and autoencoder neural network to identify different characteristics from gene expression profiles. As the foundation for the obtained features, an ensemble classifier based on the AdaBoost algorithm (PCA-AE-Ada) was constructed to predict clinical outcomes in breast cancer. During the experiments, we established an additional classifier with the same classifier learning strategy (PCA-Ada) in order to perform as a baseline to the proposed method, where the only difference is the training inputs. The area under the receiver operating characteristic curve index, Matthews correlation coefficient index, accuracy, and other evaluation parameters of the proposed method were tested on several independent breast cancer data sets and compared with representative gene signature-based algorithms including the baseline method. Experimental results demonstrate that the proposed method using deep learning techniques performs better than others.

INDEX TERMS Cancer prognosis, ensemble classifier, principal component analysis, deep learning.

I. INTRODUCTION

Identifying small feature sets which effectively characterize different disease states is an important application of genome-wide expression data analysis [1]. In breast cancer, patients with the same disease status can have obviously different treatment responses and overall outcomes. The strongest predictors like histological grade and lymph node status for metastasis still fail to accurately identify breast tumors according to their clinical manifestations. It is reported that the risk of distant metastases can be reduced by chemotherapy or hormonal therapy; however, more than 70% of patients receiving this treatment would have survived without this treatment in any case, and none of the currently reported methods allow for patient-tailored therapy strategies [2].

Recently, many methods have been proposed to classify cancer sub-phenotypes into different risk groups in order to ensure cancer sufferers receive befitting therapy. Most of the classifiers perform feature space reduction by deriving compact features via the selection or extraction of features in a supervised or unsupervised manner [2]–[5]. However, the performance of these classifiers is generally not scalable and usually declines sharply when used on datasets distinct to those used for classifier construction. For instance, two recent large scale gene expression profile studies respectively picked out a signature consisting of 70 genes [2] and another signature consisting of 76 genes [5] for predicting distant metastasis in breast cancer sufferers. Both of these studies achieved classification accuracy of 0.7 [6] on their own patient cohorts. However, when each method was applied to the other's dataset, they performed poorly, with accuracy of less than 0.55 [1].

We argue that there are two fundamental reasons why classifiers based on chosen gene signatures are so unstable and study-independent. Firstly, due to heterogeneity of gene

2169-3536 © 2018 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

expression data, the detected gene signatures in prognosis features play role as ''passengers'' instead of ''drivers'' of the phenotypic differences, resulting in a large number of passenger signals being involved in the expression profiles of tumor cells [1], [7]. Secondly, the proper performance of conventional classifiers relies heavily on handcrafted features, and identifying the most appropriate features for the given task remains difficult.

Recently, deep learning for many challenging machine learning problems have achieved great performance with respect to learning hierarchical nonlinear patterns from large scale datasets [8]–[10]. In general, the term ''deep learning'' is used in reference to learning a hierarchical representation of the data through multiple layers of abstraction (e.g. multilayer feed-forward neural networks). Nowadays, quite a few new techniques have been developed in deep learning, such as the deployment of general-purpose computing on graphics processing units [11], [12], and new training methodologies [13], [14]. With these advances, deep learning has demonstrated state-of-the-art performance in a wide range of applications, both in traditional machine learning tasks such as computer vision [15], speech recognition [16] and natural language processing [17], and in natural science applications such as RNA binding site prediction [18], protein secondary structure prediction [19] and pathogenic variants annotation [20]. While neural networks (e.g. the stacked autoencoder (SAE), the deep belief network (DBN), and the multi-layer perceptron (MLP)) have been successfully used in bioinformatics [18], [21]–[26], to the best of our knowledge, deep learning has not been employed for predicting clinical outcomes for cancer sufferers.

The goal of this paper is to enhance the performance in cancer prognosis prediction and develop a more generalized outcome classifier. To achieve this, (i) we propose a more general way of learning features by integrating feature selection and feature extraction methods [27] with several deep learning techniques [28], [29]. (ii) We construct an ensemble classifier with a boosting algorithm [30] to strongly predict distant metastasis in breast cancer. (iii) Compared with previous classifier learning approaches, the method proposed in this paper demonstrates an unsupervised feature learning and supervised classifier learning mechanism.

The rest of this paper is organized in the following manner. Section II briefly reviews the feature learning methods we employed in this paper, including principal component analysis algorithm and the autoencoder neural network. Section III outlines the overall framework of our method for predicting cancer outcomes. In Section IV, we give out some implementation details such as parameter learning methods and classifier training skills. In Section V, we discuss the evaluation results and compare them with those of representative classifiers.

II. BACKGROUND

Many different feature selection and feature extraction methods exist and are being widely used to perform dimensionality reduction for high-dimensional microarray data [31], [32]. Intuitively, all these methods consider to eliminate redundant and irrelevant information so that the classification of novel test cases will be more accurate [33]. In this section, we briefly review principal component analysis (PCA) algorithm and the autoencoder neural network used for dimensionality reduction (also called feature learning) in this paper.

A. PRINCIPAL COMPONENT ANALYSIS

PCA or Karhunen Love transform, is a multivariate technique which is said to be one of the most popular methods for linear dimensionality reduction [27]. Using the covariance matrix and its eigenvalues and eigenvectors, PCA finds the principal components in data which are uncorrelated eigenvectors, each representing some proportion of variance in the data [34].

Let $X = \{x_i\}_{i=1}^{m}$ denote a set of training data. x_i represents a variable with dimensionality D, which stands for the gene expression profiles in this paper. The aims of PCA are summarized as: (a) to extract the most important information from x_m; and (b) to compress the dimensionality of X by keeping the important information only. PCA is regarded as an orthogonal projection of the original D-dimensional data onto a new k-dimensional space ($k < D$), the objective to be minimized is the variance of the projected data as illustrated in Fig. 1.

