Neural Comput & Applic (2013) 22:457–468
DOI 10.1007/s00521-012-0847-z

Classification of bioinformatics dataset using finite impulse response extreme learning machine for cancer diagnosis

Kevin Lee · Zhihong Man · Dianhui Wang · Zhenwei Cao

Received: 5 August 2011 / Accepted: 16 January 2012 / Published online: 31 January 2012

Abstract In this paper, the classification of the two binary bioinformatics datasets, leukemia and colon tumor, is further studied by using the recently developed neural network-based finite impulse response extreme learning machine (FIR-ELM). It is seen that a time series analysis of the microarray samples is first performed to determine the filtering properties of the hidden layer of the neural classifier with FIR-ELM for feature identification. The linear separability of the data patterns in the microarray datasets is then studied. For improving the robustness of the neural classifier against noise and errors, a frequency domain gene feature selection algorithm is also proposed. It is shown in the simulation results that the FIR-ELM algorithm has an excellent performance for the classification of bioinformatics data in comparison with many existing classification algorithms.

Keywords Microarray gene expression data · Extreme learning machine · FIR filter · Classification · Linear separability

K. Lee (✉) · Z. Man · Z. Cao
Faculty of Engineering and Industrial Sciences,
Swinburne University of Technology, Hawthorn,
VIC 3122, Australia
e-mail: kklee@swin.edu.au

Z. Man
e-mail: zman@swin.edu.au

Z. Cao
e-mail: zcao@swin.edu.au

D. Wang
Department of Computer Science and Computer Engineering,
La Trobe University, Bundorra, VIC 3086, Australia
e-mail: dh.wang@latrobe.edu.au

1 Introduction

The analysis of cancer diagnosis data is one of the most important research fields in medical science and bioengineering [1, 2]. As the complete treatments of cancers have not been achieved, the early diagnosis plays an important role for doctors to help patients to control the metastatic cancer growth. Recently, the microarray gene expression datasets, consisting of thousands of gene expressions that can be used for the molecular diagnosis of cancer, have been established [2, 3]. The sample vectors in these datasets contain a large amount of information about the origin and development of cancers. However, due to the fact that all of these data contain different levels of noises and measurement errors from sampling processes, it is difficult for the existing classification techniques to accurately identify the patterns from the samples [3]. In order to use the existing classification techniques for the pattern classification of the microarray gene expression datasets, a gene selection algorithm is usually implemented to reduce the effects of the noise, error, and the high dimensionality of samples [4–6]. It has been shown in [1] that the gene selection process can help to improve the performance of the classifier by identifying representative gene expression subsets. Some of the popular gene selection algorithms include the principal component analysis [4], the singular value decomposition [5], the independent component analysis [6], genetic algorithm (GA) [7], and the recursive feature elimination method [8, 9].

Other related works on the classifiers are the cancer detection system proposed in [7], the support vector machine (SVM) in [9–12] and the extreme learning machine (ELM) in [13–17]. In [7], the GA gene selection algorithm is applied to the microarray dataset to obtain the most informative gene subsets that are classified using the multilayer perceptron (MLP). In [9–12], the support vector machine first maps the input space into the high dimensional feature space and then constructs an optimal hyperplane using selected support vectors to separate the classes. In [13–17], the ELM simplifies model selection by randomly generating the hidden layer weights and biases and performs very fast batch training using the Moore–Penrose generalized pseudo-inverse to deterministically obtain the output weights. Most importantly, the ELM performs well in several cancer diagnosis applications [14–17]. However, all the algorithms mentioned above are used for the classification of a small subset of genes that does not sufficiently represent the whole process of the origin and the development of cancers in general [2]. In addition, the robustness issues with respect to noises and disturbances are not discussed in detail.

Please note that there has not been a standard guideline in the biomedical industry for producing microarrays, and the microarrays produced by different laboratories may contain different profiles of error or different sets of genes. Furthermore, even samples taken from the same laboratory may sometimes contain different composition of cells that may bias the accuracy of a classifier [2, 3]. Therefore, the motivation of this paper is to develop a general microarray classification technique that is capable of classifying a variety of genes based on the FIR-ELM developed by Man et al. [18]. The FIR-ELM algorithm implements a single hidden layer feedforward neural network (SLFN) as the classifier, where the well-known filtering methods, such as the finite length low-pass filtering, high-pass filtering, and band-pass filtering in digital signal processing, are adopted to train the input weights in the hidden layer to extract features from the dataset. The readers may find the details of the filtering techniques from [18, 19].

As seen in [18], the hidden layer of the neural classifier with the FIR-ELM is based on FIR filter designs. The sample vectors from the microarray datasets are thus treated as time series input pattern vectors. To show the validity of applying FIR filter theory in microarray gene feature detection, a time series analysis for a binary microarray gene expression dataset is first explored with the gene expression dataset denoted as the time series type data. Then, the FIR filtering function is applied to the time series to reveal spectrum features. The filtered time series are then analysed using the cross-correlation to show the importance of proper filter design selection to achieve optimal separation of the classes. In addition, in order to provide a quantitative measure of the gene features within each dataset, a linear separability (LS) analysis on the microarrays is also discussed in detail based on the recent study in [20] on the LS of microarray gene pairs.

It will be seen that, in order to determine the filtering properties of the hidden layer of the neural classifier with the FIR-ELM, a frequency domain gene feature selection (FGFS) algorithm is developed for analyzing the frequency characteristics of all datasets. The outcomes of the FGFS are then used to determine the optimal FIR filtering strategy for the input weights' design. In addition, the effects of the time series with the randomly ordered gene samples are also examined in this paper.

The rest of this paper is organized as follows. Section 2 describes the characteristics of the time series type patterns of the microarray samples. Section 3 presents the analysis on the linear separability of the patterns. Section 4 outlines the FIR-ELM as well as the FSGS algorithm. Section 5 shows the experimental results and the performance analysis of the FIR-ELM compared with other existing algorithms. Sections 6 and 7 give the discussions and conclusions respectively.

2 Time series analysis of microarrays

In order to analyse and classify the data patterns in the microarray gene expression datasets using the FIR-ELM, in this paper, we express all samples as the time series type data. For the binary dataset with N samples, we assume that the first N_1 samples belong to class 1 and the other N_2 samples belong to class 2, with $N = N_1 + N_2$. The gene expressions in the two classes can then be denoted as:

$$C^1 = \begin{bmatrix} g^1_{1,1} & g^1_{1,2} & \cdots & g^1_{1,\tilde{n}} \\ g^1_{2,1} & g^1_{2,2} & & \vdots \\ \vdots & & \ddots & \vdots \\ g^1_{N_1,1} & \cdots & \cdots & g^1_{N_1,\tilde{n}} \end{bmatrix} = \begin{bmatrix} \mathbf{g^1_1} & \mathbf{g^1_2} & \cdots & \mathbf{g^1_n} \end{bmatrix} \tag{1}$$

$$C^2 = \begin{bmatrix} g^2_{1,1} & g^2_{1,2} & \cdots & g^2_{1,\tilde{n}} \\ g^2_{2,1} & g^2_{2,2} & & \vdots \\ \vdots & & \ddots & \vdots \\ g^2_{N_2,1} & \cdots & \cdots & g^2_{N_2,\tilde{n}} \end{bmatrix} = \begin{bmatrix} \mathbf{g^2_1} & \mathbf{g^2_2} & \cdots & \mathbf{g^2_n} \end{bmatrix} \tag{2}$$

where

$$\mathbf{g^1_l} = \left\{ g^1_{1,l}, g^1_{2,l}, \ldots, g^1_{N_1,l} \right\}^T \quad l \in 1, 2, \ldots, \tilde{n} \tag{3}$$

$$\mathbf{g^2_l} = \left\{ g^2_{1,l}, g^2_{2,l}, \ldots, g^2_{N_2,l} \right\}^T \quad l \in 1, 2, \ldots, \tilde{n} \tag{4}$$

C^1 and C^2 are the sample matrices for class 1 and class 2, respectively, $\mathbf{g^1_l}$ and $\mathbf{g^2_l}$ are the gene indices, and $\tilde{n}$ is the number of genes in a sample.

