

Journal of Healthcare Engineering

Machine Learning Theory and Applications for Healthcare

Lead Guest Editor: Ashish Khare

Guest Editors: Moongu Jeon, Ishwar K. Sethi, and Benlian Xu





Machine Learning Theory and Applications for Healthcare

Journal of Healthcare Engineering

Machine Learning Theory and Applications for Healthcare

Lead Guest Editor: Ashish Khare

Guest Editors: Moongu Jeon, Ishwar K. Sethi, and Benlian Xu



Copyright © 2017 Hindawi. All rights reserved.

This is a special issue published in “Journal of Healthcare Engineering.” All articles are open access articles distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Editorial Board

Saverio Affatato, Italy
William Bertucci, France
Tiago H. Falk, Canada
Mostafa Fatemi, USA
Jesus Favela, Mexico
Joseph Finkelstein, USA
Antonio Gloria, Italy
Philippe Gorce, France

Andreas H. Hielscher, USA
Zhongwei Jiang, Japan
John S. Katsanis, Greece
Feng-Huei Lin, Taiwan
Maria Lindén, Sweden
Francisco Lopez-Valdes, Spain
Andreas Maier, Germany
Rafael Morales, Spain

David Moratal, Spain
Jose Joaquin Rieta, Spain
Hélder A. Santos, Finland
Emiliano Schena, Italy
Maurizio Schmid, Italy
Jiann-Shing Shieh, Taiwan
Vinoy Thomas, USA
Ioannis G. Tollis, Greece

Contents

Machine Learning Theory and Applications for Healthcare

Ashish Khare, Moongu Jeon, Ishwar K. Sethi, and Benlian Xu
Volume 2017, Article ID 5263570, 2 pages

Twin SVM-Based Classification of Alzheimer's Disease Using Complex Dual-Tree Wavelet Principal Coefficients and LDA

Saruar Alam, Goo-Rak Kwon, Ji-In Kim, and Chun-Su Park
Volume 2017, Article ID 8750506, 12 pages

R Peak Detection Method Using Wavelet Transform and Modified Shannon Energy Envelope

Jeong-Seon Park, Sang-Woong Lee, and Unsang Park
Volume 2017, Article ID 4901017, 14 pages

Hybrid Disease Diagnosis Using Multiobjective Optimization with Evolutionary Parameter Optimization

MadhuSudana Rao Nalluri, Kannan K., Manisha M., and Diptendu Sinha Roy
Volume 2017, Article ID 5907264, 27 pages

Diagnosis of Alzheimer's Disease Based on Structural MRI Images Using a Regularized Extreme Learning Machine and PCA Features

Ramesh Kumar Lama, Jeonghwan Gwak, Jeong-Seon Park, and Sang-Woong Lee
Volume 2017, Article ID 5485080, 11 pages

Cell Detection Using Extremal Regions in a Semisupervised Learning Framework

Nisha Ramesh, Ting Liu, and Tolga Tasdizen
Volume 2017, Article ID 4080874, 13 pages

Low-Grade Glioma Segmentation Based on CNN with Fully Connected CRF

Zeju Li, Yuanyuan Wang, Jinhua Yu, Zhifeng Shi, Yi Guo, Liang Chen, and Ying Mao
Volume 2017, Article ID 9283480, 12 pages

A Real-Time Analysis Method for Pulse Rate Variability Based on Improved Basic Scale Entropy

Yongxin Chou, Ruilei Zhang, Yufeng Feng, Mingli Lu, Zhenli Lu, and Benlian Xu
Volume 2017, Article ID 7406896, 10 pages

Patient-Specific Deep Architectural Model for ECG Classification

Kan Luo, Jianqing Li, Zhigang Wang, and Alfred Cuschieri
Volume 2017, Article ID 4108720, 13 pages

Editorial

Machine Learning Theory and Applications for Healthcare

Ashish Khare,¹ Moongu Jeon,² Ishwar K. Sethi,³ and Benlian Xu⁴

¹*Department of Electronics and Communication, University of Allahabad, Allahabad, India*

²*School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju, Republic of Korea*

³*Department of Computer Science and Engineering, School of Engineering and Computer Science, Oakland University, Rochester, MI, USA*

⁴*School of Electrical and Automatic Engineering, Changshu Institute of Technology, Changshu, China*

Correspondence should be addressed to Ashish Khare; ashishkhare@hotmail.com

Received 24 July 2017; Accepted 25 July 2017; Published 27 September 2017

Copyright © 2017 Ashish Khare et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The explosive growth of health-related data presented unprecedented opportunities for improving health of a patient. Machine learning plays an essential role in healthcare field and is being increasingly applied to healthcare, including medical image segmentation, image registration, multimodal image fusion, computer-aided diagnosis, image-guided therapy, image annotation, and image database retrieval, where failure could be fatal.

The purpose of this special issue is to advance scientific research in the broad field of machine learning in healthcare, with focuses on theory, applications, recent challenges, and cutting-edge techniques.

The quality level of the submissions for this special issue was very high. A total of 24 manuscripts were submitted to this issue in response to the call for papers. Based on a rigorous review process, 8 papers (33%) were accepted for the publication in the special issue. Below, we briefly summarize the highlights of each paper.

One of the papers of this special issue, “Diagnosis of Alzheimer’s Disease Based on Structural MRI Images Using a Regularized Extreme Learning Machine and PCA Features,” R. K. Lama et al. proposes a method and compared Alzheimer disease (AD) diagnosis approaches using structural magnetic resonance (sMR) images to discriminate AD, mild cognitive impairment, and healthy control subjects using a support vector machine (SVM), an import vector machine (IVM), and a regularized extreme learning machine (RELM). By means of experiments on the ADNI datasets, it has been

concluded that RELM with the feature selection approach can significantly improve classification accuracy of AD from mild cognitive impairment and healthy control subjects.

S. Alam et al. presented a method for distinguishing AD from healthy control using combination of dual-tree complex wavelet transforms, principal coefficients from the transaxial slices of MRI images, linear discriminant analysis, and twin support vector machine in their article “Twin SVM-Based Classification of Alzheimer’s Disease Using Complex Dual-Tree Wavelet Principal Coefficients and LDA.”

A semisupervised learning approach for cell detection is presented in the third article of this special issue by N. Ramesh et al. in their paper entitled “Cell Detection Using Extremal Regions in a Semisupervised Learning Framework.” The method requires very few examples of cells with simple dot annotations for training.

In the paper entitled “Patient-Specific Deep Architectural Model for ECG Classification” by K. Luo et al., a method for ECG classification is proposed. The method is based on time-frequency representation and patient-specific deep learning architectural model, and it uses deep neural network classifier.

An automatic method for segmentation of 3D magnetic resonance imaging (MRI) data, useful in the clinical diagnosis of brain tumor, named as Glioma is presented by Z. Li et al. in the article “Low-Grade Glioma Segmentation Based on CNN with Fully Connected CRF.” The method combined a multipathway convolutional neural network (CNN) and

fully connected conditional random field (CRF). Experimental results have shown that the method is useful for low-grade glioma.

A general system for hybrid disease diagnosis adopting classifier optimization procedure using evolutionary algorithms is presented by M. R. Nalluri et al. in their article “Hybrid Disease Diagnosis Using Multiobjective Optimization with Evolutionary Parameter Optimization.”

Y. Chou et al., in their paper “A Real-Time Analysis Method for Pulse Rate Variability Based on Improved Basic Scale Entropy,” proposed a method named sliding window iterative base scale entropy analysis by combining base scale entropy analysis and sliding window iterative theory for analyzing heart rate variability signal.

Another paper of this special issue by J.-S. Park et al. titled “R Peak Detection Method Using Wavelet Transform and Modified Shannon Energy Envelope” presents an R peak detection method using the wavelet transform and a modified Shannon energy envelope for rapid ECG analysis.

These 8 selected contributions basically can reflect the new achievements in the machine learning applications in healthcare, and we hope they can provide a solid foundation for future new approaches and applications.

Acknowledgments

We would like to thank all authors who submitted their work for consideration in our special issue.

*Ashish Khare
Moongu Jeon
Ishwar K. Sethi
Benlian Xu*

Research Article

Twin SVM-Based Classification of Alzheimer's Disease Using Complex Dual-Tree Wavelet Principal Coefficients and LDA

Saruar Alam,¹ Goo-Rak Kwon,¹ Ji-In Kim,¹ and Chun-Su Park²

¹Department of Information and Communication Engineering, Chosun University, 375 Seosuk-Dong, Dong-Gu, Gwangju 501-759, Republic of Korea

²Department of Computer Education, Sungkyunkwan University, 05006 209 Neungjong-ro, Gwangjin-gu, Seoul, Republic of Korea

Correspondence should be addressed to Goo-Rak Kwon; grkwon@chosun.ac.kr

Received 5 November 2016; Revised 22 February 2017; Accepted 27 April 2017; Published 16 August 2017

Academic Editor: Ashish Khare

Copyright © 2017 Saruar Alam et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Alzheimer's disease (AD) is a leading cause of dementia, which causes serious health and socioeconomic problems. A progressive neurodegenerative disorder, Alzheimer's causes the structural change in the brain, thereby affecting behavior, cognition, emotions, and memory. Numerous multivariate analysis algorithms have been used for classifying AD, distinguishing it from healthy controls (HC). Efficient early classification of AD and mild cognitive impairment (MCI) from HC is imperative as early preventive care could help to mitigate risk factors. Magnetic resonance imaging (MRI), a noninvasive biomarker, displays morphometric differences and cerebral structural changes. A novel approach for distinguishing AD from HC using dual-tree complex wavelet transforms (DTCWT), principal coefficients from the transaxial slices of MRI images, linear discriminant analysis, and twin support vector machine is proposed here. The prediction accuracy of the proposed method yielded up to 92.65 ± 1.18 over the Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset, with a specificity of 92.19 ± 1.56 and sensitivity of 93.11 ± 1.29 , and 96.68 ± 1.44 over the Open Access Series of Imaging Studies (OASIS) dataset, with a sensitivity of 97.72 ± 2.34 and specificity of 95.61 ± 1.67 . The accuracy, sensitivity, and specificity achieved using the proposed method are comparable or superior to those obtained by various conventional AD prediction methods.

1. Introduction

Alzheimer's disease (AD) is the most familiar cause of dementia, with patients comprising 50%–80% of all dementia sufferers. The disease affects memory, cognition, and behavior. As AD is a neurodegenerative condition, several types of atrophy occur in the hippocampus and other areas of the brain. Despite being the 6th leading cause of death in the USA, it is not a common disease. Currently, there is no cure; however, some preventive measures can be taken to mitigate risk factors and slow the degenerative process. An estimated \$605 billion globally and \$220 billion in USA is spent annually on diagnosing AD. Many people suffer from AD worldwide, and demands on researchers are growing rapidly. MRI is an effective medical image construction technique,

as it has the proven potential to view structural changes in the human brain, internal organs, and other tissues.

MRI produces high-quality structural images, providing distinctive tissue information, which enhances both the accuracy of brain pathology diagnosis and quality of treatment. A key advantage of this technique is its noninvasiveness. Many studies have been conducted using multivariate analysis algorithms and structural/functional MRI to classify neurological diseases [1–3]. A primary focus of these studies was the large dimensionality of extracted features and the identification of disease signatures among them where the most discriminative information of the said diseases exists. Results showed significant cerebral structural changes in several brain ROIs, particularly in the hippocampus and entorhinal cortex [4]. Global and internal intensity-based features, [3, 5], as well

as geometric- and surface-based features [6, 7], have been used in earlier studies for classifying disease. The authors presented an electroencephalogram (EEG) coherence study of Alzheimer’s disease using a probabilistic neural network (PNN) and showed significant accuracy in distinguishing true AD from the control groups [8]. Chaplot et al. [9] stratified AD using discrete wavelet coefficients as a feature for training and testing Support Vector Machines (SVMs) and neural network classifiers. Extracting essential discriminatory features from MRI brain images is imperative for competent analysis of disease diagnosis. The preferred feature extraction methods, amongst those most frequently used, are independent component analysis [10], wavelet transform [11], and Fourier transform [12]. This study has been conducted using discrete wavelet features and the k-nearest neighbor algorithm (k-NN) [11] on an artificial neural network (ANN) [11, 13]. Zhang and Wang [14] ran AD prediction models using displacement field estimation between AD and healthy controls using an SVM, twin support vector machine (TWSVM), and generalized eigenvalue proximal SVM (GEPSSVM) as classifiers. Tomar and Agarwal [15] reviewed several types of twin SVM algorithms, their optimization problems, and their applications.

The biomarkers used in our proposed method are MRI images from the Alzheimer’s Disease Neuroimaging Initiative (ADNI) and Open Access Series of Imaging Studies (OASIS) datasets. Our primary reason for using DTCWT over DWT is its effective representation of singularities (curves and lines), even though DWT has the advantage of representing the functions in multiscale and compressed forms. In DTCWT, shifts in magnitude variance can be achieved to a higher degree [16]. In our proposed method, DTCWT coefficient-based AD classification has been proposed using principal component analysis and linear discriminant analysis of extracted coefficients; a TWSVM was utilized as a supervising technique. Classification performance is documented regarding accuracy, sensitivity, and specificity, after applying 10-fold cross validation and running the program 10–20 times. Our method produced superior results when compared with several conventional AD classification methods.

2. Material and Methods

A total of 172 subjects from the ADNI dataset were used—86 AD and 86 HC. In addition, we used 95 subjects from the OASIS dataset—44 HC and 51 subjects suffering from very mild to mild AD.

2.1. Overview of Experimental Data. Data used in the preparation of this article were obtained from the Alzheimer’s Disease Neuroimaging Initiative (ADNI) database (<http://adni.loni.usc.edu>).