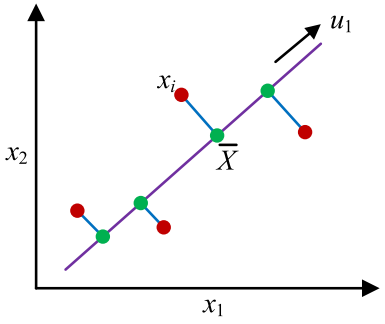


FIGURE 1. The process of orthogonal projection. PCA searches a space with smaller dimensionality, denoted as the principal subspace and indicated by the magenta line. Thus the orthogonal projection of the original data points (red dots) onto this subspace maximizes the variance of the projected points (green dots). An alternative definition of PCA is based on minimizing the sum-of-squares errors mapping the original points to the projected ones, as indicated by the blue lines.

Assume the direction of the projection space using a vector u_1 (with dimensionality of D). Then each data point x_i is projected onto a scalar value defined by $u_1^T x_i$. Next, let the mean of the projected data be equal to $u_1^T \overline{X}$, where $\overline{X}$ denotes the mean of sample set given by $\overline{X} = \frac{1}{m}\sum_{i=1}^{m} x_i$. Finally, the variance of the projected data will be given by: $\frac{1}{m}\sum_{i=1}^{m}(u_1^T x_i - u_1^T \overline{X})^2 = u_1^T S u_1$, where S denotes a common covariance matrix for all samples: $S = \frac{1}{m}\sum_{i=1}^{m}(x_i - \overline{X})(x_i - \overline{X})^T$. Now, the objective of PCA is to maximize the projected variance $u_1^T S u_1$ with respect to u_1. See more details in the literature [27].

B. AUTOENCODER NEURAL NETWORK

In this section we give a brief description about the autoencoder neural network, which is a nonlinear dimensionality reduction method and is famous for feature extraction.

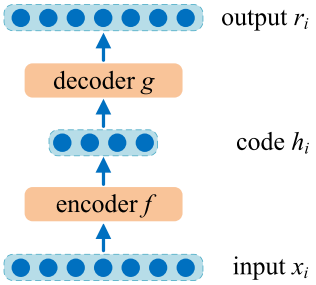


FIGURE 2. General architecture of an autoencoder, mapping the input x_i to the output r_i (called reconstruction) through a code or hidden representation h_i. The autoencoder has two components: the encoder f (mapping x_i to h_i) and the decoder g (mapping h_i to r_i).

As illustrated in Fig. 2, an autoencoder is a feed-forward neural network which is often trained to learn representations or effective encoding of the original data $X = \{x_i\}_{i=1}^{m}$. In this way, it learns a function $g\big(f(x_i)\big) \approx r_i$ that approximately represents the input data constructed from a limited number of feature activations and represented by the hidden units of the network.

Min *et al.* [35] compared stacked autoencoder with PCA and Gaussian SVM over 13 gene expression datasets, their results declared that the autoencoder performs better for the majority of the datasets, which also motivates us in this work. As per the literature [36], the general architecture of an autoencoder is divided into the following parts: (1) the input units x_i; (2) a encoder function f; (3) a "code" or hidden representation $h_i = f(x_i)$; (4) a decoder function g; (5) the output units also called "reconstruction" $r_i = g(h_i) = g\big(f(x_i)\big)$; and (6) a loss function $\mathcal{L}(x_i, r_i)$ computing a scalar $\|x_i - r_i\|_2$ which measures how good the reconstruction r_i is of the original input x_i. The optimization objective of the autoencoder is to minimize the expected values of $\mathcal{L}$ over the training examples X.

During each training iteration, the difference between input and output is measured using square error, and back-propagation will be performed through the neural network to perform the weight updates to different layers. If the number of hidden layer is greater than one, the autoencoder is considered to be deep, and the encoder function implements nonlinear dimensionality reduction. When the dimensionality of the hidden layer is less than the dimensionality of the inputs, the autoencoder is trained to find the best feature compression of the inputs on the hidden layers. Otherwise, the autoencoder is trained to map the feature to a higher-dimensional space.

III. METHODS

In this section, we introduce the proposed method in this work. First, we describe five gene expression datasets we used and perform some preprocessing on them. Then, we present our approach for the given problem.

A. GENE EXPRESSION DATASETS AND PREPROCESSING

The gene expression data were downloaded from the publicly available NCBI GEO database. Each sample comprises of 129,158 gene expression profiles from the Affymetrix microarray platform and each profile consists of 22,268 probes, with respect to 978 landmark genes and the 21,290 target genes. Specifically, we obtained five different breast cancer datasets from the LINCS Cloud[1] to evaluate the feasibility and applicability of the proposed method. All the five datasets were normalized by its original authors using algorithm MAS5.0, except for GSE4922, which was normalized using algorithm RMA. Table 1 shows the information in detail.

TABLE 1. Breast cancer datasets. Samples were removed if patients had been censored within 5-year or had received adjuvant treatment.

Data	Poor outcome	Good outcome	Total	Removed samples
GSE2034	93	183	276	10
GSE4922	30	103	133	156
GSE6532	23	77	100	227
GSE7390	36	154	190	8
GSE11121	28	154	182	18

For the five datasets, patients perform different immune and pathological parameters in order to influence the outcome prognosis. For example, the patients in GSE6532 are all *ER*-positive, while other datasets contain both patients with *ER*-positive and *ER*-negative. Datasets GSE4922 and GSE6532 contain both patients with lymph node-positive and lymph node-negative, while there are only lymph node-negative patients in the other datasets.