For the purpose of conducting a bivariate time series analysis, the cells in the microarray tests are treated as a black box that outputs two types of time series, $Y_1(t)$ and

$Y_2(t)$, to represent non-cancerous and cancerous states, or any other binary phenotypes, as follows:

$$Y_1(t) = \{y_1(t) : t \in 1, 2, \ldots, \tilde{n}\} \tag{5}$$

$$Y_2(t) = \{y_2(t) : t \in 1, 2, \ldots, \tilde{n}\} \tag{6}$$

The samples from each of the binary classes C^1 and C^2 can then be aggregated to represent their respective classes as in (5) and (6). It is well known that the gene expression values in (3) and (4) have an Lorentzian-like distribution with many outliers [3, 21]. Hence, the median that is a better maximum likelihood estimator under impulsive noise conditions is preferred, as compared with the mean to represent the distribution of each gene in a specific class [1, 22]. Each gene in (3) and (4) are then aggregated by taking the median to be a temporal datum as follows:

$$Y_1(t) = \mathrm{median}\left(\mathbf{g}_\mathbf{t}^\mathbf{1}\right) \quad t \in 1, 2, \ldots, \tilde{n} \tag{7}$$

$$Y_2(t) = \mathrm{median}\left(\mathbf{g}_\mathbf{t}^\mathbf{2}\right) \quad t \in 1, 2, \ldots, \tilde{n} \tag{8}$$

Figure 1 shows two time series type data plots, $Y_1(t)$ and $Y_2(t)$, for the colon tumor dataset.

It is seen that both pattern classes in Fig. 1 have the similar trend and are very noisy. In order to determine the input weights of the neural classifier for the pattern classification purpose, the low-pass, high-pass, and band-pass filters are used to filter $Y_1(t)$ and $Y_2(t)$, respectively, where three filters use the normalized cutoff frequency of 0.4, and the band-pass filter uses a bandwidth of ± 0.05. As the microarray samples are converted from data vectors to time series, there is no directly derivable sampling frequency. Thus, the normalized frequency is used to represent the unit of cycles per sample.

The filtered time series are defined as:

$$Y_1^f(t) = \sum_{i=0}^{\tilde{n}-1} \psi(i) Y_1(t-i) \quad t \in 1, 2, \ldots, \tilde{n} \tag{9}$$

$$Y_2^f(t) = \sum_{i=0}^{\tilde{n}-1} \psi(i) Y_2(t-i) \quad t \in 1, 2, \ldots, \tilde{n} \tag{10}$$

where $\psi = \{\psi_0, \psi_1, \ldots, \psi_{\tilde{n}-1}\}$ are the $n-1$th order filter coefficients derived from the impulse response of the respective FIR filters, a detailed discussion on the generation of filter coefficients is given in [19]. The cross-correlation between the two filtered time series (9) and (10) can then be obtained. The cross-correlation may be interpreted as a measure of efficiency between the different filters used to extract important features and reduce the effects of noise. It has been shown in [23] that the generation of weakly correlated class features improves the machine learning performance. Also, Yang et al. in [24] show that the weak correlation can be used as a decision making condition to differentiate samples.

Before computing the cross-correlations, we need to first remove the non-stationary disturbances in the two classes. A first-order forward finite difference approximation [25] is applied to the time series to remove the trend and attain stationarity. The detrended time series is defined as:

$$Y_1^{df}(t) = Y_1^f(t+1) - Y_1^f(t) \quad t \in 1, 2, \ldots, \tilde{n}-1 \tag{11}$$

$$Y_2^{df}(t) = Y_2^f(t+1) - Y_2^f(t) \quad t \in 1, 2, \ldots, \tilde{n}-1 \tag{12}$$

Remark 2.1 In order to avoid introducing random errors that may be accumulated throughout the differencing calculations, the first gene term is used as the initial state instead of an arbitrary value. Thus, the time series (11) and (12) are reduced in length by one to produce the forward finite difference approximation.

Figure 2 shows the new time series Y_1^{df} and Y_2^{df}. The cross-correlation of the two time series can then be obtained using the sample correlation coefficient defined in

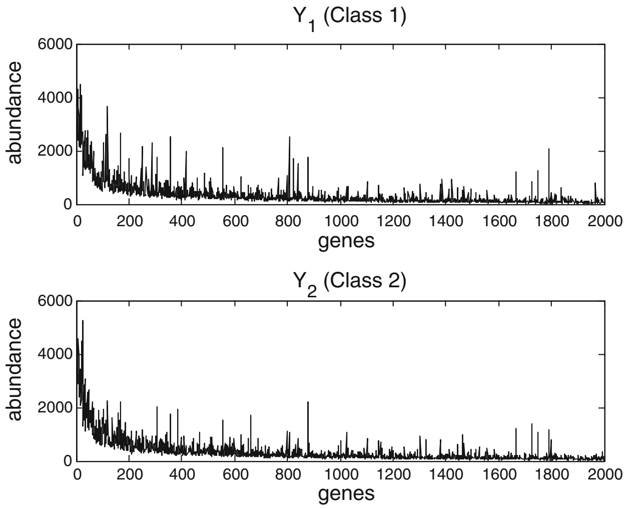

Fig. 1 Aggregated time series for the colon tumor dataset

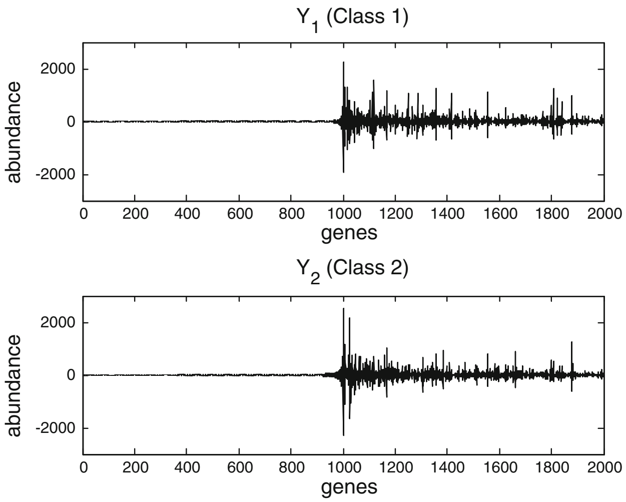

Fig. 2 The filtered and detrended time series Y_1^{df} and Y_2^{df}

Table 1 Correlation coefficient of colon tumor binary classes using different FIR filters

Filter type	Low pass	High pass	Band pass	No filter
$r_{Y_1^{df}Y_2^{df}}$	0.7480	0.6746	0.6816	0.6386

[25], which is the well-known Pearson correlation coefficient. The sample correlation coefficient is denoted as:

$$r_{Y_1^{df}Y_2^{df}} = \frac{\sum_{t=1}^{\tilde{n}-1}\left(Y_1^{df}(t)-\bar{Y}_1^{df}\right)\left(Y_2^{df}(t)-\bar{Y}_2^{df}\right)}{\sqrt{\left[\sum_{t=1}^{\tilde{n}-1}\left(Y_1^{df}(t)-\bar{Y}_1^{df}\right)^2\sum_{t=1}^{\tilde{n}-1}\left(Y_2^{df}(t)-\bar{Y}_2^{df}\right)^2\right]}} \tag{13}$$

where $\bar{Y}_1^{df}$ and $\bar{Y}_2^{df}$ are the means of Y_1^{df} and Y_2^{df}, respectively.