The ADNI was launched in 2003 as a public-private partnership led by Principal Investigator Michael W. Weiner, MD. The primary goal of the ADNI is to test whether serial MRI, positron emission tomography (PET), other biological markers, and clinical and neuropsychological assessment can be combined to measure the progression of MCI and

TABLE 1: Summary of subject’s demographics status.

	AD	Normal
	86	86
Number of subjects	43 males 43 females	46 males 40 females
Average age	77.30	76.05
Average education points	14.65	15.93
MMSE	23.48	29.08

TABLE 2: Statistical OASIS data details used in our learning.

Factors	Normal	Very mild & mild AD
Number of patients	44	51
Age	84.40 (76–96)	82.11 (76–96)
Education	3.34 (1–5)	3.13 (1–5)
Socioeconomic status	2.31 (1–5)	2.82 (1–5)
CDR (0.5/1)	0	35/16
MMSE	28.72 (25–30)	24.82 (18–30)

TABLE 3: Clinical dementia scale.

CDR	Rank
0.5	Very mild dementia
1	Mild
2	Moderate
3	Severe

early-onset Alzheimer’s disease AD. For up-to-date information, visit www.adni-info.org. The demographic details of data used from the ADNI are shown in Table 1.

In addition, we utilized MRI images downloaded from the OASIS dataset. OASIS is a database designed to compile MRI datasets and make them freely accessible to the scientific community. OASIS compiles two types of data: cross-sectional MRI data and longitudinal MRI data. Our study utilized cross-sectional MRI data, as our aims are to develop an automatic system for detecting AD, for which longitudinal MRI data is not optimal.

The OASIS dataset consists of 416 subjects aged between 18 and 96 years. Our study included 51 AD patients (35 with CDR = 0.5 and 16 with CDR = 1) out of 100 having dementia and 44 HC out of 98 normal subjects. Table 2 shows the demographic details of the subjects used in our study. Both men and women are included and all subjects are right handed. The scale of the CDR is listed in Table 3.

2.2. Proposed Approach. The proposed approach is made up of 4 phases: preprocessing and slice extraction, feature extraction, projection of features into lower dimension, and efficient classification of the disease. Figure 1 shows all phases in detail.

2.2.1. Preprocessing and Slice Extraction. All MRI images used for training and testing the TSVM of our proposed approach are viewed using the ONIS toolbox and exported

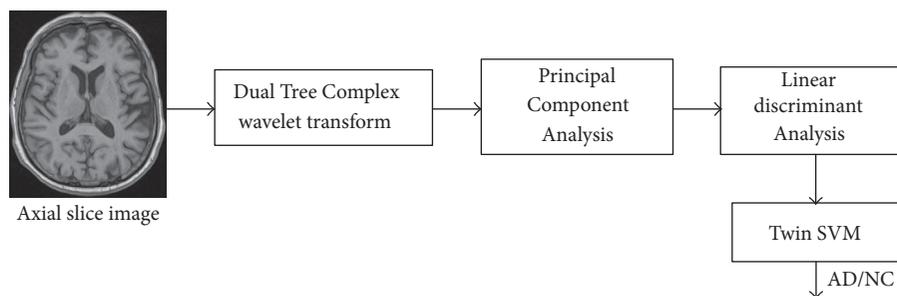


FIGURE 1: Flowchart of DTCWT-based classification performance of AD from HC.

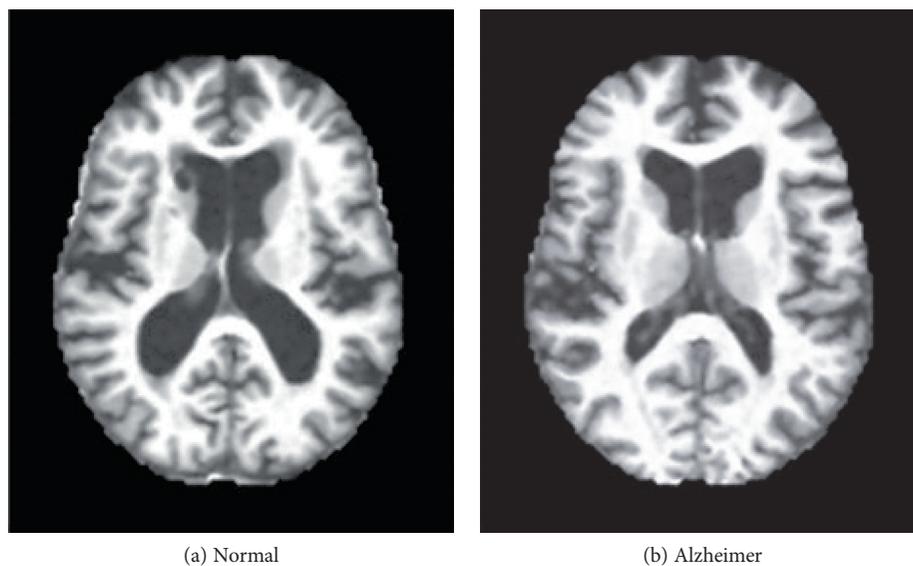


FIGURE 2: MR image slice sample (axial slice view after preprocessing).

as 2D MRI image slices. All images are in PNG format, and the dimensions of OASIS image slices are 176×208 ; the dimensions of the ADNI image slices are 256×166 . The range of selection of those slices was performed manually from the tissue center for information clarity. The images are resized to 256×256 for further processing. A sample of a brain image slice is depicted in Figure 2. LibSVM toolbox was used for kernel SVM simulation in MATLAB.

2.2.2. Dual-Tree Complex Wavelet Transform. Wavelet transform (WT) is one of the most frequently used feature extraction techniques for MR images. For our proposed approach, we extract the DTCWT [16] coefficients from the input MRI images. The features of the 5th resolution scale were used as they produced higher classification performance when compared with other resolution levels. DTCWT has a multiresolution representation, as with CWT. For efficient disease classification, it is preferable to use a few intermediate scales of the extracted coefficients as input to a classifier, as the lowest resolution scales lose fine details and high-resolution scales contain mostly noise. Thus, we prefer to choose a few intermediate scales of DTCWT coefficients. These coefficients were sent as input for principal component analysis (PCA). CWT can be represented as complex-valued scaling

functions and complex-valued wavelets. DTCWT engages two real DWTs, which provide the real and imaginary components of the wavelet transform, respectively. In addition, two filter bank types are set: analysis filter banks and synthesis filter banks. These filter banks are used for implementing DTCWT to ensure that overall transformation becomes almost analytic, as shown in Figure 3.

The DTCWT can be denoted in matrix form as

$$D = [D_h D_g], \quad (1)$$

where D_h and D_g are rectangular matrices.

For the input image x , complex wavelet coefficients can be represented as

$$T_h + jT_g, \quad (2)$$

where $T_h = D_h^* x$ is the real component and $T_g = D_g^* x$ is the imaginary part.

The DTCWT coefficients of input images are shift invariant; they do not change when an image is shifted in time or space. In addition, DTCWT employs segregation of 6 diverse directions (± 15 , ± 30 , and ± 45) for 2D images and 28 different directions for 3D images, while conventional DWT only allows for isolation of horizontal and vertical directions. For

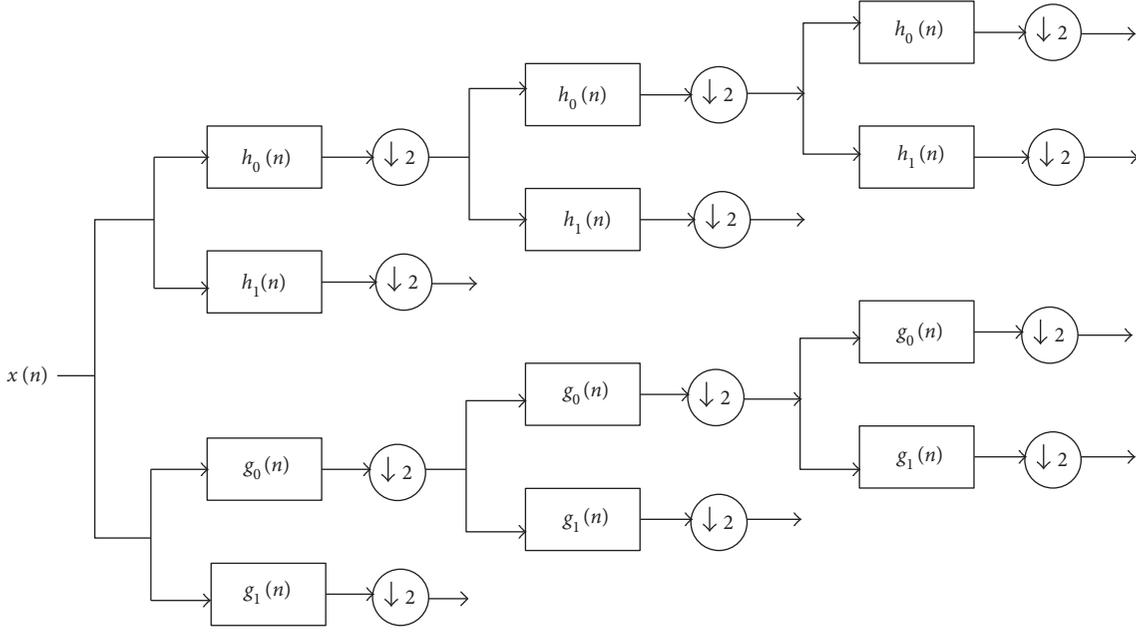


FIGURE 3: Block diagram for a 3-level DTCWT.

each 2D slice subject image, we extracted 5-level DTCWT coefficients from one scale.

2.2.3. Principal Component Analysis. Principal component analysis (PCA) [17] is a dimensionality reduction technique that is applied to map features onto lower dimensional space. This data transformation may be linear or nonlinear. One of most frequently used linear transformation is PCA, which is an orthogonal transformation used to convert possibly correlated samples to linearly uncorrelated variables. The number of principal components is lower than or equal to the number of original variables. The PCA conversion process is shown in Figure 4.

The PCA is summarized as follows:

- (i) Calculating the mean of the data and zero mean data
- (ii) Constructing the covariance matrix
- (iii) Acquiring the eigenvalue and the eigenvector
- (iv) Projecting the data matrix with eigenvectors corresponding to the highest to lowest eigenvalues.

2.2.4. Linear Discriminant Analysis. A generalized Fisher linear discriminant [18] is used for the linear projection of features to separate two or more classes. To make effective and discriminative projected features, PCA coefficients can be projected on to a new LDA projection axis.

To find the class separation projection axis, it is necessary to determine between-class scatter and within-class variability.

The between class variable matrix can be denominated by sample variance as

$$S_B = \frac{1}{c} \sum_{j=1}^c (m_j - m)(m_j - m)^T. \quad (3)$$

Within class variance matrix can be expressed as

$$S_w = \sum_{j=1}^c \sum_{z_k \in w_i} (z_k - m_i)(z_k - m_i)^T, \quad (4)$$

where z_k is k th sample variable belonging to a class.

The generalized Rayleigh coefficient is

$$J(w) = \frac{W^t S_B W}{W^t S_w W}, \quad (5)$$

where W is the matrix for LDA coefficients. This can be characterized using the generalized eigenvalue problem as

$$S_B W = \lambda S_w W, \quad (6)$$

where λ is the eigenvalue.

If S_w is singular matrix, (6) can be simplified as

$$S_w^{-1} S_B W = \lambda W, \quad (7)$$

where the eigenvectors of $S_w^{-1} S_B$ will be W . The eigenvector matrix will be W_{LDA} ,

$$W_{LDA} = [W_1 W_2 W_3 \cdots W_k], k \in Z. \quad (8)$$

The PCA coefficients can be projected onto l lower dimensional LDA projection termed by eigenvectors corresponding nonzero higher energy eigenvalues,

$$W_{LDA}^l = [W_1 W_2 W_3 \cdots W_l], l \in Z, \quad (9)$$

where $l \leq k$.

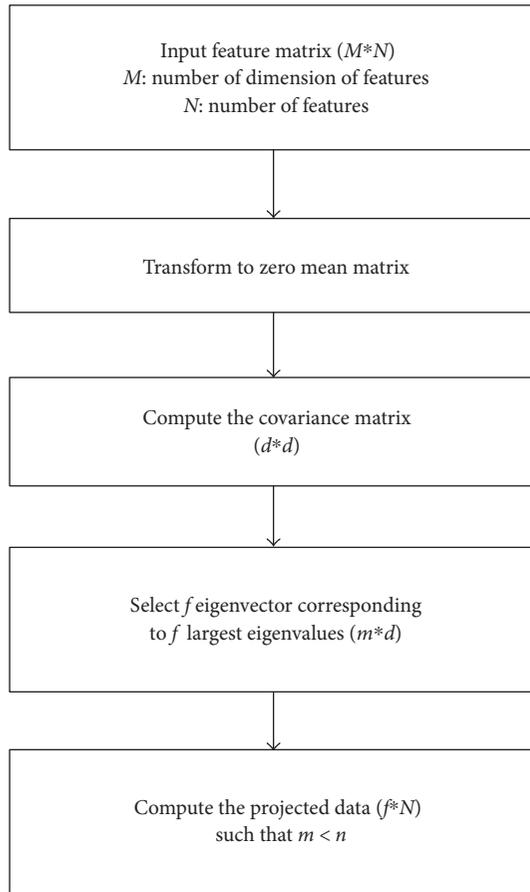


FIGURE 4: PCA implementation for feature reduction.