Considering the given classification task, we performed two-step preprocessing with the five datasets: Firstly, we follow the dataset partitioning scheme in [37], all cancer patients were divided into poor prognosis (set label as 1) and good prognosis groups (set label as 0) according to whether distant metastasis had occurred within 5-year or not. The follow-up information of patients who had been censored within 5-year or had received adjuvant treatment were removed from consideration. Secondly, since the gene expression values of the microarray platform were measured with different algorithms (MAS 5.0 or RMA), we quantiled normalized the five datasets with the algorithm MAS 5.0, and all the probes were mapped into Entrez Gene ID and averaged.

B. OVERALL APPROACH

In this paper, we aim to combine both feature selection and feature extraction methods with deep learning techniques to learn more representative characteristics from gene expression profiles, and construct a more powerful classifier for cancer prognosis prediction. Fig. 3 illustrates the flowchart of our approach.

[1] http://www.lincscloud.org/http://www.lincscloud.org/

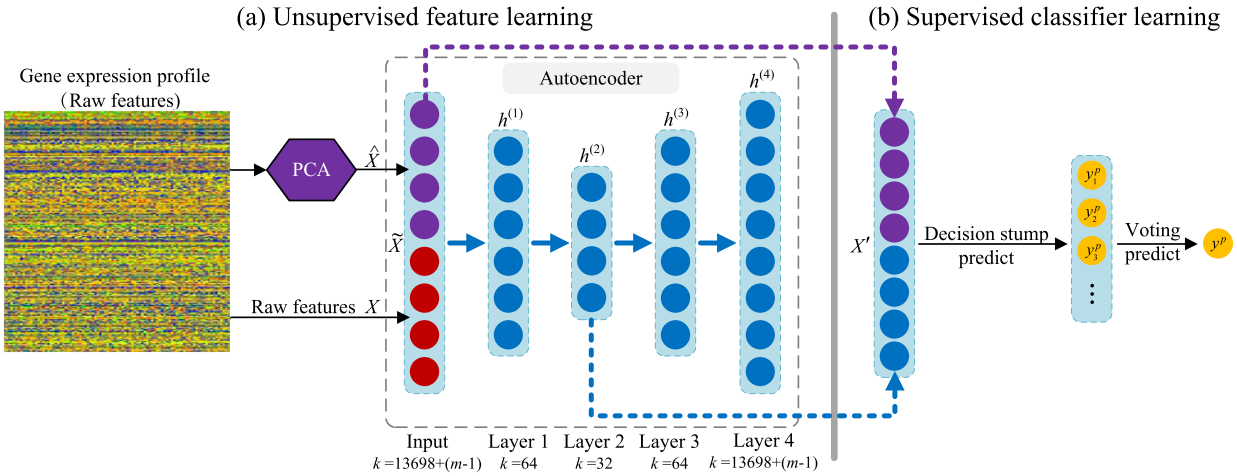


FIGURE 3. The flowchart of the proposed method. Our classification task is composed of two phases: (a) unsupervised feature learning, (b) supervised classifier learning. In the unsupervised feature learning phase, PCA algorithm and deep neural network are applied to learn concise features from gene expression profiles, while in the classifier learning phase, an ensemble classifier is constructed according to the features learned before. Note that the last encoding layer and the first decoding layer are sharing with the same parameters.

1) UNSUPERVISED FEATURE LEARNING

Motivated by the feature learning strategy demonstrated in [25], our feature learning approach is composed of two phases.

- *PCA*. Since the dimensionality of gene expression data is extremely high, and these contain redundant and noisy data, we employ PCA (as described in Section II-A) to act as the feature selection method to reduce the dimensionality of the gene expression profiles. PCA performs a linear approximation of the original data, and retains significant information in the meanwhile.
- *Autoencoder*. The resulting features after applying PCA are simply a linear function of the original data. Therefore, in order to also capture non-linear associations among expressions of different genes, an augmented form of the PCA features in addition to raw features are subsequently fed into a feature extraction architecture to learn high-level and complex features for use in the following classification approach. We employ an autoencoder neural network for the feature extraction purpose, and the configuration details are shown in Sections II-B)

As mentioned before, both of the two phases for feature learning are independent of data labels, which shows unsupervised feature learning.

2) SUPERVISED CLASSIFIER LEARNING

In order to perform the task of predicting clinical outcome for cancer patients, the features generated from the proposed two-phase unsupervised feature learning approach are subsequently appended with a set of labels for classifier learning. In this paper, we take a variant of AdaBoost algorithm [30] which demonstrates excellent performance in the classification tasks, as the learning approach for the classifier. We name this classifier in terms of the PCA-AE-Ada (see details for classifier construction in Section IV). The procedure of classifier learning is dependent on sample labels, which shows supervised classifier learning.

The main components of our proposed method for predicting clinical outcome of breast cancer patients are shown as follows.

- Firstly, given a set of gene expression profiles $X = \{x_i\}_{i=1}^{m}$;
- Secondly, principal component analysis is employed to learn compressed feature sets $\hat{X} = \{\hat{x}_i\}_{i=1}^{m}$, with respect to $\frac{1}{m}\sum_{i=1}^{m}(u_1^T x_i - u_1^T \overline{X})^2 = u_1^T S u_1$;
- Thirdly, raw gene expression data X and compressed features $\hat{X}$ are merged into $\tilde{X}$, where $\tilde{X} = \{\tilde{x}_i | \tilde{x}_i = (\hat{x}_i, x_i)\}_{i=1}^{m}$. And $\tilde{X}$ are taken as the inputs of autoencoder neural network in order to learn more complex representation $h^{(2)}$ using deep learning techniques.
- Finally, the compressed features $\hat{X}$ and deep representations $h^{(2)}$ are concatenated to a comprehensive manner X' that is used for training an ensemble classifier.