The correlation coefficient in (13) calculates the level of linear association between the two time series within [−1, 1] where 0 is completely uncorrelated and 1 is proportionally correlated, while −1 is inversely correlated. Table 1 shows the correlation coefficient of the two times series Y_1^{df} and Y_2^{df}, respectively, after low-pass, high-pass, and band-pass filtering at the same normalized cutoff frequency of 0.4. The correlation coefficient of the non-filtered data is also given as a reference. The results in Table 1 show that the high-pass filtering produces the time series pair, which are least correlated among the filtered time series.

Remark 2.2 Although the correlation coefficient of the filtered time series are higher than the non-filtered case, their practicality in comparing the effectiveness of different filter types is still relevant. The higher correlation coefficients among the filtered time series are mainly due to the similarity of the non-distinctive residue components that remain after the filtering process as seen in the first half of the plots in Fig. 2. This is consistent with the motivation of using the filtering process to discover features at specific frequency ranges. Therefore, it is adequate to compare the results among the filtered time series only to determine the suitability of each filter design.

To visualize the differences in the actual time series (11) and (12), we then compute the residual that is the squared differences between time series (11) and (12) as follows:

$$y_{\text{residual}}(t) = \left(Y_1^{df}(t) - Y_2^{df}(t)\right)^2 \quad t \in 1, 2, \ldots, \tilde{n}-1 \tag{14}$$

It can be seen from the plot of the residual in Fig. 3 that a certain region of genes contains expression values that are very different between the two classes. The genes within the indices of 1,000 to 1,100 show significant residual magnitudes with the highest at the index of 1,015.

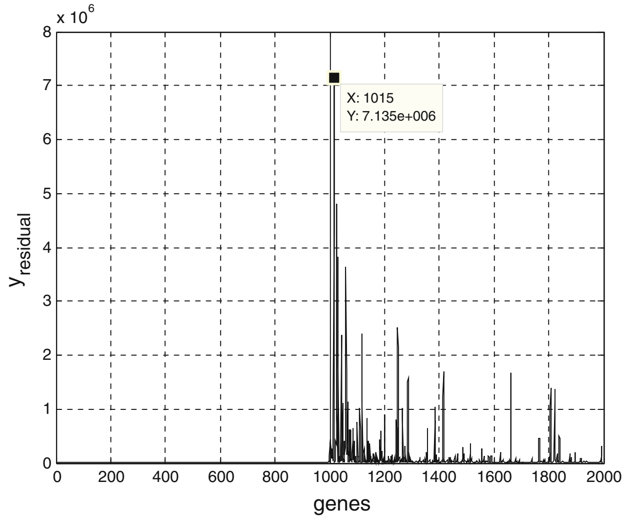

Fig. 3 Plot of residual between Y_1^{df} and Y_2^{df}

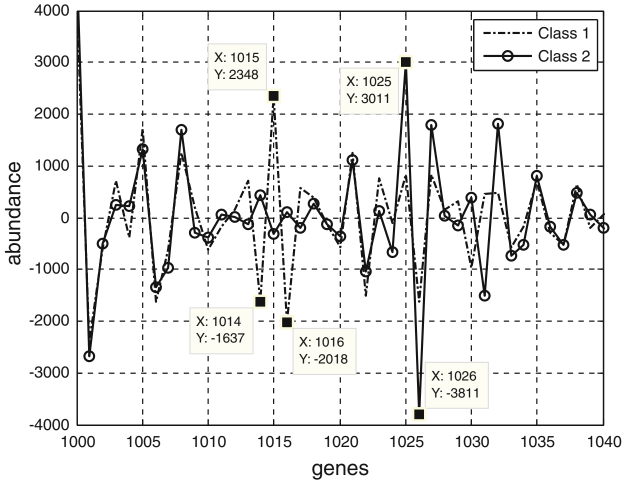

Fig. 4 Overlaid plot of Y_1^{df} and Y_2^{df} for genes 1,000–1,040

Closer inspection of the two filtered time series overlaid on each other to show the gene expressions between indices 1,000 to 1,040 in Fig. 4 confirms the observations. The three genes 1,014, 1,015, and 1,016 are seen to behave dissimilarly (opposite of y axis), while the genes 1,025 and 1,026 have significant differences in magnitudes between the two classes.

Remark 2.3 Generally, the samples from both classes in a binary microarray dataset tend to look similar. Therefore, the genes with large residual magnitudes contribute to the linear separability of the samples in the microarray dataset. However, it is quite obvious that it is not sufficient to identify the genes with the cross-correlation. This point can be seen in Table 1. Hence, a more comprehensive test for linear separability is required.

3 Linear separability of microarrays

The time series analysis using the cross-correlation in the previous section reveals some features of the separability of the classes in bioinformatics datasets. In this section, however, we will further investigate the linear separability and provide a quantitative measurement of linear separability of the data patterns.

Please note that the single gene test was first developed in [26] where each gene was tested using all samples to find the total number of linearly separable genes. Then, the gene pair linear separability analysis was developed in [20], where pairs of genes are tested using all possible combinations. The genes that are found to be linearly separable in the gene pair analysis is also guaranteed to include the linearly separable single genes. Therefore, the gene pair analysis provides extra information on genes that may only show the linear separability characteristic in pairs.

The gene pair analysis algorithm in [20] is used here because of the relatively low computational cost using an incremental testing approach. First, the $N = N_1 + N_2$ samples are separated into their respective classes as in (1) and (2). Then, each pair of genes in the dataset can be defined as $g_{ij} = \left(g_i, g_j\right)$, and for a dataset with $\tilde{n}$ genes, there would be $^{\tilde{n}}C_2$ possible combinations. The pairs of genes can then be projected on the 2D plane. The algorithm states that a pair of genes is linearly separable if there exists a line L where all the N_1 points of class 1 are located on one side of L and all the N_2 points of class 2 are located on the other side (no point is allowed to reside on L itself). Each gene pair sample is added incrementally, and the algorithm stops whenever a new gene pair introduced violates the separability condition.

However, as stated in [20], it might be impossible to find linearly separable gene pairs in medium to large datasets even if they are highly separable. In order to solve this problem, we propose a new sample selection process for testing the leukemia and colon tumor datasets described in Table 2, as follows: We choose the number of samples according to the guidelines provided in [27], which states the minimum number of genes required for statistical significance. Finally, the test is repeated for 20 times to obtain averaged results.

Table 3 shows the sample selection from each class and the number of linearly separable gene pairs for each dataset. The leukemia dataset has a high number of linearly separable gene pairs; therefore, it should be more easily classified. The colon tumor dataset, however, is found to consist of only a small number of linearly separable gene pairs. Hence, the colon tumor dataset is defined intuitively as 'harder' to classify than the leukemia dataset. The results presented here are descriptive of the original data itself. These results will later be used as a benchmark in the analysis of the FIR-ELM in the experiments section.