The final feature matrix F is evaluated as

$$F = \left(W_{LDA} \right)^T \cdot \psi(x)_{pc}. \quad (10)$$

2.2.5. Twin Support Vector Machine. Jayadeva and Chandra [19] proposed a novel dual hyperplane-based variant twin SVM. The concepts of generalized eigenvalues proximal support vector machine (GEP-SVM) are applied here, which require two nonparallel optimum hyperplanes for each class. There are two quadratic programming (QP) problems optimized as TSVM pairs, as in a typical SVM.

Mathematically, the TSVM primal problem can be optimized by solving the following two quadratic programming problems:

$$\begin{aligned} \min_{w_1, b_1, q} & \frac{1}{2} (X_1 w_1 + o_1 b_1)^T (X_1 w_1 + o_1 b_1) + C_1 o_2^T \xi_1 \\ \text{s.t.} & -(X_2 w_1 + o_2 b_1) + \xi_1 \geq o_2, \xi_1 \geq 0, \end{aligned} \quad (11)$$

$$\begin{aligned} \min_{w_2, b_2, q} & \frac{1}{2} (X_2 w_2 + o_2 b_2)^T (X_2 w_2 + o_2 b_2) + C_2 o_1^T \xi_2 \\ \text{s.t.} & -(X_1 w_2 + o_1 b_2) + \xi_2 \geq o_1, \xi_2 \geq 0. \end{aligned} \quad (12)$$

Here, X_i ($i = 1, 2$) are input features, w_i ($i = 1, 2$) are the normal hyperplane vectors, b_i ($i = 1, 2$) are bias terms,

C_i ($i = 1, 2$) are the vectors of positive penalty parameters, o_i ($i = 1, 2$) are the suitable dimensional matrices of ones, and ξ_i ($i = 1, 2$) are the slack variables. Hence, the TSVM finds two hyperplanes, each of which is nearer to the data sample of one class than to that of another. Therefore, minimizing (11) and (12) will compel the hyperplanes to approximate the data of each class and enhance the classification rate. The optimization problem can be solved in the Lagrange duality principle [15].

3. Results and Discussions

3.1. Background. In this article, our proposed approach is presented using Fisher linear discriminant analysis of DTCWT principal components. The details of our proposed method are shown in Figure 1. The advantage of WT over FT is its multiple-scaled representations and frequency components with spatial domain information. Fourier coefficients only produce image frequency information, whereas wavelets contain powerful observations of the spatial and frequency domain in a multiscaled format. In addition, wavelet representation is spatially localized; Fourier functions are not spatially localized as they consist only of image frequency components. MRI images can be represented and processed at numerous resolutions and can therefore be used as an incisive framework for processing multiresolution images. Finally, DWT coefficients can be extracted by using arrays of low and high pass filter banks.

However, there are multiple drawbacks to conventional wavelet transform. These include drift in wavelet coefficient oscillation towards positive and negative around singularities, shift variance of signal (which may cause oscillation of wavelet coefficient samples around singularities), substantial aliasing of amply spaced wavelet coefficient patterns, and lack of directional selectivity perturbs to process and model geometric image features (such as edges and ridges). In these cases, flaws regarding conventional DWT are not experienced by Fourier transform. Inspired by Fourier transform, our improved DTCWT is used to overcome these drawbacks. Previous studies have shown that DTCWT feature-based AD disease detection performs better than typical DWT-based feature extraction [20]. Furthermore, DTCWT produces superior singularities of line and curve representation. Thus, discriminative feature can be extracted comparatively, which is crucial for any pattern classification problem.

Misclassification rates and higher dimensionality of features present problems concerning pattern classification. For smooth classification, dimensionality reduction techniques are employed to transform data from higher to lower dimensional spaces. PCA is the most frequently applied linear transformation and addresses these concerns. Extracted features are analyzed using PCA for feature reduction. For each MRI image from the OASIS and ADNI datasets, there are 49,152 (1536×32) features. After applying PCA, this is reduced to 95×94 for OASIS data and 172×171 for ADNI data.

After PCA, the classification may still not be sufficient, as PCA does not account for variability of features within a class or between classes. To ensure that the PCs are more

TABLE 4: Confusion matrix for a binary classifier to distinguish between two classes (S_1 and S_2).

True class	Predicted class	
	S_1 (patients)	S_2 (controls)
S_1 (patients)	TP	FN
S_2 (controls)	FP	TN

separable, it is needed to transform data onto another space combining directions that will find axes, which will maximize the gap between different classes. Thus, LDA is applied to project PCs onto new projection axes for more effective disease classification.

TSVM is an emerging efficient pattern classification and regression algorithm in machine learning. Numerous studies have shown that TSVM is highly effective in terms of classification, regression performance, and time complexity [19, 21–23]. Hence, we have applied TSVM using linear discriminant DTCWT principal components as input features.

All programs are executed in MATLAB 2015b installed on an Intel (R) Core (TM) i3-4160 CPU system. The time complexity of the extraction of DTCWT and DWT coefficients from a 2D MRI image slice are 0.5148 and 0.5109, respectively. There is no significant difference in CPU-elapsed time when comparing transform methods. As a dimensionality reduction technique, we used PCA to omit higher dimensional input features.

In addition, it is not feasible to train and test a classifier with higher dimensional features due to elapsed time. The CPU-elapsed time to achieve TSVM classification performance was approximately 88.40 seconds without reducing dimensions. The time required for our proposed method is approximately 15.74 seconds—faster than the methods that do not employ fisher discriminant analysis.

3.2. Performance Evaluation. The performance of a binary classifier can be visualized using a confusion matrix, as shown in Table 4. The number of examples correctly predicted by the classifier is located on the diagonal. These may be divided into true positives (TP), representing correctly identified patients, and true negatives (TN), representing correctly identified controls. The number of examples wrongly stratified by the classifier may be divided into false positives (FP), representing controls incorrectly classified as patients, and false negatives (FN), representing patients incorrectly classified as controls.

Accuracy is determined measuring the proportion of examples that are correctly labeled by a classifier:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \quad (13)$$

This may not be an ideal performance metric if the class distribution of the dataset is unbalanced.

For example, if class C_1 is much larger than C_2 , a high accuracy value could be obtained by a classifier that labels all examples as belonging to class C_1 . Sensitivity is the rate of true positives (TP), and specificity is the rate of true negatives (TN). Sensitivity and specificity are defined as

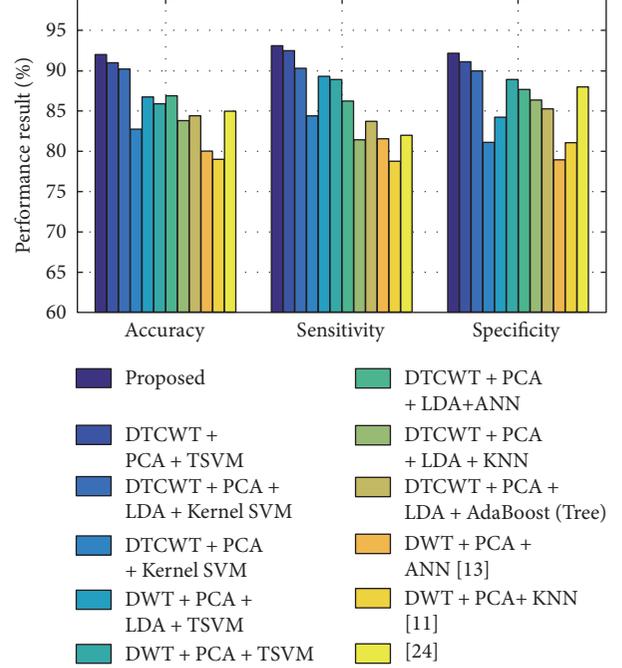


FIGURE 5: Bar chart of DTCWT-based classification performance of AD from HC over ADNI dataset.

$$\text{sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (14)$$

$$\text{specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}.$$

Sensitivity measures the proportion of correctly identified patients, and specificity measures the proportion of correctly identified controls. Additionally, some other frequently used statistical performance evaluation measures such as precision, recall, f -measure, and g mean are also calculated.

These measures are defined as

$$\begin{aligned} \text{Recall} &= \text{sensitivity}, \\ \text{Precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\ f_measure &= 2 * \left(\frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \right), \\ gmean &= \text{sqrt}(\text{TP rate} * \text{TN rate}). \end{aligned} \quad (15)$$

The previous measures are likely to provide an efficient overall performance assessment of a classifier.

3.3. Performance of Classification. In this study, the proposed hybrid method has been used for OASIS and ADNI data to distinguish control subjects from AD subjects. The recorded classification performance regarding accuracy (acc), sensitivity (sens), and specificity (spec) has been shown in a bar diagram in Figure 5 and in Figure 6. Performance varies depending on the principal components used for training and testing, as shown in Figure 7 for ADNI data. After testing with different PC values for both datasets, it was concluded that optimal classification performance was achieved with $PC=20$. To run a strict

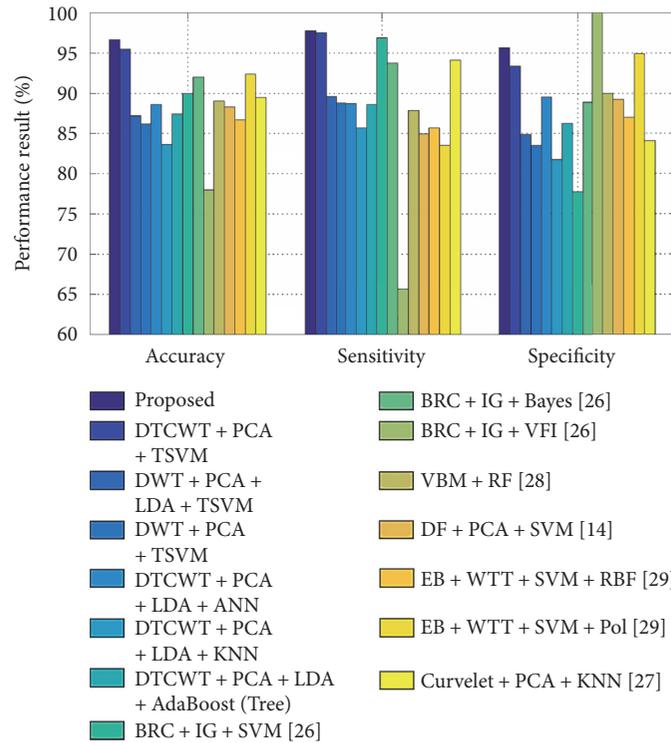


FIGURE 6: Bar chart of DTCWT-based classification performance of AD from HC over OASIS dataset.

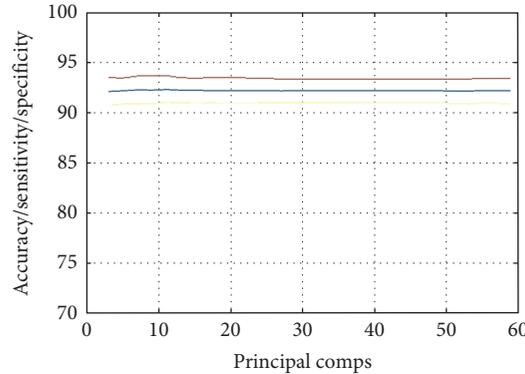


FIGURE 7: The number of principal components versus classification performance graph of proposed method.

statistical analysis, stratified cross validation (SCV) is applied. We have applied 5-fold CV to OASIS data and 10-fold CV to ADNI data, as the number of subjects in the OASIS dataset is lower than that of the ADNI dataset. 5-fold CV divides the dataset into five folds, whereas the 10-fold CV divides the dataset into ten folds.

The accuracies, sensitivities, specificities, and other statistical performance measures obtained with 10–20 runs of 10-fold SCV and 5-fold SCV are shown in Tables 5 and 6, respectively.

Although comparison with conventional methods can be difficult, we have compared our approach with some recent conventional disease detection algorithms using both datasets.

To analyze the performance over the ADNI dataset, the classification performance has been documented with both run-wise fold-wise classification, as shown in Tables 7 and 8. Table 8 shows the classification performance where linear discriminant analysis is not used. Individual columns and rows represent the classification accuracy of the corresponding runs and folds. Consequently, accuracy is calculated taking the average of all folds and runs. The classification performance in all 10 or 5 folds of each run can be analyzed with that.

We have compared several recently used sets of algorithms and methods [11, 13, 24], using the same datasets as in this article. We have obtained a $92.65 \pm 1.18\%$ accuracy, which outperforms the DWT-based method

TABLE 5: Performance evaluation over ADNI dataset.

Methods	Accuracy	Sensitivity	Specificity	Precision	Recall	f_measure	gmean
Proposed	92.65 ± 1.18	93.11 ± 1.29	92.19 ± 1.56	92.78 ± 1.27	93.11 ± 1.29	92.63 ± 1.19	92.46 ± 1.24
DTCWT+PCA+TSVM	91.77 ± 0.85	92.48 ± 0.89	91.13 ± 1.31	91.73 ± 0.95	92.48 ± 0.89	91.72 ± 0.77	91.57 ± 0.91

TABLE 6: Performance evaluation over OASIS dataset.

Methods	Accuracy	Sensitivity	Specificity	Precision	Recall	f_measure	gmean
Proposed	96.68 ± 1.44	97.72 ± 2.34	95.61 ± 1.67	96.13 ± 1.57	97.72 ± 2.34	96.76 ± 1.51	96.56 ± 1.44
DTCWT+PCA+TSVM	95.46 ± 1.35	97.55 ± 1.26	93.36 ± 2.39	94.14 ± 2.01	97.55 ± 1.26	95.61 ± 1.28	95.29 ± 1.42

TABLE 7: Run- and fold-wise classification performance of proposed approach over ADNI dataset.