Additionally, as a comparison in the evaluation experiments executed in this paper, a baseline classifier named PCA-Ada using PCA compressed results as input features is constructed to contrast with the classifier constructed with features generated from the two-step feature learning framework.

IV. IMPLEMENTATION DETAILS

A. DATA ALIGNMENT

On the one hand, when the PCA algorithm is applied to gene expression data for dimensionality reduction, we can obtain compressed feature vectors with different dimensions due to the fact that the size of each dataset is different, and this is incompatible with the input size of the feature extraction neural network. For sake of adapting the model we constructed to data with different dimensions, we padded all the feature

vectors with values of zero to force the feature sets in the same dimension without causing performance damage.

On the other hand, our PCA-AE-Ada model contains feature dimensionality of $(m-1)+32$, while the baseline model PCA-Ada contains a different feature dimensionality of $m-1$, which implicitly indicates these two methods are faced with different hyperparameter configurations. Therefore, in order to perform a fairer comparison, we randomly added 32 raw gene expression data features to the input of the PCA-Ada, which forces these two methods to a more comparative configurations without increasing redundant information.

B. OBJECTIVE FUNCTION FOR FEATURE EXTRACTION

For the second phase of the proposed feature learning approach, we utilized the stacked autoencoder[2] to create a deep neural network by stacking multiple autoencoders hierachically.

1) NON-LINEAR TRANSFORMATION

Taking the merged representations of the original data $\tilde{X}$ as input, the encoder and decoder parts of the autoencoder consist of several non-linear transformation layers as follows:

$$\begin{aligned} h^{(1)} &= \sigma(\omega^{(1)}\tilde{X} + b^{(1)}), \\ h^{(j)} &= \sigma(\omega^{(j)}h^{(j-1)} + b^{(j)}), \quad j = 2, \ldots, n. \end{aligned} \tag{1}$$

Here, n denotes the number of layers and σ denotes the activation function. $h^{(j)}$, $\omega^{(j)}$ and $b^{(j)}$ denote the hidden vector, weight matrix and bias vector in the j-th layer respectively.

2) RECONSTRUCTION LOSS

Autoencoder aims to minimize the distance between inputs $\tilde{X}$ and the reconstructed outputs $h^{(n)}$. Since the number of parameters in neural network is exponential, and the availability of training samples is tightly restricted, training deep neural network is challenged by the risk of overfitting. To mitigate this problem, we imposed some sparsity penalties to the hidden layers, thus, the reconstruction loss is shown as:

$$\mathcal{L}_{rec} = ||\tilde{X} - h^{(n)}||_2^2 + \eta \sum_{j=1}^{n} ||b^{(j)}||_2^2. \tag{2}$$

Here, $h^{(n)}$ represents the reconstruction outputs and η is a hyper-parameter to balance the bias of different parts. In this work, the network is heuristically set up as one input layer, three hidden layers, and one output layer. The dimensionality of each layer is set as $13698+(m-1)$, 64, 32, 64, and $13698+(m-1)$, respectively.

C. ACTIVATION FUNCTION IN AUTOENCODER

We employed ELU (Exponential Linear Unit) [28], which speeds up training in deep neural networks and leads to higher classification accuracy, as the activation function σ:

$$\sigma(h^{(j)}) = \begin{cases} h^{(j)} & if \quad h^{(j)} > 0 \\ \alpha(exp(h^{(j)}) - 1) & if \quad h^{(j)} \leq 0 \end{cases} \tag{3}$$

[2]https://blog.keras.io/https://blog.keras.io/

$$\sigma'(h^{(j)}) = \begin{cases} 1 & if \quad h^{(j)} > 0 \\ \sigma(h^{(j)}) + \alpha & if \quad h^{(j)} \leq 0 \end{cases} \tag{4}$$

Here, the ELU hyperparameter α (set $\alpha = 1.0$) controls the value to which an ELU saturates for negative net inputs.

D. OPTIMIZATION WITH ADAM

We utilize a stochastic gradient-based optimization method: Adam (adaptive moment estimation) [29] to train the autoencoder neural network by minimizing the squared reconstruction loss with a sparsity penalty. As far as we know, it is the best technique to accelerate gradient-based optimization. In each training iteration, Adam requires first-order gradients with a small memory consumption. In this case, it computes adaptive learning rates separately for different parameters from estimates of first and second moments of the gradients, combining the advantages of AdaGrad [38] (which works well with sparse gradients) and RMSProp [39] (which works well in on-line and non-stationary settings). The parameter settings in this work are batch size is 64, iteration times is 10k, learning rate is 0.001. Other parameters are set as default.

E. NORMALIZED INITIALIZATION

During training of the neural network, we initialized the biases as 0 and the weight matrix $\omega^{(j)}$ at each layer with a commonly used uniform distribution, as defined in [40] and [41]:

$$\omega^{(j)} \sim U[-\frac{1}{\sqrt{k}}, \frac{1}{\sqrt{k}}] \tag{5}$$

where $U[-a, a]$ is the uniform distribution in the interval $(-1\times10^{-4}, 1\times10^{-4})$ and k is the size of the hidden layer (the number of columns of ω).

F. ADABOOST ALGORITHM FOR CLASSIFIER LEARNING

In the classifier learning phase of our proposed method, the AdaBoost [42] algorithm is employed to train the classifier, which is a supervised learning algorithm designed to calculate a binary classifier that best separates the positive and negative instances.

Given a set of training examples $\{(x_i', y_i)\}_{i=1}^m$, where x_i' denotes the training samples and y_i is a Boolean value allocated according to the clinical information of cancer sufferers during the dataset pre-processing stage. AdaBoost is an effective procedure which boosts the classification accuracy of a simple learning algorithm by combining a collection of weak classifiers $\{g_j(x')\}$ into a stronger classifier $g(x')$. In this work, we adopt decision stump as the weak classifier learning algorithm. The output of $g(x')$ is 1, if x' is classified as a positive instance and 0 otherwise.