Table 2 Summary of leukemia and colon dataset

Dataset	Samples	Genes	Class 1	Class 2
Leukemia	72	7129	47 (ALL)	28 (AML)
Colon	62	2000	40 (tumor)	22 (normal)

Table 3 Linearly separable gene pairs for leukemia and colon dataset

Dataset	Samples ($N_1 + N_2$)	Mean	SD
Leukemia	30 (15 + 15)	29,009	15,875
Colon	30 (15 + 15)	93	171

Remark 3.1 The standard deviations for the linearly separable gene pairs given in Table 3 are large when compared to their mean values. This is typical for the microarray datasets as the gene values are often disturbed by systemic and random noise and hence follows an Lorentzian-like distribution with wider tails [21]. The random selection of samples during each iteration also contributes to a wider distribution of the trials as there may be samples with a large number of outliers.

4 Outline of the FIR-ELM

Recently, a new breed of SLFN introduced by Huang et al. in [13], called the ELM, has been proven to simplify the neural network training process into solving a set of linear equations. The ELM algorithm first initializes the hidden layer weights and biases randomly and then proceeds to compute the output weights deterministically using the Moore–Penrose pseudo-inverse. The learning capabilities of the ELM have been shown in [14–17] to produce good results in terms of classification accuracy. However, the randomly generated weights provide sub-optimal classifiers that are prone to noise and other disturbances within the data [18, 28].

The FIR-ELM in [18] is a modified version of the ELM with the purpose of improving the robustness. The hidden layer weights of the SLFN are designed using the FIR filter theory and the output layer weights are derived using convex optimization methods. A brief overview of the FIR-ELM is as follows:

4.1 FIR-ELM

For a set of N distinct samples, $\{(C,T)|C = [c_1, \ldots, c_N], T = [t_1, \ldots, t_N].\}$ where $c_i \in R^n$ is an $n \times 1$ input vector and $t_i \in R^m$ is an $m \times 1$ target vector, the $\tilde{N}$ neuron SLFN, with the activation function $h(x)$ can be modeled as

$$f_{\tilde{N}}(c_i) = \sum_{k=1}^{\tilde{N}} \beta_k h(w_k \cdot c_i + b_k) \quad i \in 1, 2, \ldots, N \tag{15}$$

where $w_k = [w_{k1}, \ldots, w_{kn}]$ is the n-dimensional weight vector connecting the kth hidden node and the input nodes, $\beta_k = \left[\beta_{k1}, \ldots, \beta_{k\tilde{N}}\right]^T$ is the output weights vector connecting the kth hidden node and the output nodes, b_k is the bias vector of the kth hidden node, and, $t_i = [t_1, \ldots t_N]^T$ is the target output vector with respect to $c_i = [c_1, \ldots c_N]^T$.

For any bounded non-constant piecewise continuous activation function $h(x)$, it has been proven that SLFNs with $\tilde{N}$ hidden nodes can approximate $N \leq \tilde{N}$ samples with zero error such that $\sum_{j=1}^{N} \left| f_{\tilde{N}_j} - t_j \right| = 0$ [29]. Therefore, there exists β_k, and b_k that satisfies (15) above, and the equation can be simplified into matrices $H\beta = T$ shown in (16)

$$H = \begin{bmatrix} h(w_1 \cdot c_1 + b_1) & \cdots & h\left(w_{\tilde{N}} \cdot c_1 + b_{\tilde{N}}\right) \\ \vdots & \ddots & \vdots \\ h(w_1 \cdot c_N + b_1) & \cdots & h\left(w_{\tilde{N}} \cdot c_N + b_{\tilde{N}}\right) \end{bmatrix} \tag{16}$$
$$\beta = \left[\beta_1, \ldots \beta_{\tilde{N}}\right]^T \quad \text{and} \quad T = [t_1, \ldots, t_N]^T$$

The definition of the SLFN up until (16) is similar to the ELM. However, in addition to the typical SLFN architecture, the FIR-ELM introduces an input tapped delay line with $n - 1$ delay units at the front of the SLFN and uses both the linear hidden nodes and linear output nodes. The input tapped delay line represents a finite depth memory where the current state and $n - 1$ past states of a variable are used as the input to the SLFN. Such SLFN architecture introduces system dynamics in the training process and is capable of universal approximation [18]. A diagram of the SLFN architecture used in this paper is given in Fig. 5, where D is the unit delay element, and k is the index for the input sample c_i.

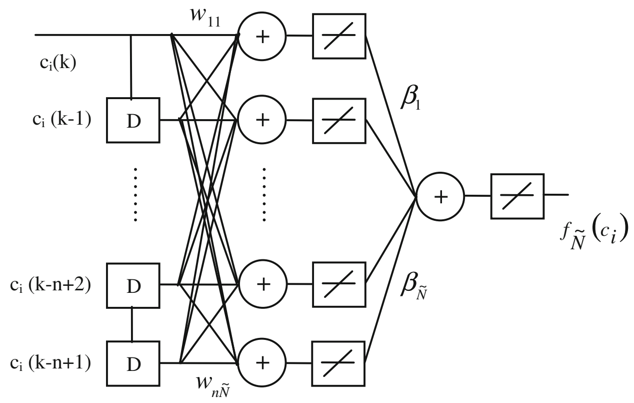

Fig. 5 A single hidden layer feedforward neural network with linear nodes and an input tapped delay line

Remark 4.1 The FIR-ELM algorithm requires that the hidden layer weights be assigned using FIR filter design techniques to reduce disturbances in the data. Hence, given that it is possible to have prior knowledge of the frequency responses from the training datasets, appropriate hidden layer weights can be designed.

Without loss of generality a low-pass filter for the kth hidden layer node can be represented in time domain as

$$\hat{h}_i(l) = \frac{1}{2\pi} \int_{-\omega_c}^{\omega_c} e^{-j\omega(n-1)/2} e^{j\omega l} d\omega = \frac{\sin(\omega_c(l - (n-1)/2))}{\pi(l - (n-1)/2)} \tag{17}$$

for $0 \leq l \leq n - 1$ where n is the filter length and ω_c is the cutoff frequency. The filter coefficients in (17) can then be assigned in the kth hidden layer node as shown in (18)

$$w_{k1} = \hat{h}_i(0), \quad w_{k2} = \hat{h}_i(1), \ldots, w_{kn} = \hat{h}_i(n-1) \tag{18}$$

It is also possible to design other types of filters such as the band-pass and high-pass filters depending on the requirement of the dataset.

The optimal output weights for the FIR-ELM are then calculated based on the minimization of the norms of the output error and the output weights matrix, with two risk balancing constants γ and d introduced to balance the empirical and structural risk. The output weights can be obtained as

$$\beta = \left(\frac{d}{\gamma} I + H^T H\right)^{-1} H^T T \tag{19}$$

The FIR-ELM algorithm can be summarized as:

1. Given a training dataset $[C, T]$, design the hidden layer weights of the SLFN as in (17)
2. Calculate the hidden layer output H
3. Solve for β using (19).

4.2 Frequency domain gene feature selection

It is usually hard to define specific frequency specifications to filter the microarray gene expression data even when prior knowledge of the frequency response is available as they contain many components of similar magnitude. Figure 6 shows an example of the frequency response for a sample in the colon tumor dataset. Therefore, in order to analyse the frequency profiles of the respective datasets, an exhaustive FIR filter design search algorithm called frequency domain gene feature selection (FGFS) is proposed. In this paper, a frequency profile is defined as a collection of filter design and their respective classification performance.