Folds	Runs	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10	
Fold 1		94.44	100	100	100	94.44	88.8889	100	87.5	94.44	100	
Fold 2		100	94.117	100	88.23	82.35	94.11	81.25	94.11	94.44	88.88	
Fold 3		94.117	94.117	94.11	88.88	100	82.35	100	100	88.88	88.23	
Fold 4		94.117	88.235	88.88	94.11	100	93.75	94.117	94.11	100	82.35	
Fold 5		87.5	88.888	88.88	94.11	88.23	100	93.75	100	87.5	94.44	
Fold 6		100	94.117	87.5	88.88	100	76.47	88.23	77.77	94.11	94.44	<i>Average accuracy</i> 92.659
Fold 7		87.5	94.117	87.5	93.75	100	83.33	100	94.11	82.35	93.75	
Fold 8		87.5	100	100	88.88	100	94.44	100	83.33	87.5	94.11	
Fold 9		94.444	100	94.44	94.11	88.88	94.11	100	88.235	88.888	87.5	
Fold 10		94.444	83.333	83.33	94.11	82.35	100	88.88	94.117	100	100	
Fold-wise accuracy		93.406	93.692	92.46	92.512	93.62	90.747	94.624	91.3317	91.813	92.37	

TABLE 8: Run- and fold-wise classification performance of the DTCWT + PCA + TSVM method over ADNI dataset.

Folds	Runs	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10	
Fold 1		88.88	94.11	94.11	87.5	87.5	77.77	75	100	82.35	88.88	
Fold 2		94.11	100	100	94.44	100	94.44	88.88	94.11	88.23	87.5	
Fold 3		94.11	87.5	88.23	93.75	94.11	94.11	88.88	88.23	93.75	76.47	
Fold 4		93.75	82.35	88.23	88.23	100	94.11	94.11	88.23	76.47	100	
Fold 5		88.88	94.11	94.11	83.33	82.35	94.11	82.35	88.88	94.44	100	
Fold 6		94.11	82.35	94.44	100	100	100	87.5	94.11	88.88	88.88	<i>Average accuracy</i> 91.77
Fold 7		83.33	94.44	100	100	83.33	87.5	100	88.23	100	100	
Fold 8		87.5	94.44	83.33	82.35	88.23	93.75	94.44	88.23	93.75	83.33	
Fold 9		94.44	100	94.44	88.88	100	88.23	100	82.35	88.88	100	
Fold 10		94.11	94.11	88.23	88.23	94.11	100	100	100	100	100	
Fold-wise accuracy		91.32	92.34	92.51	90.67	92.96	92.40	91.11	91.24	90.67	92.50	

proposed by El-Dahshan et al. [11] and Zhang et al., [13] as shown in Table 9 and Figure 5. The proposed method was also executed applying conventional DWT principal coefficients. We have seen that the DTCWT-based method outperforms DWT-based method. In addition, performance is documented without using LDA for both types of feature.

However, classification performance has become more efficient when LDA-projected features are considered, as shown in Tables 5 and 9 and Figure 5. Our method has been distinguished from the volumetric feature-based research study proposed by Schmitter et al. [24], and it outperforms the results thereof, as shown in Figure 5. Additionally, our results

TABLE 9: Classification performance of AD from HC over ADNI data.

Methods	Accuracy	Sensitivity	Specificity
Proposed	92.65 ± 1.18	93.11 ± 1.29	92.19 ± 1.56
DTCWT + PCA + TSVM	91.77 ± 0.85	92.48 ± 0.89	91.13 ± 1.31
DTCWT + PCA + LDA + Kernel SVM	90.181 ± 0.97	90.276 ± 1.60	90.101 ± 1.23
DTCWT + PCA + Kernel SVM	82.74 ± 1.24	84.43 ± 1.51	81.18 ± 1.85
DWT + PCA + LDA + TSVM	86.75 ± 1.69	89.32 ± 1.43	84.23 ± 2.21
DWT + PCA + TSVM	85.88 ± 1.16	88.93 ± 1.61	88.93 ± 2.02
DTCWT + PCA + LDA + ANN	86.97 ± 1.30	86.25 ± 1.78	87.72 ± 3.51
DTCWT + PCA + LDA + KNN	83.89 ± 0.75	81.41 ± 1.33	86.34 ± 1.08
DTCWT + PCA + LDA + AdaBoost (tree)	84.48	83.72	85.26
DWT + PCA + ANN [13]	80.05 ± 0.72	81.538 ± 1.41	78.974 ± 1.09
DWT + PCA + KNN [11]	79.964 ± 1.19	78.771 ± 2.37	81.08 ± 1.67
[24]	85	82	88

TABLE 10: Run- and fold-wise classification performance of the proposed approach over OASIS dataset.

Folds	Runs	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10	
Fold 1		94.44	94.44	100	88.23	100	100	100	94.11	100	100	
Fold 2		94.11	100	88.23	88.88	88.88	100	94.11	100	88.88	93.75	
Fold 3		94.44	94.11	100	94.11	100	94.44	100	94.11	100	100	
Fold 4		100	100	100	94.44	100	94.44	100	100	100	100	<i>Average accuracy</i> 96.58
Fold 5		100	88.88	100	100	94.44	100	94.11	100	94.11	94.44	
Fold-wise accuracy		96.60	95.49	97.64	93.13	96.66	97.77	97.64	97.64	96.60	97.63	

TABLE 11: Run- and fold-wise classification performance of the DTCWT + PCA + TSVM method over OASIS dataset.

Folds	Runs	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10	
Fold 1		100	94.11	94.44	100	94.44	94.44	94.11	94.44	88.8	94.44	
Fold 2		100	100	88.23	94.44	94.44	100	94.11	94.44	100	94.44	
Fold 3		94.44	83.33	94.44	94.11	100	94.44	94.44	94.11	94.44	100	
Fold 4		94.44	100	94.44	94.11	94.44	88.88	100	100	100	88.88	<i>Average accuracy</i> 95.46
Fold 5		94.11	100	94.11	94.44	100	87.5	100	94.44	100	100	
Fold-wise accuracy		96.60	95.49	93.13	95.42	96.66	93.05	96.53	95.49	96.66	95.55	

were compared with kernel SVM-based classification and produced superior performance.

Likewise, to analyze and stratify OASIS dataset, identical methods have been used, namely run-wise and fold-wise classifications, as depicted in Tables 10 and 11.

We observed, as shown in Tables 6 and 12 and Figure 6, that our method yielded an accuracy of 96.68 ± 1.44 , a sensitivity of 97.72 ± 2.34 , and a specificity of 95.61 ± 1.67 . This classification performance has also been documented without using LDA; however, results improve when LDA is applied on principal dual-tree complex wavelet transform coefficients or principal DWT coefficients and TSVM is used as a classifier. The result is efficient when DTCWT principal coefficients are used over DWT method.

To further verify the efficacy of the proposed method, we compared it with 12 state-of-the-art approaches, as shown in Table 12, which utilized different statistical settings.

The results show that US + SVD-PCA + SVM-DT [25] yielded an accuracy of 90%, a sensitivity of 94%, and a specificity of 71%; BRC + IG + SVM [26] achieved an accuracy of 90.00%, a sensitivity of 96.88%, and a specificity of 77.78%; and curvelet + PCA + KNN [27] obtained stratification an accuracy of 89.47%, a sensitivity of 94.12%, and a specificity of 84.09%. We observed that these methods have lower specificity compared to the other methods mentioned previously. In contrast, BRC + IG + Bayes [26] yielded higher specificity.

Similarly, BRC + IG + VFI [26] yielded a classification accuracy of 78%, sensitivity of 65.63%, and specificity of

TABLE 12: Algorithm performance comparison over OASIS MRI data.

Algorithm	Accuracy	Sensitivity	Specificity	Precision
Proposed	96.68 ± 1.44	97.72 ± 2.34	95.61 ± 1.67	96.13 ± 1.57
DTCWT + PCA + TSVM	95.46 ± 1.35	97.55 ± 1.26	93.36 ± 2.39	94.15 ± 2.01
DWT + PCA + LDA + TSVM	87.23 ± 1.65	89.61 ± 2.25	84.85 ± 1.66	86.66 ± 1.99
DWT + PCA + TSVM	86.19 ± 1.50	88.83 ± 1.98	83.5 ± 1.87	85.66 ± 1.84
DTCWT + PCA + LDA + ANN	88.59 ± 2.08	88.75 ± 2.75	89.55 ± 3.96	NA
DTCWT + PCA + LDA + KNN	83.69 ± 1.57	85.7 ± 1.94	81.8 ± 1.45	NA
DTCWT + PCA + LDA + AdaBoost (tree)	87.45	88.59	86.26	NA
BRC + IG + SVM [26]	90.00 (77.41, 96.26)	96.88 (82.01, 99.84)	77.78 (51.92, 92.63)	NA
BRC + IG + Bayes [26]	92.00 (79.89, 97.41)	93.75 (77.78, 98.27)	88.89 (63.93, 98.05)	NA
BRC + IG + VFI [26]	78.00 (63.67, 88.01)	65.63 (46.78, 80.83)	100.00 (78.12, 100)	NA
MGM + PEC + SVM [30]	92.07 ± 1.12	86.67 ± 4.71	N/A	95.83 ± 5.89
GEODAN + BD + SVM [30]	92.09 ± 2.60	80.00 ± 4.00	NA	88.09 ± 5.33
TJM + WTT + SVM [30]	92.83 ± 0.91	86.33 ± 3.73	N/A	85.62 ± 0.85
VBM + RF [28]	89.0 ± 0.7	87.9 ± 1.2	90.0 ± 1.1	NA
DF + PCA + SVM [14]	88.27 ± 1.9	84.93 ± 1.21	89.21 ± 1.6	69.30 ± 1.91
EB + WTT + SVM + RBF [29]	86.71 ± 1.93	85.71 ± 1.91	86.99 ± 2.30	66.12 ± 4.16
EB + WTT + SVM + Pol [29]	92.36 ± 0.94	83.48 ± 3.27	94.90 ± 1.09	82.28 ± 2.78
Curvelet + PCA + KNN [27]	89.47	94.12	84.09	NA
US + SVDPCA + SVM-DT [25]	90	94	71	NA

100%. Although it yielded high specificity, accuracy and sensitivity yielded by this algorithm were comparatively poor.

All other methods achieved satisfying results. VBM + RF [28] obtained an accuracy of $89.0 \pm 0.7\%$, a sensitivity of $87.9 \pm 1.2\%$, and a specificity of 90.0 ± 1.1 . These promising results were achieved largely due to voxel-based morphometry (VBM).

DF + PCA + SVM [14] yielded an accuracy of $88.27 \pm 1.89\%$, a sensitivity of $84.93 \pm 1.21\%$, and a specificity of $89.21 \pm 1.63\%$. This method is based on a novel approach called displacement field (DF).

EB + WTT + SVM + RBF [29] obtained an accuracy of $86.71 \pm 1.93\%$, a sensitivity of $85.71 \pm 1.91\%$, and a specificity of $86.99 \pm 2.30\%$; however, EB + WTT + SVM + Pol [29] yields better classification performance.

In addition, MGM + PEC + SVM [30], GEODAN + BD + SVM [30], and TJM + WTT + SVM [30] achieved approximately 92% accuracy with similarly high sensitivity and precision; specificity was not calculated for these methods.

Finally, taking classification performance into consideration, our approach outperforms all other methods analyzed here. We have also produced promising performance metrics for sensitivity and specificity. Hence, we submit that our results are either superior or comparable to the other compared methods.

4. Conclusions

Our proposed experiment uses LDA on the principal components of DTCWT coefficients and TSVM to stratify AD. Our proposed detection method for the ADNI dataset yielded an accuracy of $92.65 \pm 1.18\%$ with high sensitivity and

specificity. Our proposed method also outperforms those of Zhang et al. [13] and El-Dahshan et al. [11] and the volumetric feature-based classification proposed by Schmitter et al. [24]. In addition, the classification performance of our proposed experiment for OASIS data performs better when compared with the several state-of-the-art approaches specified in this paper—yielding an accuracy of 96.68 ± 1.44 with similarly high sensitivity and specificity.

In the future, we will carry forward our research focusing on the following: (i) 3D DTCWT-based feature extraction with multiresolution analysis and classification and (ii) convolutional neural network- (CNN-) based classification using 3D MRI.

Additional Points

Data Access. Data used in reparation of this article were obtained from the Alzheimer’s Disease Neuroimaging Initiative (ADNI) database (<http://adni.loni.usc.edu>). A complete listing of ADNI investigators can be found at https://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf.

Disclosure

The investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in the analysis or writing of this report.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the Brain Research Program through the National Research Foundation of Korea funded by the Ministry of Science, ICT & Future Planning (NRF-2014M3C7A1046050). And this study was supported by the research funds from Chosun University, 2017. Data collection and sharing for this project was funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant no. U01 AG024904) and DOD ADNI (Department of Defense Award no. W81XWH-12-2-0012). ADNI is funded by the National Institute on Aging and the National Institute of Biomedical Imaging and Bioengineering and through generous contributions from the following: AbbVie, Alzheimer's Association; Alzheimer's Drug Discovery Foundation; Araclon Biotech; BioClinica Inc.; Biogen; Bristol-Myers Squibb Company; CereSpir Inc.; Cogstate; Eisai Inc.; Elan Pharmaceuticals Inc.; Eli Lilly and Company; EuroImmun; F. Hoffmann-La Roche Ltd. and its affiliated company Genentech Inc.; Fujirebio; GE Healthcare; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development LLC; Johnson & Johnson Pharmaceutical Research & Development LLC; Lumosity; Lundbeck; Merck & Co. Inc.; Meso Scale Diagnostics LLC; NeuroRx Research; Neurotrack Technologies; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Takeda Pharmaceutical Company; and Transition Therapeutics. The Canadian Institutes of Health Research provide funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (<http://www.fnih.org/>). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Therapeutic Research Institute at the University of Southern California.