Here, we employ a variant of the AdaBoost algorithm proposed by [30]. This variant restricts weak classifier to depending on a single feature f_j only. As a result, each weak classifier consists of a single feature f_j, a threshold θ_j, and a parity p_j which is either -1 or 1, thus indicating the direction

of the inequality.

$$g_j(x') = \begin{cases} 1 & if \ \ p_j f_j(x') < p_j \theta_j \\ 0 & otherwise. \end{cases} \tag{6}$$

The boosting algorithm calculates the optimal values for θ_j and p_j for each weak classifier $g_j(x')$, such that the number of misclassified training samples is minimized. To achieve this, it considers all possible combinations of both p_j and θ_j, for which the number is limited since only an infinite number of training example are given:

$$(p_j, \theta_j) = \arg\min_{(\theta_k, p_k)} \sum_{i=1}^{m} |g_k(x'_i) - y_i| \tag{7}$$

The resulting algorithm is given in Algorithm 1.

Algorithm 1 The AdaBoost Algorithm

Require:

A set of examples $\{(x'_i, y_i)\}_{i=1}^m$, where $y_i = 0, 1$ for negative and positive samples respectively.

Initialization:

Let l and l' be the number of negatives and positives respectively, initialize weights $w_{1,i} = \frac{1}{2l}, \frac{1}{2l'}$ according to the value of y_i.

for $t = 1, ..., T$ **do**

Normalize the weights $w_{t,i}$, according to $\sum_{i=1}^{m} w_{t,i} = 1$.

For each feature f_j, train a weak classifier g_j.

For error ϵ_j of a classifier g_j is evaluated with respect to the weights $w_{t,1}, ..., w_{t,m}$:

$$\epsilon_j = \sum_{i=1}^{m} w_{t,i} |g_j(x'_i) - y_i|.$$

Choose the classifier g_j with the lowest error ϵ_j and set $(g_t, \epsilon_t) = (g_j, \epsilon_j)$.

Update the weights $w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$, where $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$ and $e_i = 0$ while example x'_i is correctly classified by g_t and 1, otherwise.

return The strong classifier is:

$$g(x') = \begin{cases} 1 & if \ \ \sum_{t=1}^{T} \log \frac{1}{\beta_t} g_t(x') \geq \frac{1}{2} \sum_{t=1}^{T} \log \frac{1}{\beta_t} \\ 0 & otherwise. \end{cases}$$

V. RESULTS AND DISCUSSIONS

A. EXPERIMENTAL SETTINGS

1) REPRESENTATIVE CLASSIFIERS FOR COMPARISON PURPOSE

In order to evaluate our ensemble classifier, four typical methods for distant metastasis prediction were analyzed: the 70-gene classifier [2], the 76-gene classifier [5], and two versions of gene set statistics classifier [37]: set-median and set centroid. The 70-gene and 76-gene classifiers are the most famous gene signature based methods used to predict cancer outcome, while the gene set statistics classifiers have been proven to have the comparable performance while being more stable than the previous classifiers [37].

In the 70-gene classifier [2], a total of 70 genes were selected as gene signatures. The average vectors of the 70 genes' expression levels were calculated as the patterns of the two outcome groups (good outcome and bad outcome), and the samples were assigned to the more correlated groups using Person's correlation coefficients. In the 76-gene classifier [5], a total of 76 genes were selected as 76 gene signatures. Based on the 76 genes, a relapse score for each sample was calculated by using the weighted linear combination of the 76 genes' expression values, then each sample was assigned to one of the two outcome groups according to whether the relapse score is higher than a threshold. In the gene set statistics classifier [37], it first downloaded the pre-specified gene sets from the database of MSigDB [10], then the statistical value was calculated to select the optimal feature set which was derived from the gene sets and being used to construct the centroid classifier. There are several statistics methods used to evaluate the gene sets [37], including set-centroid, set-median, PCA, and t-test. We choose the set-centroid and set-median statistics methods as they are reported to perform better than others [37].

Furthermore, we construct a baseline classifier using the same strategy of our proposed method for a comparison between the classifier using deep learning and the classifier not using deep learning.

2) PERFORMANCE MEASURE METRICS

There is a serious imbalance between the number of patients with good outcomes and those with poor outcomes in the cancer datasets. For example, compared against 154 good outcome patients, there are only 28 poor outcome patients in dataset GSE11121. In this scenario, the MCC (Matthews correlation coefficient, shown as Eq. (8)), and the AUC (the area under the receiver operating characteristic curve) which are reported to be the most reliable measure criterions when the distribution of the dataset is highly unbalanced [43], were applied as the two main evaluation metrics.

$$MCC = \frac{t^+ t^- - f^+ f^-}{\sqrt{(t^+ + f^+)(t^+ + f^-)(t^- + f^+)(t^- + f^-)}} \tag{8}$$

Here, t^+ represents true positive, t^- represents true negative, f^+ represents false positive and f^- represents false negative. In addition, accuracy (ACC), specificity (SP) , and sensitivity (SN) were also included in this section, described as follows.

$$ACC = \frac{t^+ + t^-}{t^+ + t^- + f^+ + f^-} \tag{9}$$

$$SN = \frac{t^+}{t^+ + f^-} \tag{10}$$

$$SP = \frac{t^-}{t^- + f^+} \tag{11}$$

B. PERFORMANCES OF OUR ENSEMBLE CLASSIFIERS

In our study, we used GSE2034 as the joint training-evaluation dataset to learn features from gene expression profiles and construct classifiers. Specifically, we performed 10 times five-fold cross validation on GSE2034. For each time, the whole dataset was randomly divided into five groups, four of them were used for training the model, and the rest was used for evaluating the presented method. We averaged all the results and reported the final score as the evaluation performance. During test experiments, we additionally performed independent tests on the other four GEO datasets mentioned in Section III-A. All the results for the proposed two methods were shown in Table 2 and 3, respectively.