First, the N samples in a dataset are divided into training and testing sets, p_1 and p_2. Then, within the training set p_1,

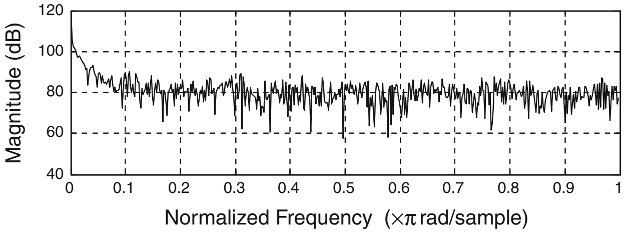

Fig. 6 Frequency response of a sample from the colon dataset

the samples are split once more into subsets q_1 and q_2 specifically for filter design selection. The subsets q_1 and q_2 will be used iteratively to evaluate the suitability of different FIR filter designs for the dataset over a range of normalized cutoff frequencies from 0.1 to 0.9 with a step size of 0.1. A band width of ±0.05 is assigned for the band-pass filter. Finally, the frequency profile of the dataset can be generated from the testing accuracies achieved for each filter design using the training samples $p_1 = q_1 + q_2$. The best performing combination of FIR filter design is then selected to train the FIR-ELM using all the samples in p_1, and the trained classifier is tested on samples in p_1. A flow chart of the algorithm is given in Fig. 7.

Remark 4.2 In the above, we have developed a gene feature selection algorithm that preserves the microarray vectors and utilizes all the genes within a sample for classification. Hence, it is different from the conventional gene selection algorithm that selects subsets of genes. The proposed method is more robust in terms of handling the noise that may severely affect parts of the gene expression readings, such as experimental errors that produce outliers and other disturbances that may cause parts of the sample to be unusable.

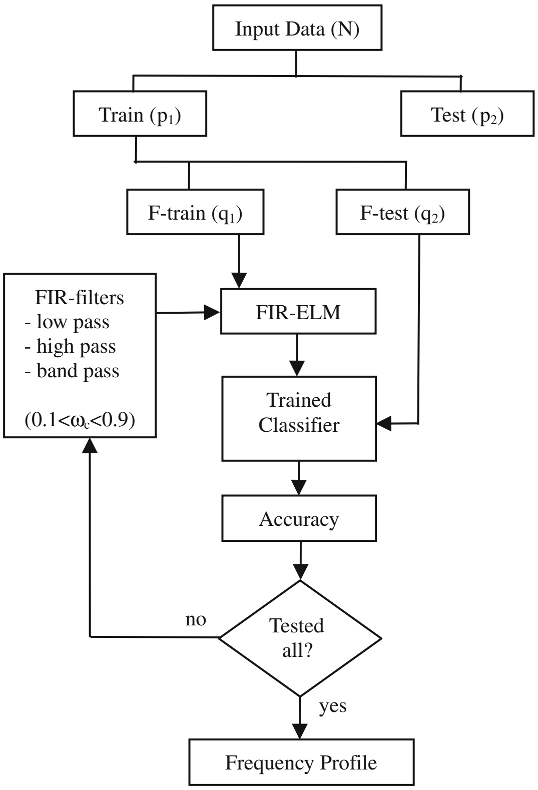

Fig. 7 An FIR filter design search algorithm for FIR-ELM

5 Experiments and results

The performance of the FIR-ELM with the FGFS algorithm is investigated in this section for both the leukemia and colon tumor datasets. The classification results will then be compared with popular algorithms such as the MLP, ELM, and SVM. A conventional frequency-based gene selection process known as the discrete cosine transform (DCT) is implemented for the MLP, ELM, and SVM algorithms to compare the two different approaches to gene feature selection. Lastly, the linear separability of the hidden layer output of the SLFN for ELM and FIR-ELM is discussed.

5.1 Biomedical datasets

Two binary microarray gene expression datasets are investigated, namely, leukemia and colon tumor from the Kent Ridge biomedical data repository [30]. The leukemia dataset consists of two classes of acute leukemia known as acute lymphoblastic leukemia (ALL), arising under lymphoid precursors, and acute myeloid leukemia (AML), arising under myeloid precursors. There are 72 bone marrow samples in the dataset with 47 ALL and 25 AML cases and each contain 7129 gene probes. For the colon tumor dataset, a total of 62 samples were collected from colon-cancer patients, where 40 biopsies are tumors and 22 others are normal samples from healthy parts of the colon.

As conventional classifiers tend to have problems classifying microarray data due to the high number of variables [1, 3], a frequency transformation-based gene feature selection method known as the DCT will be used to perform feature selection for the MLP, ELM, and SVM tests. The DCT is a well-known method in pattern recognition to compress the energy in a sequence, and it has been successfully implemented in the detection of stomach cancer [31]. The DCT for one-dimensional array is defined in (20)

$$y(k) = d(k) \sum_{n=1}^{N} x(n) \cos\left(\frac{\pi(2n-1)(k-1)}{2N}\right)$$

$$k \in 1, 2, \ldots, N \tag{20}$$

$$d(k) = \begin{cases} \frac{1}{\sqrt{N}} & k = 1 \\ \sqrt{\frac{2}{N}} & 2 \leq k \leq N \end{cases}$$

where N is the length of the array sequence x.

The DCT generates coefficients that will then be used as input data for evaluating the performance of the MLP, ELM, and SVM algorithms. In order to select the most relevant coefficients, the 90% criterion is employed to select coefficients that represent 90% of the total energy. Although a lower percentage can be chosen, this criterion is selected to avoid losing too much information from the dataset and reducing the classification performance. A summary of properties for the datasets and the DCT feature selection is presented in Table 4.

5.2 Experimental settings

As the classification of microarray data concerns the critical diagnosis of cancer, the misclassification rate for each class must also be minimized; hence, both the classification accuracy and minimum sensitivity are usually considered [15, 32]. The sensitivity is defined as the number of correct patterns predicted to be in a class with respect to the total number of patterns in the class. The minimum sensitivity is selected from the class with the lowest sensitivity measure within the confusion matrix. In order to properly select the meta-parameters for each classification algorithm, a classification performance measurement based on the classification accuracy (A) and minimum sensitivity (MS) is used. For each of the meta-parameter $M = [M_1, M_2, \ldots, M_l]$ considered, the optimal value $\hat{M}$ is selected from (21)

$$\hat{M} = \arg\max_{M_i} \frac{\bar{A}_{M_i} + \overline{MS}_{M_i}}{2} \quad i \in 1, 2, \ldots, l \tag{21}$$

where $\bar{A}_{M_i}$ is the mean of A, and $\overline{MS}_{M_i}$ is the mean of MS. All the parameters are evaluated using the repeated 10-fold stratified cross-validation (CV) process with the training data only. The CV process is repeated 20 times for each considered parameter.

For both the MLP and ELM, the number of neurons is considered from 1 to 200 in increments of 1, the sigmoidal activation function is used in the hidden layer, and the linear activation function is used in the output layer. The scaled conjugate gradient algorithm is used to train the MLP. The linear kernel is implemented in the SVM after testing using several other popular kernels such as the RBF and polynomial gave poor results. The regularization parameter C for the SVM is considered within the range of 10^{-6} to 10^{6} with logarithmic increments of 1. The FIR-ELM filter length is similar to the dimensions of the gene sample, and the regularization parameters are selected as $\gamma = 1$ and $d = 0.01$ based on the authors' prior knowledge in [18]. Lastly, the targets for both datasets are defined as $[1, -1]$.