References

- [1] J. Ashburner and S. Klöppel, "Multivariate models of inter-subject anatomical variability," *NeuroImage*, vol. 56, no. 2, pp. 422–439, 2011.
- [2] N. Koutsouleris, E. M. Meisenzahl, C. Davatzikos et al., "Use of neuroanatomical pattern classification to identify subjects in at-risk mental states of psychosis and predict disease transition," *Archives of General Psychiatry*, vol. 66, no. 7, pp. 700–712, 2009.
- [3] M. Weygandt, K. Hackmack, C. P. Füller et al., "MRI pattern recognition in multiple sclerosis normal-appearing brain areas," *PloS One*, vol. 6, no. 6, article e21138, 2011.
- [4] M. Chupin, E. Gerardin, R. Cuingnet et al., "Fully automatic hippocampus segmentation and classification in Alzheimer's disease and mild cognitive impairment applied on data from ADNI," *Hippocampus*, vol. 19, no. 6, pp. 579–587, 2009.
- [5] S. Klöppel, C. M. Stonnington, C. Chu et al., "Automatic classification of MR scans in Alzheimer's disease," *Brain*, vol. 131, no. 3, pp. 681–689, 2008.
- [6] C. Ecker, A. Marquand, J. Mourão-Miranda et al., "Describing the brain in autism in five dimensions-magnetic resonance imaging assisted diagnosis of autism spectrum disorder using a multiparameter classification approach," *The Journal of Neuroscience*, vol. 30, no. 32, pp. 10612–10623, 2010.
- [7] R. A. Yotter, I. Nenadic, G. Ziegler, P. M. Thompson, and C. Gaser, "Local cortical surface complexity maps from spherical harmonic reconstructions," *NeuroImage*, vol. 56, no. 3, pp. 961–973, 2011.
- [8] Z. Sankari and H. Adeli, "Probabilistic neural networks for EEG-based diagnosis of Alzheimer's disease using conventional and wavelet coherence," *Journal of Neuroscience Methods*, vol. 197, no. 1, pp. 165–170, 2011.
- [9] S. Chaplot, L. M. Patnaik, and N. R. Jagannathan, "Classification of magnetic resonance brain images using wavelets as input to support vector machine and neural network," *Biomedical Signal Processing and Control*, vol. 1, no. 1, pp. 86–92, 2006.
- [10] C. H. Moritz, V. M. Houghton, D. Cordes, M. Quigley, and M. E. Meyerand, "Whole-brain functional MR imaging activation from finger tapping task examined with independent component analysis," *American Journal of Neuroradiology*, vol. 21, no. 9, pp. 1629–1635, 2000.
- [11] T. H. El-Dahshan and A. B. M. Salem, "Hybrid intelligent techniques for MRI brain images classification," *Digital Signal Processing*, vol. 20, no. 2, pp. 433–441, 2010.
- [12] R. N. Bracewell, *The Fourier Transform and Its Applications*, McGraw-Hill, New York, Third edition, 1999.
- [13] Y. Zhang, Z. Dong, L. Wu, and S. Wang, "A hybrid method for MRI brain image classification," *Expert Systems with Applications*, vol. 38, no. 8, pp. 10049–10053, 2011.
- [14] Y. Zhang and S. Wang, "Detection of Alzheimer's disease by displacement field and machine learning," *PeerJ*, vol. 3, article e1251, 2015.
- [15] D. Tomar and S. Agarwal, "Twin support vector machine: a review from 2007 to 2014," *Egyptian Informatics Journal*, vol. 16, no. 1, pp. 55–69, 2015.
- [16] A. Barri, A. Doods, and P. Schelkens, "The near shift-invariance of the dual-tree complex wavelet transform revisited," *Journal of Mathematical Analysis and Applications*, vol. 389, no. 2, pp. 1303–1314, 2012.
- [17] H. Andersen, D. M. Gash, and M. J. Avison, "Principal component analysis of the dynamic response measured by fMRI: a generalized linear systems framework," *Journal of Magnetic Resonance Imaging*, vol. 17, no. 6, pp. 795–815, 1999.
- [18] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.
- [19] K. R. Jayadeva and S. Chandra, "Twin support vector machines for pattern classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp. 905–910, 2007.
- [20] S. Alam and G. R. Kwon, "Classification of Alzheimer disease using dual tree complex wavelet transform and minimum redundancy & maximum relevant feature," *The Journal of Korean Institute of Next Generation Computing*, vol. 12, no. 3, pp. 51–59, 2016.
- [21] J. A. Nasiri, N. M. Charkari, and K. Mozafari, "Energy-based model of least squares twin support vector machines for human action recognition," *Signal Processing*, vol. 104, pp. 248–257, 2014.
- [22] Y. H. Shao, W. J. Chen, J. J. Zhang, Z. Wang, and N. Y. Deng, "An efficient weighted Lagrangian twin support vector machine for imbalanced data classification," *Pattern Recognition*, vol. 47, no. 9, pp. 3158–3167, 2014.

- [23] Z. Xu, Z. Qi, and J. Zhang, "Learning with positive and unlabeled examples using biased twin support vector machine," *Neural Computing & Applications*, vol. 25, no. 6, pp. 1303–1311, 2014.
- [24] D. Schmitter, A. Roche, B. Maréchal et al., "An evaluation of volume-based morphometry for prediction of mild cognitive impairment and Alzheimer's disease," *NeuroImage: Clinical*, vol. 7, pp. 7–17, 2015.
- [25] Y. Zhang, S. Wang, and Z. Dong, "Classification of Alzheimer disease based on structural magnetic resonance imaging by kernel support vector machine decision tree," *Progress in Electromagnetics Research*, vol. 144, pp. 171–184, 2014.
- [26] C. Plant, S. J. Teipel, A. Oswald et al., "Automated detection of brain atrophy patterns based on MRI for the prediction of Alzheimer's disease," *NeuroImage*, vol. 50, no. 1, pp. 162–174, 2010.
- [27] D. Jha and G. R. Kwon, "Alzheimer disease detection in MRI using curvelet transform with KNN," *Journal of Korean Institute of Information Technology*, vol. 14, no. 8, pp. 121–129, 2016.
- [28] K. R. Gray, P. Alijabar, R. A. Heckemann, A. Hammers, and D. Rueckert, "Random forest-based similarity measures for multi-modal classification of Alzheimer's disease," *NeuroImage*, vol. 65, pp. 167–175, 2013.
- [29] Y. Zhang, Z. Dong, P. Phillips et al., "Detection of subjects and brain regions related to Alzheimer's disease using 3D MRI scans based on eigenbrain and machine learning," *Frontiers in Computational Neuroscience*, vol. 9, p. 66, 2015.
- [30] A. Savio and M. Grana, "Deformation based feature selection for computer aided diagnosis of Alzheimer's disease," *Expert Systems with Applications*, vol. 40, no. 5, pp. 1619–1628, 2013.

Research Article

R Peak Detection Method Using Wavelet Transform and Modified Shannon Energy Envelope

Jeong-Seon Park,¹ Sang-Woong Lee,² and Unsang Park³

¹Department of Multimedia, Chonnam National University, 50 Daehak-ro, Yeosu, Jeollanamdo 59626, Republic of Korea

²Department of Software, Gachon University, 1342 Seongnam-daero, Sujeong-gu, Seongnam, Gyeonggido 13120, Republic of Korea

³Department of Computer Science & Engineering, Sogang University, 35 Baekbeom-ro, Mapo-gu, Seoul 04107, Republic of Korea

Correspondence should be addressed to Unsang Park; unsangpark@sogang.ac.kr

Received 16 December 2016; Revised 3 April 2017; Accepted 23 April 2017; Published 5 July 2017

Academic Editor: Benlian Xu

Copyright © 2017 Jeong-Seon Park et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Rapid automatic detection of the fiducial points—namely, the P wave, QRS complex, and T wave—is necessary for early detection of cardiovascular diseases (CVDs). In this paper, we present an R peak detection method using the wavelet transform (WT) and a modified Shannon energy envelope (SEE) for rapid ECG analysis. The proposed WTSEE algorithm performs a wavelet transform to reduce the size and noise of ECG signals and creates SEE after first-order differentiation and amplitude normalization. Subsequently, the peak energy envelope (PEE) is extracted from the SEE. Then, R peaks are estimated from the PEE, and the estimated peaks are adjusted from the input ECG. Finally, the algorithm generates the final R features by validating R-R intervals and updating the extracted R peaks. The proposed R peak detection method was validated using 48 first-channel ECG records of the MIT-BIH arrhythmia database with a sensitivity of 99.93%, positive predictability of 99.91%, detection error rate of 0.16%, and accuracy of 99.84%. Considering the high detection accuracy and fast processing speed due to the wavelet transform applied before calculating SEE, the proposed method is highly effective for real-time applications in early detection of CVDs.

1. Introduction

An electrocardiogram (ECG) is a recording of the electrical activity of the heart [1, 2] and a graphical representation of the signals obtained from electrodes placed on the skin near the heart [1, 3, 4]. The recent use of computers in conducting ECG analysis allows the patterns of the ECG signal, composed of multiple cycles that include numerous sample points, to be visualized [5]. Some of these sample points are fiducial—namely, the P wave, QRS complex, and T wave [4]. The identification of these points is a critical step in analyzing the ECG signal and has become possible by analyzing its morphological patterns [6].

In 2012, cardiovascular diseases (CVDs) accounted for 37% of premature (under the age of 70) noncommunicable disease mortality [7–9]. The detection of fiducial points is

critical for the initial diagnosis and analysis of CVDs. In Figure 1, we can see the peaks of the QRS complex, the highest of which is known as the R peak in the QRS interval [10]. Other time segments are also shown, such as the P-Q and S-T segments. The R peak in the QRS interval is the most important feature for analyzing the ECG data. All these waves are electrical manifestations of the contractile activity of the heart [11]. Detection of the main characteristic waves in an ECG is one of the most essential tasks, and the performance of any CVD analysis method depends on the reliable detection of these waves. R peak detection in ECG is one such method that is widely used to diagnose heart rhythm irregularities and estimate heart-rate variability (HRV) [12, 13].

Significant research efforts have been devoted to the detection of the fiducial points of an ECG signal. Those

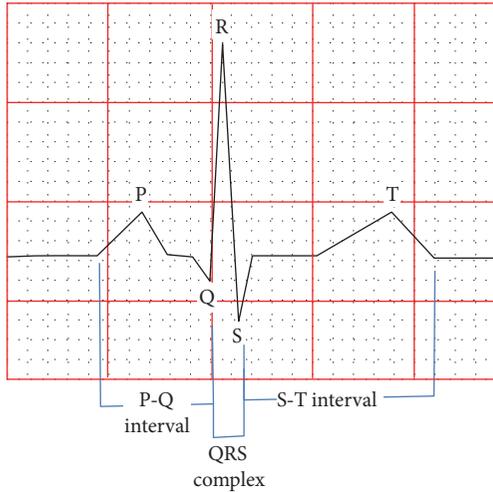


FIGURE 1: Fiducial points of an ECG signal: P wave, QRS complex, T wave, and time intervals.

methods include slope-based threshold methods [14, 15], wavelet-transform-based methods [1, 4, 10, 16–18], mathematical-morphology-based methods [6, 19], digital filtering methods [14, 20, 21], and Shannon energy envelope- (SEE-) based methods [22–25]. There are also other studies that use the wavelet transform for denoising ECG signals [4, 17].

The Pan and Tompkins methods (PT) [14] appear to be the most common benchmark given that they incorporate several fundamental techniques including low-pass filtering, high-pass filtering, derivative filtering, squaring, and windowing for the detection of the R peaks. The principal drawback of a filtering-based approach is the adverse effect on performance [20] because of the change in frequency of the characteristic wave. Shannon energy with the Hilbert transform method (SEHT) [22] provides good accuracy for detecting R peaks. However, the Hilbert transform in SEHT requires large memory and processing time, making it unsuitable for real-time application. Moreover, SEHT detects many noise peaks in ECG data with long pauses. Zhu and Dong [23] developed an R peak detection method called PSEE by using only the SEE. The authors used an amplitude threshold that affects the performance of the algorithm for valid peak detection. A QRS complex generally overlaps in the frequency domain [26], resulting in false positive detections. Some of the threshold techniques are highly noise-sensitive [17]; therefore, developments of sophisticated, automatic, and computationally efficient techniques are required that can outperform existing methods to ensure the real-time analysis of an ECG for the proper diagnosis of CVDs.

In this paper, we propose a novel approach of using the wavelet transform (WT) and modified SEE for the rapid detection of R peaks in the QRS complex. First, the proposed WTSEE algorithm performs a WT to reduce the size and noise of ECG signals and subsequently calculates the SEE after first-order differentiation and amplitude normalization. Following this step, the peak energy envelope (PEE) is made from the SEE for easy identification of peaks. R peaks are

then estimated from the PEE, and the estimated peaks are adjusted from the input ECG. Finally, the algorithm generates the final R peaks by validating R-R intervals and updating the extracted R peaks.

The proposed R peak detection method was validated using 48 first-channel ECG records of the MIT-BIH arrhythmia database with sensitivity, positive predictability, detection error rate, and accuracy. We also measured the mean of R-R interval (MRR), standard deviation of normal to normal R-R intervals (SDNN), and root mean square of successive heartbeat interval differences (RMSSDs), which can be used for analyzing heart rate variability.

The remainder of this paper is organized as follows. In Section 2, we discuss the proposed R peak detection method using the wavelet transform and SEE. In Section 3, we provide the experimental results, the performance analysis of the proposed method, and a discussion of the real-time implementation. In Section 4, we conclude our work and provide insight into future study.