TABLE 2. The performance of the PCA-AE-Ada classifier.

Data	ACC	SN	SP	AUC	MCC
GSE2034	0.75	0.76	0.59	0.73	0.33
GSE4922	0.72	0.68	0.56	0.68	0.21
GSE6532	0.77	0.83	0.57	0.69	0.18
GSE7390	0.75	0.77	0.66	0.73	0.27
GSE11121	0.85	0.84	0.55	0.74	0.32

TABLE 3. The performance of the PCA-Ada classifier.

Data	ACC	SN	SP	AUC	MCC
GSE2034	0.65	0.71	0.55	0.63	0.14
GSE4922	0.53	0.57	0.40	0.52	-0.01
GSE6532	0.55	0.55	0.43	0.50	-0.02
GSE7390	0.66	0.71	0.42	0.58	0.12
GSE11121	0.68	0.74	0.32	0.53	0.05

From Table 2, it can be concluded that our proposed PCA-AE-Ada classifier performs well, and it behaves in an stable manner on both the training-validation set and the independent test sets. It achieves fairly good AUC scores (almost above 0.70) and ACC rates (almost above 0.8) and demonstrates superior performances on detecting either positive instances or negative instances (with fairly good SN and SP).

Comparing the performance of the PCA-AE-Ada classifier with that of PCA-Ada classifier, we can find that PCA-AE-Ada with deep learning techniques works better than PCA-Ada, which is constructed with compressed features from PCA for all the evaluation metrics. Interestingly, from the SN and SP rates, it can be seen that PCA-AE-Ada demonstrates better performance for detecting both positive instances and negative instances, while PCA-Ada works well for positive instances only. This means that deep learning can effectively alleviate the problem of unbalanced distribution of training datasets, and enhance the generalization ability of the classifiers.

C. COMPARING THE RESULTS WITH REPRESENTATIVE METHODS

The AUC and MCC scores resulted from our method and the other four classifiers on the five datasets are shown in Fig. 4 and 5, respectively. Note that the details of the other four methods are not shown.

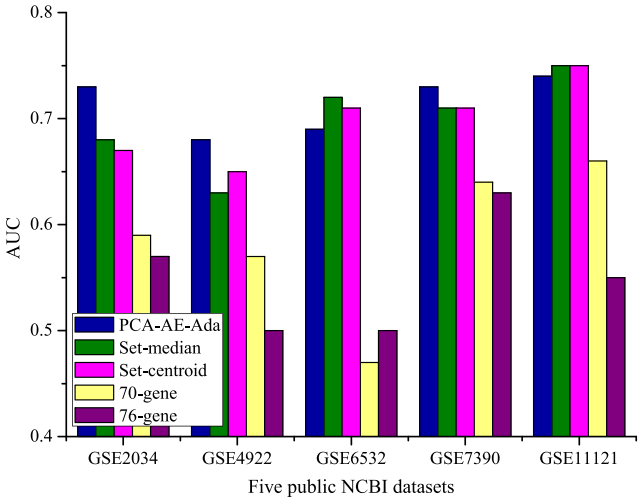


FIGURE 4. The AUC scores of the five classifiers on the five datasets.

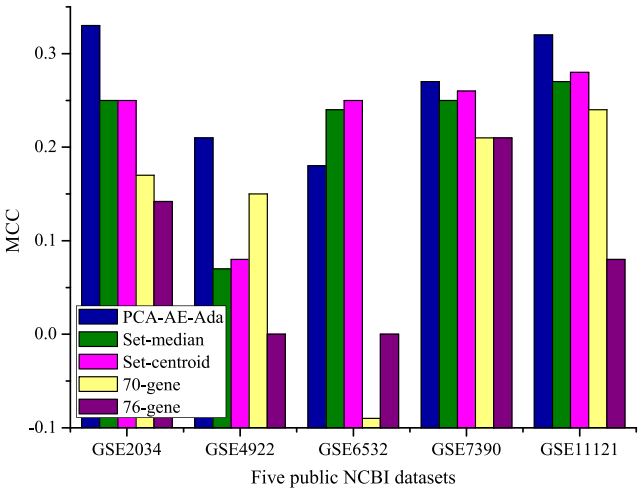


FIGURE 5. The MCC scores of the five classifiers on the five datasets.

From Fig. 4, our ensemble classifier achieves the best AUC performance on most datasets, albeit it does not perform as well as the two gene set based methods (set-median and set-centroid) on GSE11121. Then is the gene set based methods, and these two methods achieve better AUC performances than the two gene signature classifiers. The similar phenomenon can be found through the scores of MCC.

From Fig. 5, four representative classifiers (especially the gene signature based classifiers) except for ours perform worse on GSE4922 and GSE6532 than on the other three datasets. We explain this for the fact that these two datasets contain both lymph node-negative and lymph node-positive patients, while the other datasets contain lymph node-negative patients only. Surprisingly, our ensemble classifier shows a robust performance with respect to GSE4922: it reaches MCC of 0.21 (AUC of 0.68), which significantly outperforms the others.