Table 4 Summary of leukemia and colon dataset

Dataset	Samples	Genes	DCT Coef.
Leukemia	72	7,129	2,325
Colon	62	2,000	613

A repeated twofold stratified CV is implemented to train the classifier algorithms after the meta-parameter selection process is completed. The CV cycle is repeated 20 times for each algorithm to obtain the mean classification accuracy. The confusion matrices show the mean number of correctly classified as well as misclassified samples for each algorithm.

5.3 Leukemia dataset

The frequency profile of the leukemia dataset for the low-pass, high-pass, and band-pass filters are shown in Figs. 8, 9, and 10, respectively, with error bars showing the standard deviation of the classification performance. The optimal filter design based on (21) is the high-pass filter with a mean normalized cutoff frequency of 0.29. However, it is not possible to state which filter type is better due to the large standard deviations in the frequency profile plots. Instead, the result shows that at each iteration of testing, the optimal filter design is based on the selection of samples for training. Different filtering criteria may be derived in the meta-parameter selection process based on the training samples. The possibility of using different filter designs to produce similar classification performance indicates that different filter design criteria produce vastly different data patterns that can still be mapped by the output layer of the SLFN. Therefore, the selection of the appropriate filter remains subjective and dependent on the classification requirements (e.g., Type of noise present).

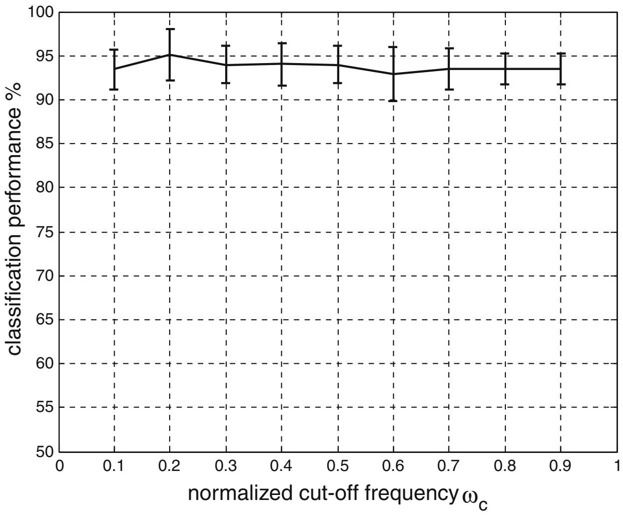

Fig. 8 Classification performance for leukemia with low-pass filter

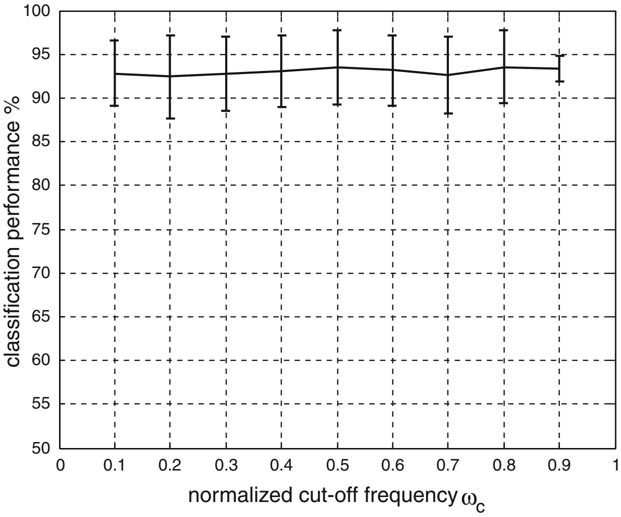

Fig. 9 Classification performance for leukemia with high-pass filter

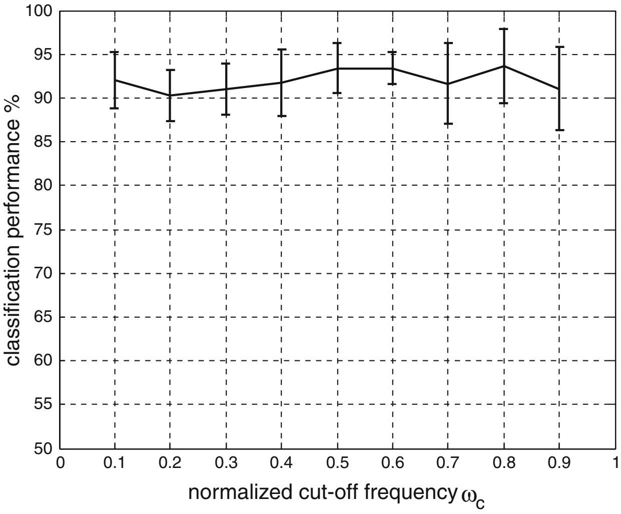

Fig. 10 Classification performance for leukemia with band-pass filter

Table 5 Classification performance for leukemia dataset

Algorithm	MLP	SVM	ELM	FIR-ELM	FIR-ELM (R)
Accuracy (%)	88.01	95.50	76.90	**96.53**	94.49
SD (%)	3.78	2.42	6.39	**1.79**	1.91
Time (s)	10.87	0.66	**0.4**	1.12	1.12

(R): FIR-ELM with random gene order

The classification performance in Table 5 shows that the FIR-ELM has achieved the best result, with an accuracy of 96.53% and a standard deviation of 1.79% which is better than the benchmark of the SVM. The worst performing algorithm is the ELM with an accuracy of 76.90% and the largest standard deviation. The confusion matrix is shown in Table 6, where the ALL cases are labeled as class 1 and AML cases are labeled as class 2. The FIR-ELM has the most similar sensitivities for both cases. From the meta-parameter selection process, the number of neurons for the MLP is 6, the ELM requires 174 neurons, and the SVM regularization parameter is 0.0046.

5.4 Colon tumor dataset

The frequency profiles for the colon tumor dataset using the low-pass, high-pass, and band-pass filters are shown in Figs. 11, 12, and 13, with error bars showing the standard deviation of the classification performance. The optimal filter design based on (21) is the high-pass filter with a mean normalized cutoff frequency of 0.47. Similar to the leukemia dataset, it is not possible to state which filter type is better due to the large standard deviations in the frequency profile plots. The classification performance of the colon tumor dataset is then presented in Table 7. The SVM achieves the highest mean accuracy followed by the FIR-ELM, which is the best performing algorithm among the neural network-based algorithms. However, due to the large standard deviations of the classification accuracy, it is indeed impossible to declare a best classifier for the colon tumor dataset. From the meta-parameter selection process, the number of neurons for the MLP is 42, the number of neurons for the ELM is 142, and the SVM regularization parameter is 0.0082.

The confusion matrix for the colon tumor dataset is shown in Table 8. The tumor cells are labeled as class 1 and the healthy cells are labeled as class 2. It can be seen from Table 8 that the SVM has the best sensitivity for class 1, while the FIR-ELM has the best sensitivity for class 2.