2. Methods

An ECG signal consists of many cycles of P, Q, R, and S waves, with each cycle comprising many sample points. In MIT-BIH record 100, a cycle comprises approximately 280 sample points [27]. There are up to 10 fiducial points that determine the overall characteristics of a cycle of ECG signal. Among various existing SEE-based methods, band-pass filters, such as the Chebyshev type I filter and Butterworth filter, are used for denoising input ECG signals as preprocessing steps. In the proposed method, wavelet transform replaces the band-pass filters by level 2 down-sampling, soft thresholding for denoising detailed coefficients [17], and reconstructing the level 1 signal. By applying this procedure, we can reduce the size and time required for extracting R peaks.

The proposed WTSEE algorithm extracts these fiducial points (R peaks) from the down-sampled ECG signals by calculating SEE, repeating a similar procedure as SEE to emphasize peak information (thus, we call this procedure the peak energy envelope), detecting R peaks, and updating the R peaks by comparing the R-R intervals. Figure 2 presents the data flow in the proposed R peak detection algorithm with its principle stages. First, the proposed WTSEE algorithm performs a wavelet transform, instead of a band-pass filter, to reduce the size and noise of ECG signals. It then calculates SEE after first-order differentiation and amplitude normalization. In the third stage, the PEE is created from the SEE for easy identification of peaks. In the next stage, R peaks are estimated from the PEE, and the estimated peaks are adjusted from the input ECG. Finally, the algorithm generates the final R peaks by validating R-R intervals and updating the extracted R peaks. In Figure 2, original time space and down-sampled time space are represented by t_1 and t_2 , respectively.

2.1. Discrete Wavelet Transform. The wavelet transform (WT) is a good technique for signal compression and noise reduction. The computational complexity for the discrete wavelet transform (DWT) is $O(n)$. Wavelet

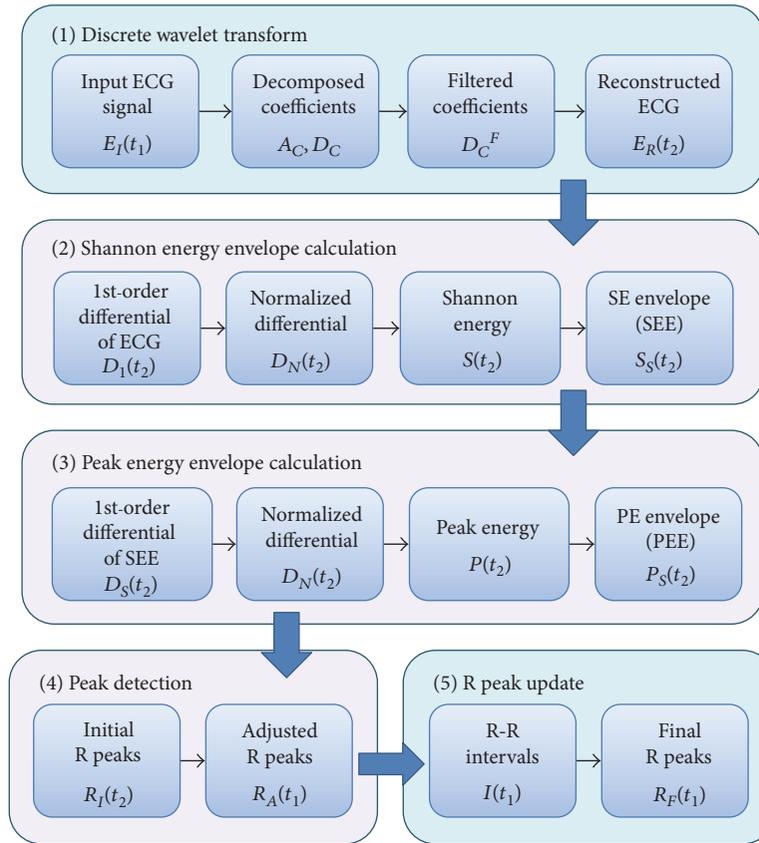


FIGURE 2: Data flow in the proposed R peak detection method using WT and modified SEE.

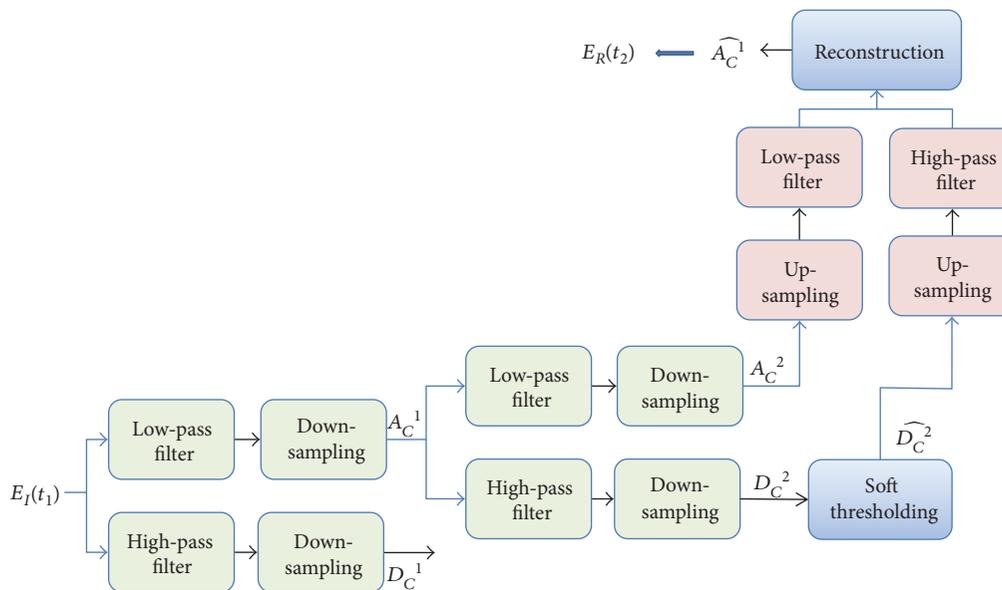


FIGURE 3: Procedure of wavelet transform comprising filtering, down-sampling, thresholding, up-sampling, and reconstruction.

analysis combines filtering and down-sampling as shown in Figure 3 [4, 17].

(1) As the first step of the proposed method, WT reduces the size and noise of an ECG. By applying these

transform, memory requirements and processing time are dramatically reduced.

Symlets wavelet (sym5) is chosen as the wavelet function to decompose the ECG signals, and the thresholding method

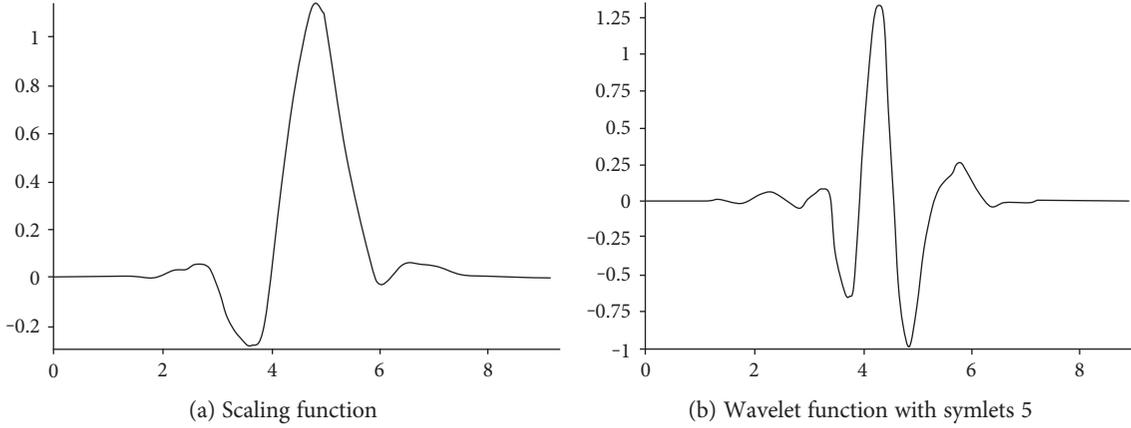


FIGURE 4: Comparison of scaling function and wavelet function with symlets 5 [28].

is employed to remove the noise [4]. Figure 4 demonstrates the difference between the scaling function and wavelet function with symlets [28]. By applying this procedure, the peak signal becomes clearer. The sym5 wavelet transformation is performed as

$$[A_C, D_C] = \text{dwt}(E_I, 'sym5'), \quad (1)$$

where A_C is an approximation coefficient vector, D_C is a detail coefficient vector, E_I is an input ECG signal vector, and $\text{dwt}(\cdot)$ is a DWT function.

(2) To reduce the noise of an ECG signal, we applied a soft thresholding that is recognized as more powerful than hard thresholding as

$$\widehat{D_C(j)} = \begin{cases} \text{sign}(D_C(j))(|D_C(j)| - t), & |D_C(j)| > t \\ 0, & |D_C(j)| \leq t, \end{cases} \quad (2)$$

where coefficients $\widehat{D_C(j)}$ and $D_C(j)$ are detail coefficients after and before thresholding, respectively.

In the proposed scheme, we chose the universal threshold selection method. Here, the value of threshold (t) is computed as

$$t = \sigma \sqrt{\frac{2 \log(N)}{N}}, \quad (3)$$

where N is the total number of wavelet coefficients and $\sigma = \text{median}(|D_C(j)|)/0.6745$ is the standard deviation of the noise.

After the noise removal, reconstruction is applied to obtain noise-free ECG signal as follows:

$$E^F = \text{idwt}(A_C, \widehat{D_C}, 'sym5'), \quad (4)$$

where A_C is a previously extracted approximation coefficients vector, $\widehat{D_C}$ is a noise-removed detail coefficient vector, E^F is a denoised ECG signal, and $\text{idwt}(\cdot)$ is an inverse DWT function.

2.2. Shannon Energy Envelope Calculation. (3) After applying the 2nd DWT along with noise removal and reconstruction,

we perform the first-order differentiation of the signal to obtain the slope information. The first-order differentiation is equivalent with a high-pass filter that passes high-frequency components (QRS complex) and attenuates lower frequency components (P and T waves). The mathematical implementation of the first-order differentiation can be shown as

$$D_n = E_{n+1}^F - E_n^F. \quad (5)$$

(4) Next, the signal is normalized to scale its value to 1. This is to prepare the signal for SEE computation.

$$\widehat{D_n} = \frac{D_n}{\max(|D_n|)}. \quad (6)$$

(5) After differentiating the ECG signal, it becomes a bipolar signal. Given that the method is based on peak detection, we must transform the differentiated signal into a unipolar signal. The unipolar signal can be obtained by the following Shannon energy function:

$$S_n^E = -(\widehat{D_n})^2 \log((D_n)^2). \quad (7)$$

(6) To obtain a smooth SEE, the signal is passed through a zero-shift moving average filter [13] that can be considered as a moving window integrator. The mathematical expression of a moving average filter is expressed in the following where the window width is set as 33:

$$S_n^S = \frac{1}{N} \{S_{n-N/2}^E + \dots + S_{n-1}^E + S_n^E + S_{n+1}^E + \dots + S_{n+N/2}^E\}. \quad (8)$$

The length of the moving average filter is an important parameter for this detection method. Generally, the length of the moving average filter is taken as approximately the width of the QRS complex. In the existing method [13], 65 samples are used for the moving average filter. However, in this paper, we reduce the length to 33.

2.3. Peak Energy Envelope Calculation. After the SEE calculation stage, we obtain the smooth peak signal, but it contains both true R peaks and false R peaks. In this stage, we thus attenuate the false R peaks and emphasize the true R peaks.