With the aforementioned observation, it can be seen that the proposed method is less sensitive to unbalanced datasets and is actually more stable, in contrast to the other four classifiers show dramatic variations with respect to the

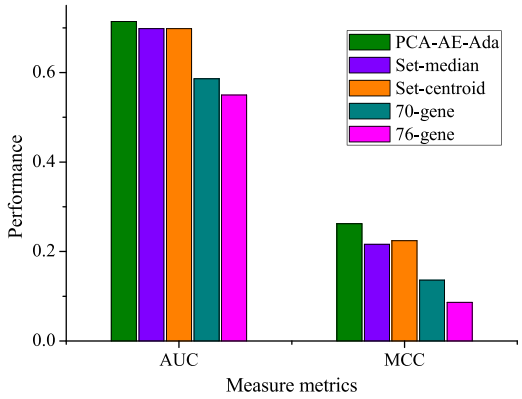


FIGURE 6. The overall performance of our classifier and other four representative classifiers over the five NCBI datasets.

different datasets. However, our classifier still fails to work on dataset GSE6532; it may be caused by an implicit drive towards the way to go precisely because these are the harder cases in the clinical setting requiring proper treatment planning.

Furthermore, to demonstrate the performance of our method in a more comprehensive manner, the AUCs and MCCs of our method against other four methods were averaged over the five public NCBI datasets, and Fig. 6 reported the final results.

Comprehensive analysis shows that our classifier reaches an AUC of over 0.714, while the two gene set classifiers have a significantly worse result about 0.55. Nevertheless, the two gene signature classifiers can only reach an AUC of smaller than 0.6. The similar phenomenon can be seen from the indexes of the MCC.

In conclusion, our ensemble classifier based on PCA and autoencoder features is superior to other published methods, it achieves better classification accuracy as well as better generalization ability with respect to the different datasets.

VI. CONCLUSION

In this paper, we present a new method to predict the clinical outcomes of cancer patients with using deep learning. In the feature learning phase, principal component analysis and an autoencoder neural network are combined with the exportation of deep learning techniques for the purpose of learning more representative features from gene expression data. In the classifier learning phase, we utilize the AdaBoost algorithm to construct an ensemble classifier for the final prediction task. As the evaluation test results demonstrated, our proposed method shows more powerful prediction ability, and the classifier constructed with deep learning techniques performs better than the others. Through our analysis and discussion, the features which extracted automatically by the neural network showed an excellent ability for rapid generalization and explicitly improved the performance of outcome prediction.

However, there are still some drawbacks to our classifier. Firstly, the model constructed is not easy to analyze—this is a common problem in neural networks. In addition, identifying which features are most important to the prediction task is difficult. Secondly, due to the complex structure of the deep learning model, the amount of data is less prone to over-fitting. Although our model has achieved good results, the generalization capacity needs to be further improved with more publicly available datasets.

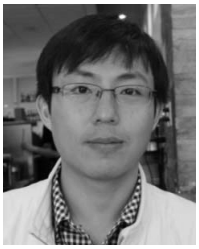
REFERENCES

[1] W. K. Lim, E. Lyashenko, and A. Califano, "Master regulators used as breast cancer metastasis classifier," in *Proc. Pacific Symp. Biocomput.*, 2009, pp. 504–515.
[2] M. J. Van de Vijver *et al.*, "A gene-expression signature as a predictor of survival in breast cancer," *New England J. Med.*, vol. 347, no. 25, pp. 1999–2009, 2002.
[3] T. Sørlie *et al.*, "Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications," *Proc. Nat. Acad. Sci. USA*, vol. 98, no. 19, pp. 10869–10874, 2001.
[4] J. Li *et al.*, "Identification of high-quality cancer prognostic markers and metastasis network modules," *Nature Commun.*, vol. 1, Jul. 2010, Art. no. 34.
[5] Y. Wang *et al.*, "Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer," *Lancet*, vol. 365, no. 9460, pp. 671–679, 2005.
[6] H.-Y. Chuang, E. Lee, Y.-T. Liu, D. Lee, and T. Ideker, "Network-based classification of breast cancer metastasis," *Mol. Syst. Biol.*, vol. 3, no. 1, pp. 140–150, 2007.
[7] X. Zhou, J. Liu, X. Ye, W. Wang, and J. Xiong, "Ensemble classifier based on context specific miRNA regulation modules: A new method for cancer outcome prediction," *BMC Bioinform.*, vol. 14, no. S12, pp. 1–11, 2013.
[8] T. N. Sainath, A.-R. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2013, pp. 8614–8618.
[9] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 577–585.
[10] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–9.
[11] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 3642–3649.
[12] A. Coates, B. Huval, T. Wang, D. Wu, B. Catanzaro, and N. Andrew, "Deep learning with cots HPC systems," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1337–1345.
[13] P. Baldi and P. J. Sadowski, "Understanding dropout," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2814–2822.
[14] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1139–1147.
[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[16] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.
[17] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng, "Parsing natural scenes and natural language with recursive neural networks," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 129–136.
[18] M. K. Leung, H. Y. Xiong, L. J. Lee, and B. J. Frey, "Deep learning of the tissue-regulated splicing code," *Bioinformatics*, vol. 30, no. 12, pp. i121–i129, 2014.
[19] P. Di Lena, K. Nagata, and P. Baldi, "Deep architectures for protein contact map prediction," *Bioinformatics*, vol. 28, no. 19, pp. 2449–2457, 2012.
[20] D. Quang, Y. Chen, and X. Xie, "DANN: A deep learning approach for annotating the pathogenicity of genetic variants," *Bioinformatics*, vol. 31, no. 5, pp. 761–763, 2015.
[21] R. Heffernan *et al.*, "Improving prediction of secondary structure, local backbone angles, and solvent accessible surface area of proteins by iterative deep learning," *Sci. Rep.*, vol. 5, p. 11476, Jun. 2015.
[22] M. Spencer, J. Eickholt, and J. Cheng, "A deep learning network approach to ab initio protein secondary structure prediction," *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 12, no. 1, pp. 103–112, Jan./Feb. 2015.