6 Discussions

Overall, the performance of the FIR-ELM has been shown to achieve comparable or better results in both the leukemia and colon tumor classification problems. While the design of the FIR filters remain as an art and is still widely open to interpretation, the method proposed in this paper gives a straightforward suggestion based on the conventional training of neural networks. For both microarray datasets, the ELM is seen to be the fastest followed by SVM, FIR-ELM, and MLP. The time recorded represents 20 iterations of training and testing for each algorithm. The results for the randomly permuted gene order case for both datasets have shown that the classification accuracy remains similar to that of the original gene order. Based on these results, the FIR-ELM with FGFS seems to be

Table 6 Confusion matrix for classification of leukemia dataset

Algorithm	MLP		SVM		ELM		FIR-ELM		FIR-ELM (R)	
Prediction	1	2	1	2	1	2	1	2	1	2
1	44.25	5.9	46.2	2.45	39.05	8.65	46.63	2.12	46	3
2	2.75	19.1	0.8	22.55	7.95	16.35	0.37	22.88	1	22
Sen* (%)	94.15	76.40	98.30	90.20	83.09	65.40	**99.20**	**91.50**	97.87	88.00

(R): FIR-ELM with random gene order, * Sen: sensitivity

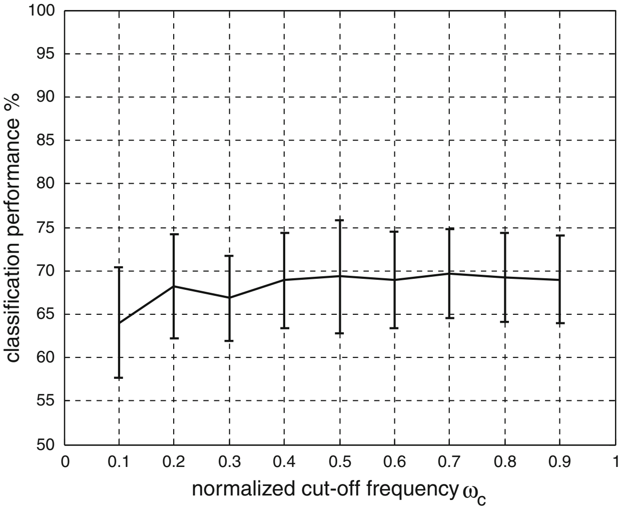

Fig. 11 Classification performance for colon with low-pass filter

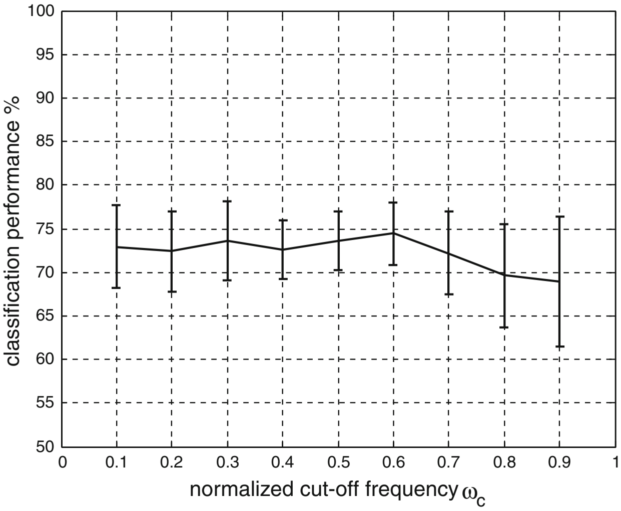

Fig. 12 Classification performance for colon with high-pass filter

insensitive to the gene ordering and is capable of learning from different variants of the dataset. This is acceptable as large standard deviations have been observed in Figs. 8, 9, 10, 11, 12, and 13, which indicates that the FGFS algorithm adapts to the sample characteristic in selecting the optimal filter design.

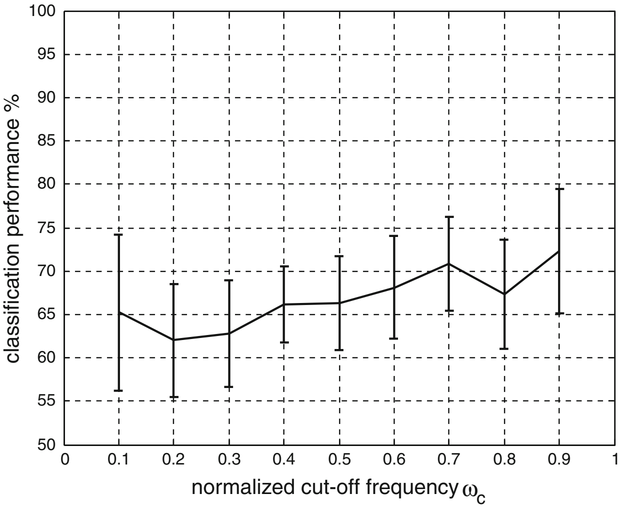

Fig. 13 Classification performance for colon with band-pass filter

Table 7 Classification performance for colon tumor dataset

Algorithm	MLP	SVM	ELM	FIR-ELM	FIR-ELM (R)*
Accuracy (%)	69.76	**79.76**	71.53	76.85	76.61
SD (%)	8.52	**3.57**	6.09	6.18	4.1
Time (s)	5.72	0.41	**0.12**	22.74	22.74

(R): FIR-ELM with random gene order

6.1 Linear separability of the hidden layer output for SLFN

Without loss of generality, SLFNs typically utilize the hidden layer as a pre-processor to map the input data into the desired feature space so that the data points will be easily separable. The output layer then maps the features to the target classes. In order to compare the performance of the chosen hidden layer weights for the ELM and FIR-ELM, the LS gene pair testing algorithm is implemented for the outputs of the hidden layer for both the classifiers, where the outputs of the hidden layer are defined in (22) as

Table 8 Confusion matrix for classification of colon tumor dataset

Algorithm	MLP		SVM		ELM		FIR-ELM		FIR-ELM (R)	
Prediction	1	2	1	2	1	2	1	2	1	2
1	30.3	9.05	34.50	7.05	39.05	8.65	32.45	6.8	32.5	7
2	9.7	12.95	5.55	14.95	7.95	16.35	7.55	15.2	7.5	15
Sen* (%)	75.75	58.86	**86.25**	67.95	78.13	59.55	81.13	**69.09**	81.25	68.18

(R): FIR-ELM with random gene order, * Sen: sensitivity

$$H_{additive} = \begin{bmatrix} w_1 \cdot c_1 + b_1 & \cdots & w_{\tilde{N}} \cdot c_1 + b_{\tilde{N}} \\ \vdots & \ddots & \vdots \\ w_1 \cdot c_N + b_1 & \cdots & w_{\tilde{N}} \cdot c_N + b_{\tilde{N}} \end{bmatrix} \quad (22)$$

It is seen that (22) omits the activation function from the earlier defined hidden layer output (16). This is because the sigmoid activation functions would squish the outputs into a much smaller range and therefore discard the original mappings of the hidden layer weights.

Using the same allocation of samples as in Sect. 3, Table 9 shows the linearly separable gene pair testing results of the hidden layer outputs for ELM and FIR-ELM. It can be seen from Table 9 that the hidden layer of the FIR-ELM reveals more LS gene pairs compared with the ELM. The empirical results achieved suggest that the hidden layer design of the FIR-ELM improves the performance in terms of feature discovery, and it is consistent with the improved classification accuracy obtained for the leukemia and colon dataset. It should be noted that this result is only comparable relatively between the two classifiers under our constraints.

However, when the results in Table 9 are compared with Table 3 that shows the LS pairs for the original dataset, the positive correlation between number of LS pairs and classification accuracy does not hold. It is seen that the LS pairs for the leukemia dataset have increased while the LS pairs for the colon dataset decreased. Ideally, the number of LS pairs is expected to increase to indicate the discovery of more features at the hidden layer output. This may be due to the transformation of the original data into the feature space that inhibits direct comparisons.