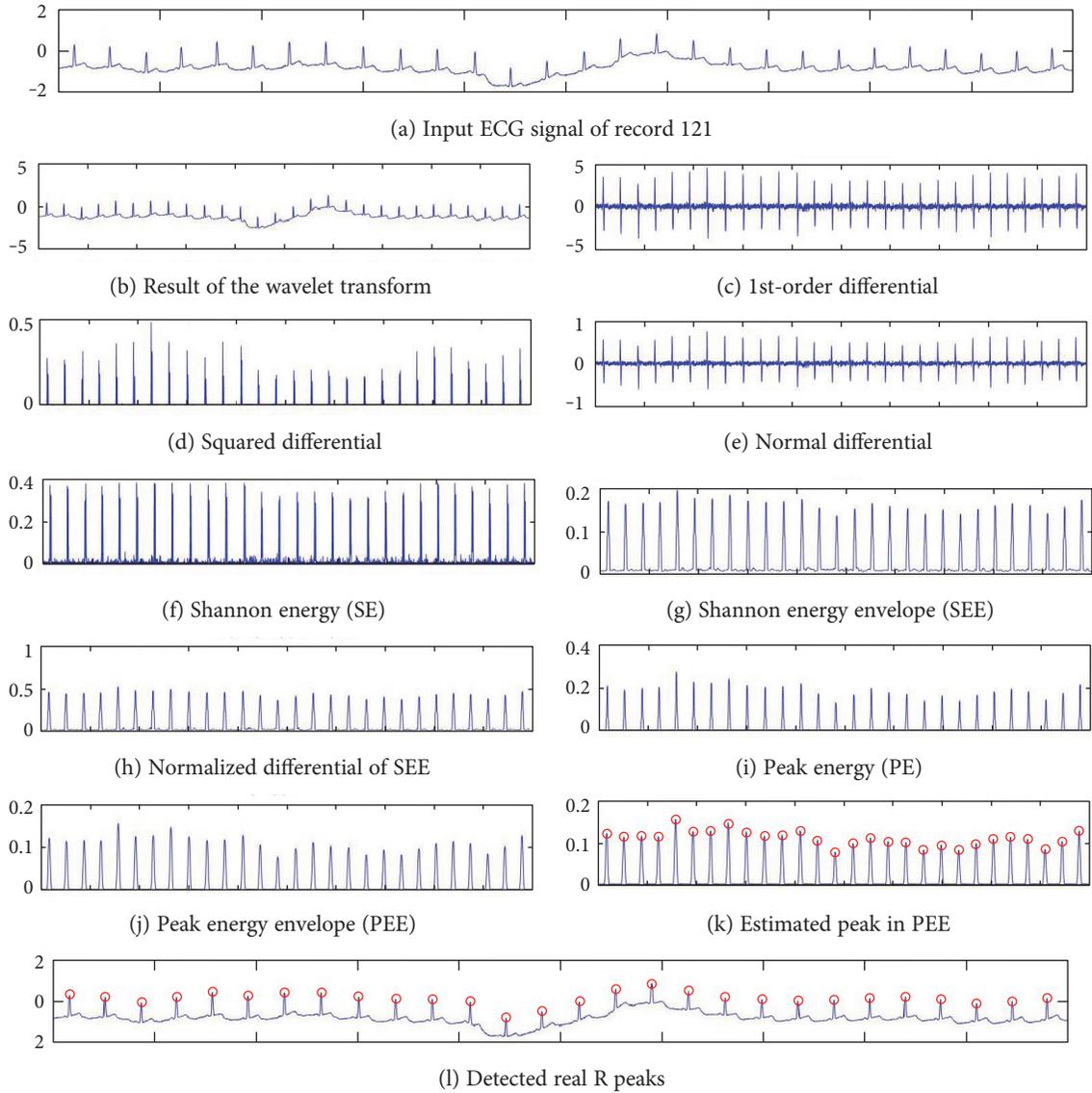


FIGURE 5: Example flow of the proposed R peak detection methods using WT and modified SEE. (a) Input ECG signal of record 121, (b) resized ECG after wavelet transform, (c) output of 1st-order differential, (d) squared differential, (e) normalized differential, (f) calculated Shannon energy (SE), (g) extracted Shannon energy envelope (SEE) after moving average, (h) normalized difference of SEE, (i) calculated peak energy (PE) after square operation, (j) extracted peak energy envelope (PEE) after moving average, (k) estimated peaks in PEE (red circles), and (l) true detected R peaks in ECG signal (red circles).

This is performed by similar steps to those of the SEE, so we call this procedure the peak energy envelope (PEE).

(7) It is natural that the amplitude values for true R peaks are higher than those for false peaks. Thus, if we take the first-order differentiation of the signal, it stores the slope information of the true peaks but reduces the slope information of the false peaks. The first-order differentiation can be expressed as

$$D_n^S = S_{n+1}^S - S_n^S. \quad (9)$$

(8) Next, the signal is amplitude normalized to unity as

$$\widehat{D}_n^S = \frac{D_n^S}{\max(|D_n^S|)}. \quad (10)$$

(9) The normalized bipolar signal is converted to a unipolar signal by a squaring operation. The amplitude of false peaks is very low, and the squaring operation will attenuate these peaks completely. As a result, the true R peaks are amplified, and false R peaks are diminished. The squared unipolar signal can be expressed as:

$$P_n^E = \left(\widehat{D}_n^S\right)^2. \quad (11)$$

(10) The signal is then passed through a moving average filter to obtain a resulting signal with smooth peak. In our proposed technique, we use a moving average filter length of 43 samples, which is smaller than the existing length of

85 [13]. The final signal expressed next is used for peak detection where the window width is set as 43.

$$P_n^S = \frac{1}{N} \{P_{n-N/2}^E + \dots + P_{n-1}^E + P_n^E + P_{n+1}^E + \dots + P_{n+N/2}^E\}. \quad (12)$$

2.4. Peak Detection. Figure 5 shows an example flow of the proposed WTSEE method for detecting R peaks. After squaring, to obtain a smooth SEE, a moving average filter operation is performed. The output of the moving average filter does not provide any unnecessary rising peaks.

(11) The locations of rising peaks are referred to as the locations of true R peaks, as shown in the Figure 5(k). Thus, no amplitude threshold value is required for the detection of R peaks.

$$R^E = \text{findPeaks}(P^S), \quad (13)$$

where R^E is the estimated locations of rising peaks and $\text{findPeaks}(\cdot)$ is a peak-finding function from smooth PEE P^S .

(12) In this stage, the R peaks are detected by applying a peak-finding algorithm. However, the detected peak locations are slightly different from the actual positions of the R peaks in the ECG signal. Thus, to find the real positions of R peaks, the actual sample instant of R peaks in the input ECG signal is found by searching for the maximum amplitude within ± 25 samples of the identified location in the previous step as

$$R_k^C = \arg \max_k \{E_{k-25}^F, \dots, E_k^F, \dots, E_{k+25}^F\}, \quad (14)$$

where R_k^C is the searched real position from the input ECG signal and E_k^F is the amplitude of the k th positions of estimated R peak R_k^E .

In Figure 5, the total signal processing of the proposed method is shown, where red circles indicate the detected R peaks by using the proposed method in Figure 5(l).

2.5. R Peak Update. As the final procedure, the previously detected peak set R^C is validated and updated using the following steps. This validation and update process is based on the R-R intervals between neighboring R peaks. The objective of this procedure is balancing the R-R intervals.

(13) Measure the x -axis intervals between neighboring R peaks as follows:

$$\Delta R^C(k) = x_{k+1} - x_k, \quad k = 1, 2, \dots, K-1. \quad (15)$$

(14) Generate the final R feature, $R^F(x)$, by repeating the validation and update for all candidate peaks of $R^C(x)$. Each peak $R^C(x_k)$ can be classified into one of three categories according to the value of the $\Delta R^C(k)$:

- (a) $R^C(x_k)$ is not included to $R^F(x)$ if the x -axis intervals between two neighboring peaks, $\Delta R^C(k-1)$ or $\Delta R^C(k)$, are less than a given threshold; that is,

$$R^C(x_k) \notin R^F(x), \quad \text{if } \Delta R^C(k-1) < \theta_{\Delta 1} \text{ or } \Delta R^C(k) < \theta_{\Delta 1}, \quad (16)$$

where $\theta_{\Delta 1}$ is a threshold value that determines the minimum offset between R-R peaks. This threshold value can be set as follows:

$$\theta_{\Delta 1} = \alpha * \mu_{\Delta}, \quad (17)$$

where $\alpha = 0.5$ and μ_{Δ} is the average spacing of R-R peaks.

- (b) Find additional R peaks between $R^C(x_{k-1})$ and $R^C(x_{k+1})$ when x -axis intervals with two neighboring peaks are greater than a given threshold:

$$E^F(x_a) \in R^F(x), \quad x_{k-1} + \gamma * \mu_{\Delta} < x_a < x_{k+1} - \gamma * \mu_{\Delta}, \quad (18)$$

if $\Delta R^C(k-1) > \theta_{\Delta 2}$ or $\Delta R^C(k) > \theta_{\Delta 2}$,

where $\theta_{\Delta 2} = \beta * \mu_{\Delta}$ is a threshold value that determines the maximum interval between R-R peaks and γ is a parameter for defining the size of search area. We used β and γ as 1.5 and 0.5, respectively.

- (c) $R^C(x_k)$ is maintained as $R^F(x)$ when the x -axis intervals are located at the proper intervals:

$$R^C(x_k) \in R^F(x), \quad \text{if } \theta_{\Delta 1} \leq \Delta R^C(k) \leq \theta_{\Delta 2}, \quad (19)$$

$\theta_{\Delta 1} \leq \Delta R^C(k+1) \leq \theta_{\Delta 2}$.

3. Results

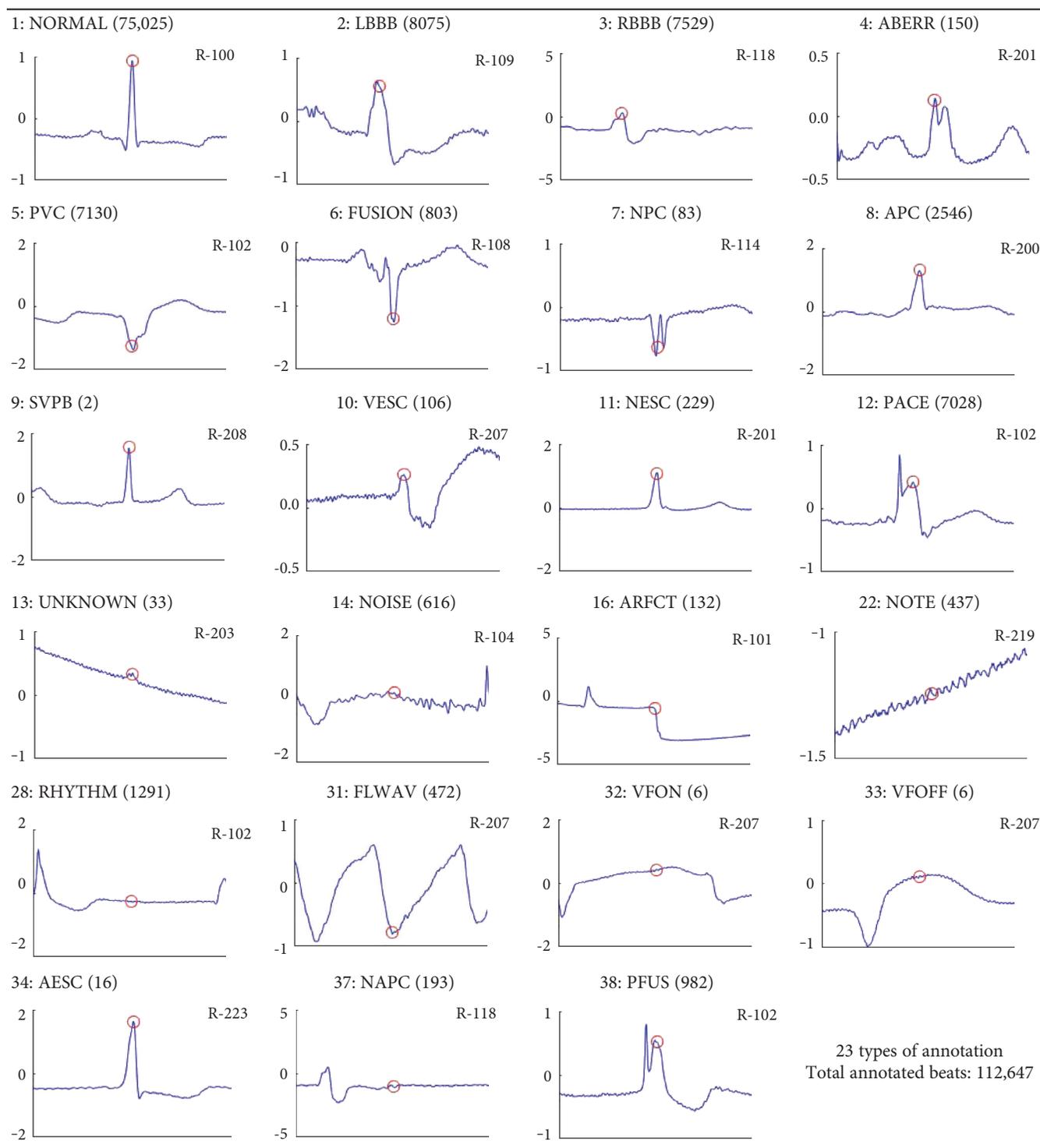
3.1. Experimental Data. We used the MIT-BIH arrhythmia database [28] from the PhysioNet site. This database was developed with the aim of benchmarking references for automated analysis of ECG signals. The MIT-BIH arrhythmia DB has 48 two-channel ambulatory ECG recordings of 30 min duration each. They are available at different frequencies and different time lengths. These signals are sampled at 360 samples per second and have a resolution of 11 bits over a 10 mV range. We used the first lead of all 48 records to validate the performance of the proposed method.

There are 112,647 labeled beats, where the ‘‘Normal beat’’ class contains 75,052 beats of annotation type ‘‘1.’’ Table 1 summarizes the annotations in the MIT-BIH arrhythmia DB and shows the examples of the 23 types of annotated beats. In the table, 1: NORMAL, 2: LBBB, ..., and 38: PFUS are the annotation types and their abbreviations, respectively (please refer to the site for the meaning, annotation types, and their abbreviations). And # indicates the number of corresponding beats in MIT-BIH DB, and R-# is the record number of illustrated data.

As shown in Table 1, there are many abnormal beats in the ECG signals, so considering all annotations to be the reference peaks is not the correct way to verify the performance of the R peak detection methods. To solve the problem of incorrect or ambiguous annotation, we apply the validation process to determine whether each annotated position is true peak or not.

3.2. R Peak Detection Results. Figure 6 shows examples of the detected R peaks from various records in the MIT-BIH DB. In this figure, black asterisks (*) denote the annotated beats in DB and red circles (O) denote the extracted peaks. As

TABLE 1: Summary and example of the 23 types of annotated beats in the MIT-BIH arrhythmia DB.



shown in the figure, the proposed method can detect normal R peaks under various conditions such as baseline drift, noisy signal, tall T waves as shown in Figure 6(c), or long paused waves as shown in Figure 6(h).

Figure 7 shows another example of the detected R peaks and annotated beats from various records in the MIT-BIH database. In this figure, numbers, such as 1, 2, 3, 5, 14, and

28, indicate the annotation types of each beat. As shown in the figure, the proposed method can remove abnormal beats at the initial position of the ECG signal and correct the location errors of record 117 as shown in Figure 7(c).