[23] T. Lee and S. Yoon, "Boosted categorical restricted Boltzmann machine for computational prediction of splice junctions," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2483–2492.
[24] T. W. Nilsen and B. R. Graveley, "Expansion of the eukaryotic proteome by alternative splicing," *Nature*, vol. 463, no. 7280, p. 457, 2010.
[25] E. Asgari and M. R. Mofrad, "Continuous distributed representation of biological sequences for deep proteomics and genomics," *PLoS ONE*, vol. 10, no. 11, p. e0141287, 2015.
[26] R. Fakoor, F. Ladhak, A. Nazi, and M. Huber, "Using deep learning to enhance cancer diagnosis and classification," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1–7.
[27] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
[28] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. (2016). "Fast and accurate deep network learning by exponential linear units (ELUs)." [Online]. Available: https://arxiv.org/abs/1511.07289
[29] D. Kingma and J. Ba. (2015). "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980
[30] M. J. Jones and P. Viola, "Robust real-time object detection," in *Proc. Workshop Stat. Comput. Theories Vis.*, vol. 266, 2001, p. 56.
[31] P. Langley, "Selection of relevant features in machine learning," in *Proc. AAAI Fall Symp. Relevance*, vol. 184, 1994, pp. 245–271.
[32] E. Segal, D. Pe'er, A. Regev, D. Koller, and N. Friedman, "Learning module networks," *J. Mach. Learn. Res.*, vol. 6, pp. 557–588, Apr. 2005.
[33] L. Jeleń, A. Krzyżak, T. Fevens, and M. Jeleń, "Influence of feature set reduction on breast cancer malignancy classification of fine needle aspiration biopsies," *Comput. Biol. Med.*, vol. 79, pp. 80–91, Dec. 2016.
[34] Z. M. Hira and D. F. Gillies, "A review of feature selection and feature extraction methods applied on microarray data," *Adv. Bioinform.*, vol. 2015, 2015, Art. no. 198363. [Online]. Available: https://www.hindawi.com/journals/abi/2015/198363/cta/, doi: 10.1155/2015/198363.
[35] S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," *Briefings Bioinform.*, vol. 18, no. 5, pp. 851–869, 2016.
[36] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
[37] G. Abraham, A. Kowalczyk, S. Loi, I. Haviv, and J. Zobel, "Prediction of breast cancer prognosis using gene set statistics provides signature stability and biological context," *BMC Bioinform.*, vol. 11, no. 1, p. 277, 2010.
[38] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, Feb. 2011.
[39] T. Tieleman and G. Hinton, "Lecture 6.5-RMSProp: Divide the gradient by a running average of its recent magnitude," *COURSERA, Neural Netw. Mach. Learn.*, vol. 4, no. 2, pp. 26–31, 2012.
[40] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.
[41] Y. Chen, Y. Li, R. Narayan, A. Subramanian, and X. Xie, "Gene expression inference with deep learning," *Bioinformatics*, vol. 32, no. 12, pp. 1832–1839, 2016.
[42] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Proc. Eur. Conf. Comput. Learn. Theory*, Springer, 1995, pp. 23–37.
[43] MAQC Consortium, "The MicroArray quality control (MAQC)-II study of common practices for the development and validation of microarray-based predictive models," *Nature Biotechnol.*, vol. 28, no. 8, pp. 827–838, 2010.

DEJUN ZHANG was born in 1982. He received the Ph.D. degree from the School of Computer, Wuhan University, China, in 2015. He is currently a Lecturer with the Faculty of the College of Information and Engineering, Sichuan Agricultural University, China. He has authored or co-authored over 10 papers in journals and conferences. His research areas include machine learning, bioinformatics, and computer graphics. He has conducted a considerable amount of research in digital geometric processing and computational photography. Since 2015, he has been serving as a Senior Member of the China Society for Industrial and Applied Mathematics (CSIAM) and a Committee Member of the geometric design & computing of CSIAM.

LU ZOU was born in Chengdu, China, in 1996. She is currently a Senior Student with the College of Information and Engineering, Sichuan Agricultural University, China. She has been engaged in scientific research for 3 years and has authored eight articles in referred journals and proceedings in the areas of bioinformatics and computer vision. She is a member of the China Computer Federation.



XIONGHUI ZHOU was born in China in 1986. He received the Ph.D. degree from the School of computer, Wuhan University, China, in 2014. He is currently an Associate Professor with the Faculty of the College of Informatics, Huazhong Agricultural University, China. He has authored or co-authored over 10 papers in journals and conferences. His research areas include data mining and bioinformatics. He has conducted a considerable amount of research in bioinformatics. Since 2016, he has been serving as a member of China Artificial Intelligence Association.

FAZHI HE received the Ph.D. degree from the Wuhan University of Technology. He was a Post-Doctoral Researcher with The State Key Laboratory of CAD & CG, Zhejiang University, a Visiting Researcher with the Korea Advanced Institute of Science & Technology, and a Visiting Faculty Member with the University of North Carolina at Chapel Hill. He became an Assistant Professor with Wuhan University in 2001, and became a Professor in 2006. He is currently a Professor with the State Key Laboratory of Software Engineering, School of Computer, Wuhan University. He has authored or co-authored over 100 refereed articles in journals and conference proceedings. His research interests are computer graphics, computer-aided design, and computer supported cooperative work. He has been serving as a Senior Member with the China Society for Industrial and Applied Mathematics (CSIAM) and a Committee Member of the geometric design & computing of CSIAM. He is a member of the Editorial Board for the *Journal of Computer-Aided Design & Computer Graphics*. He conducts a program for the Asian Conference on Design and Digital Engineering in 2018 and a program for the 2018 IEEE 22st International Conference on Computer Supported Cooperative Work in Design. He has received several best paper awards to date.