From the results obtained, it can be concluded that the positive correlation between the number of LS pairs and classification accuracy holds only when comparing different SLFN training algorithms. The hidden layer output of SLFNs need not be more linear separable than the original dataset to achieve good classification performance. The criterion above could find many applications in the selection of the optimal hidden layer weights for SLFNs.

Table 9 Linearly separable gene pairs for the hidden layer output in ELM and FIR-ELM for leukemia and colon datasets

Leukemia		Colon tumor	
ELM	FIR-ELM	ELM	FIR-ELM
17	35420	3	33

7 Conclusion

In this paper, the FIR-ELM has been implemented for the binary classification of two biomedical datasets. It has been seen that the microarray gene expression samples are treated as time series to form the input patterns for the classification with the FIR-ELM. To assign the optimal input weights of the neural classifier, a frequency domain gene feature selection (FGFS) algorithm has been proposed to evaluate the suitability of different FIR filter designs. For both the leukemia and colon tumor datasets, the FIR-ELM with the FGFS has shown a better performance compared with the other existing algorithms. It has been further shown in the simulation section that the FIR-ELM achieves much better results in terms of gene feature discovery as compared to the ELM. Some future works on the testing of more filter types for the hidden layer designs of the neural classifier and the extension to the multi-class pattern classifications are under the authors' consideration.

References

1. Dudoit S, Fridlyand J (2002) Introduction to classification in microarray experiments. In: Berrar D, Dubitzky W, Granzow M (eds) A practical approach to microarray data analysis. Kluwer, Boston
2. Lu Y, Han J (2003) Cancer classification using gene expression data. Inform Syst 28(4):243–268
3. Huber W, Heydebreck AC, Vingron M (2003) Analysis of microarray gene expression data. In: Bishop M et al (eds) Handbook of statistical genetics. Wiley, Chichester
4. Misra J, Schmitt W, Hwang D, Hsiao L, Gullans S, Stephanopoulos G (2002) Interactive exploration of microarray gene expression patterns in a reduced dimensional space. Genome Res 12(7):1112–1120
5. Wall ME, Rechtsteiner A, Rocha LM (2003) Singular value decomposition and principal component analysis. In: Berrar DP, Dubitzky W, Granzow M (eds) A practical approach to microarray data analysis. Kluwer, Norwell, pp 91–109

6. Liao X, Dasgupta N, Lin SM, Carin L (2002) ICA and PLS modelling for functional analysis and drug sensitivity for DNA microarray signals. In Proceedings of workshop on genomic signal processing and statistics
7. Chen A, Hsu J-C (2010) Exploring novel algorithms for the prediction of cancer classification. In: 2nd international conference on software engineering and data mining (SEDM), pp 378–383
8. Zhang R, Huang G-B, Sundararajan N, Saratchandran P (2007) Multicategory classification using an extreme learning machine for microarray gene expression cancer diagnosis. IEEE/ACM Trans Comput Biol Bioinform 4(3):485–495
9. Ramaswamy S, Tamayo P, Rifkin R, Mukherjee S, Yeang C-H, Angelo M, Ladd C, Reich M, Latulippe E, Mesirov J, Poggio T, Gerald W, Loda M, Lander E, Golub T (2002) Multiclass cancer diagnosis using tumor gene expression signatures. Proc Natl Acad Sci USA 98(26):15149–15154
10. Baboo D, Sasikala M (2010) Multicategory classification using support vector machine for microarray gene expression cancer diagnosis. Global J Comput Sci Technol
11. Vapnik VN (1999) The nature of statistical learning theory, 2nd edn. Springer, New York
12. Abe S (2005) Support vector machines for pattern classification. Springer, London
13. Huang G-B, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70:489–501
14. Helmy T, Rasheed Z (2009) Multi-category bioinformatics dataset classification using extreme learning machine. Evolutionary computation, 2009. CEC '09. IEEE congress on, pp 3234–3240
15. Sanchez-Monedero J, Cruz-Ramirez M, Fernandez-Navarro F, Fernandez J, Gutierrez P, Hervas-Martinez C (2010) On the suitability of extreme learning machine for gene classification using feature selection. Intelligent systems design and applications (ISDA), 2010 10th international conference on, pp 507–512
16. Baboo S, Sasikala S (2010) Multicategory classification using an Extreme Learning Machine for microarray gene expression cancer diagnosis. Communication control and computing technologies (ICCCCT), 2010 IEEE international conference on, pp 748–757
17. Bharathi A, Natarajan A (2010) Microarray gene expression cancer diagnosis using machine learning algorithms. Signal and image processing (ICSIP), 2010 international conference on, pp 275–280
18. Man Z, Lee K, Wang D, Cao Z, Miao C (2011) A new robust training algorithm for a class of single-hidden layer feedforward neural networks. Neurocomputing 74(16):2491–2501
19. Diniz PSR, Silva EABD, Netto SL (2002) Digital signal processing system analysis and design. Cambridge University Press, Cambridge
20. Unger G, Chor B (2010) Linear separability of gene expression data sets. IEEE/ACM Trans Comput Biol Bioinform 7(2):375–381
21. Brody JP, Williams BA, Wold BJ, Quake SR (2002) Significance and statistical errors in the analysis of DNA microarray data. Proc Natl Acad Sci USA 99(20):12975–12978
22. Arce G, Li Y (2002) Median power and median correlation theory. IEEE Trans Signal Process 50(11):2768–2776
23. Salakhutdinov R (2009) Learning in Markov random fields using tempered transitions. In: Bengio Y, Schuurmans D, Lafferty J, Williams C, Culota A (eds) Advances in neural information processing systems, 22. MIT Press, Cambridge
24. Yang L, Yan H, Dong YX, Fei LY (2010) A kind of correlation classification distance of whole phase based on weight. Environmental science and information application technology (ESIAT), 2010 international conference on, 3: 668–671
25. Chatfield C (2004) The analysis of time series: an introduction. 6th Ed, Chapman and Hall
26. Ben-Dor A, Bruhn A, Friedman N, Nachman I, Schummer M, Yakhini Z (2000) Tissue classification with gene expression profiles. J Computational Biol 7(3/4):559–583
27. Mukherjee S, Tamayo P, Rogers S, Rifkin R, Engle A, Campbell C, Golub TR, Mesirov JP (2003) Estimating dataset size requirements for classifying DNA microarray data. J Comput Biol 10(2):119–142
28. Miche Y, Bas P, Jutten C, Simula O, Lendasse A (2008) A methodology for building regression models using extreme learning machine: OP-ELM. In: ESANN 2008, European symposium on artificial neural networks, Bruges, Belgium
29. Huang G-B, Chen L, Siew CK (2006) Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans Neural Netw 17:879–892
30. Li J, Liu H (2004) Kent ridge bio-medical data set repository. School of Computer Engineering, Nanyang Technological University, Singapore, 2004. Online available: http://levis.tongji.edu.cn/gzli/data/mirror-kentridge.html
31. Sarhan AM (2009) Cancer classification based on microarray gene expression data using DCT and ANN. J Theoretical Appl Inform Technol (JATIT) 6(2):208–216
32. Ali AH (2008) Self-organization maps for prediction of kidney dysfunction. In Proceedings of 16th Telecommunications Forum TELFOR, Belgrade, Serbia