Despite the superiority of the proposed method, detection errors still exist. Figure 8 demonstrates examples of false positives (FP) and false negatives (FN) of the proposed

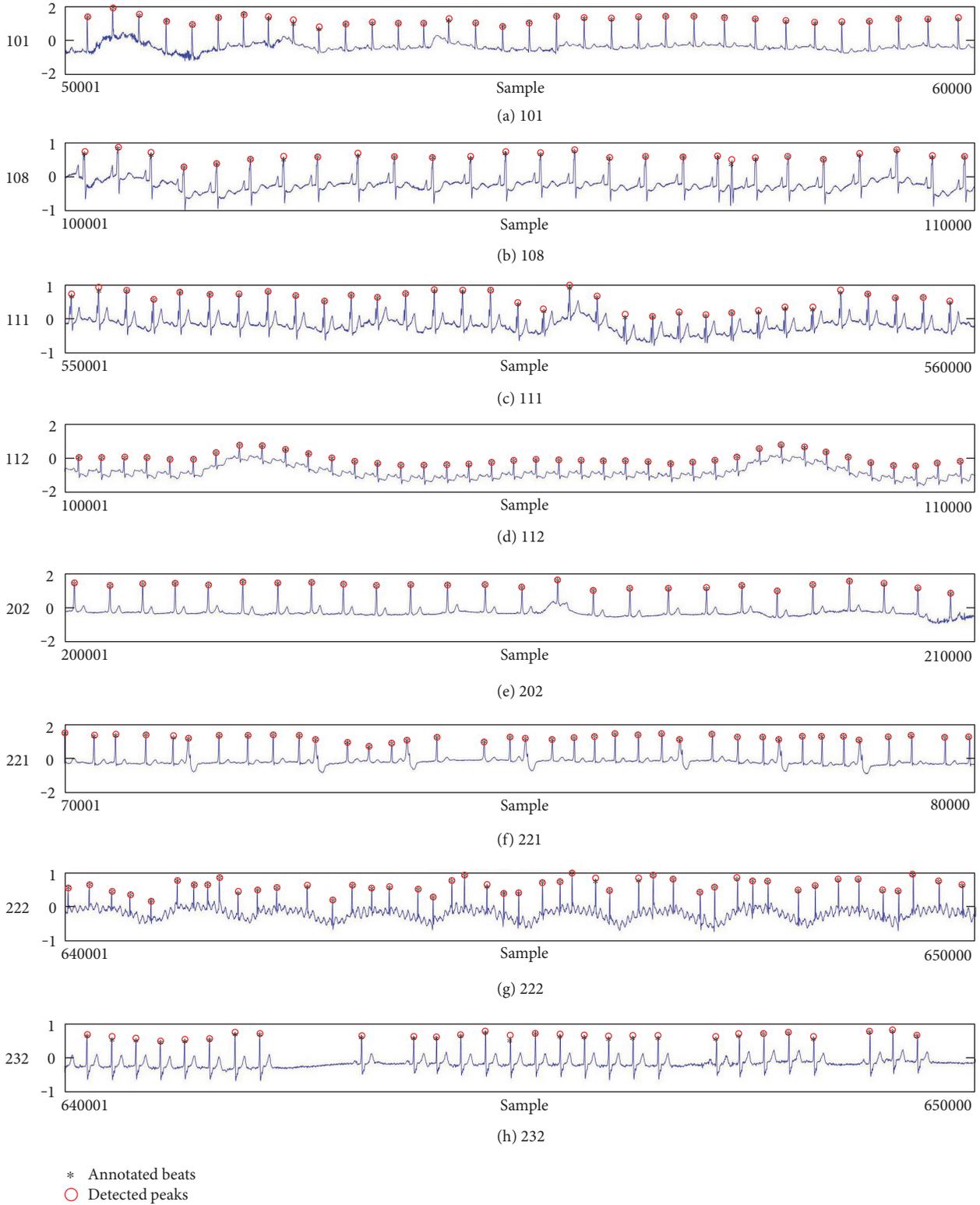


FIGURE 6: Examples of the detected R peaks from various normal records: (a) 101, (b) 108, (c) 111, (d) 112, (e) 202, (f) 221, (g) 222, and (h) 232. In the figure, black asterisks (*) denote the annotated beats in MIT DB and red circles (O) denote the extracted R peaks.

method. FP is mainly generated in a high-frequency region, and FN occurs in a small-noise area between R peaks. Especially, the DER of the proposed method appears the best among others.

3.3. *Evaluation Measure and Results.* To evaluate the performance of our proposed R peak detection method, we require three parameters—namely, true positive (TP), false negative (FN), and false positive (FP)—from the detected R peak.

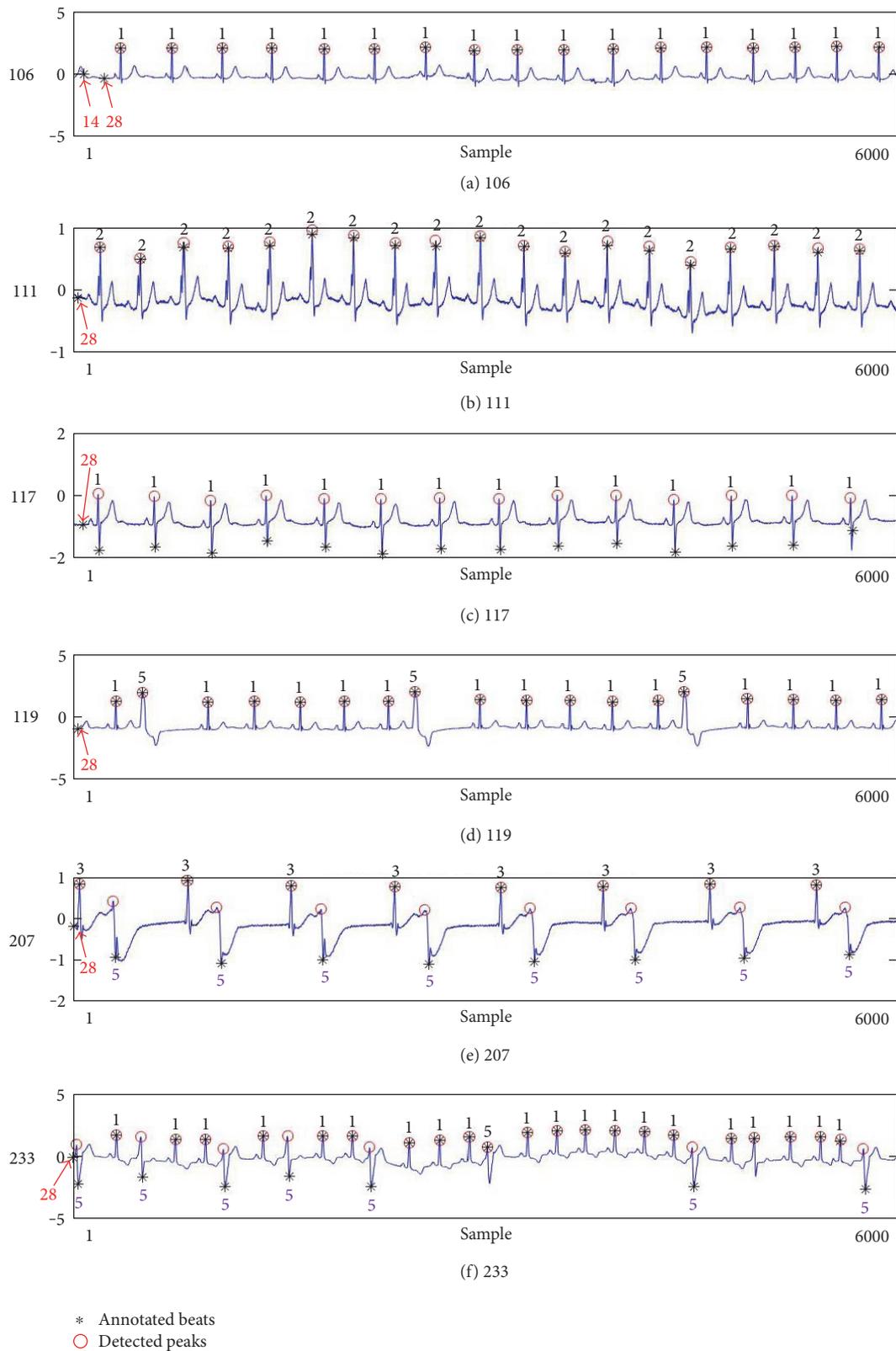


FIGURE 7: Examples of the annotated beats and detected real peaks from various records: (a) 106, (b) 111, (c) 117, (d) 119, (e) 207, and (f) 233. In the figure, the numbers such as 1, 2, 3, 5, 14, and 18 indicate the annotation types of each beat.

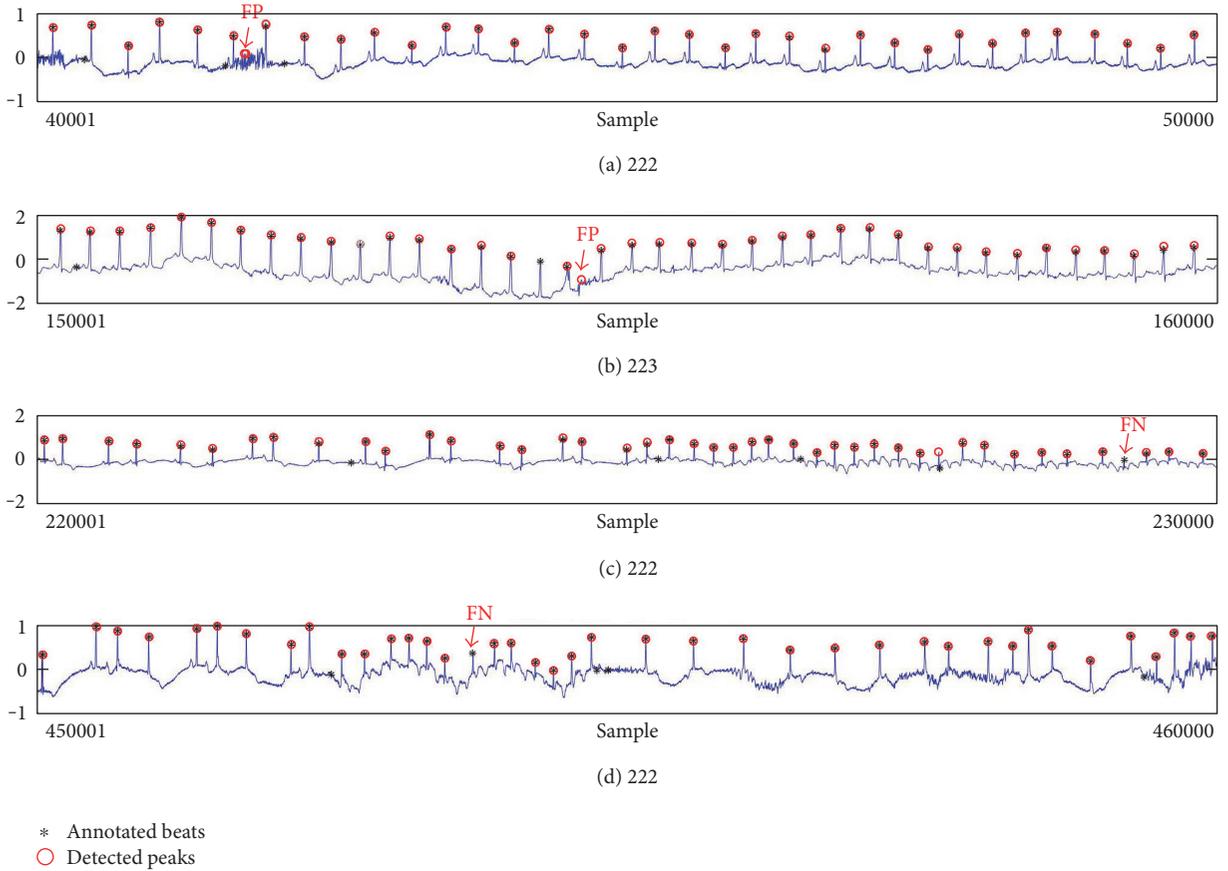


FIGURE 8: Detection failure of the proposed WTSEE method.

Here, TP is the number of correctly detected R peaks, FN is the number of missed R peaks, and FP is the number of noise spikes incorrectly detected as R peaks. Sensitivity (Se), positive predictability (+P), detection error rate (DER), and accuracy (Acc) can be computed by using TP, FN, and FP by using the following equations, respectively, as widely used in the literatures [1, 2, 4, 13, 24–26]:

$$\begin{aligned}
 \text{Se} &= \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%, \\
 +\text{P} &= \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\%, \\
 \text{DER} &= \frac{\text{FP} + \text{FN}}{\text{TP}} \times 100\%, \\
 \text{Acc} &= \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \times 100\%,
 \end{aligned} \tag{20}$$

where TP are the correctly detected beats (true positive), FP are falsely detected beats (false positive), and FN are the undetected beats (false negative). The sensitivity is used for evaluating the ability of the algorithm to detect true beats, the positive predictability is used for evaluating the ability of the algorithm to discriminate between true and false beats, and the error rate is used for evaluating the accuracy of the algorithm [25].

The performance of the proposed R peak detection method for 48 ECG recordings of the MIT-BIH arrhythmia

database is summarized in Table 2. The proposed method detects a total of 109,415 (In total, there are 116,137 annotated beats in MIT-BIH DB, but this number varies among different references due to the use of different steps and comparing tools [24]) true peaks. It also produces 79 FNs and 99 FPs. The average accuracy for the proposed method is 99.838%, the sensitivity is 99.93%, the positive predictability is 99.91%, and the detection error rate is 0.163%. The obtained performance is comparable to the best performances reported in the literature considering the processing speed and memory use. The high detection accuracy of the proposed method is especially meaningful because it is based on wavelet transform. Previous studies using wavelet transform did not show high performance compared with nonwavelet transform-based methods. Moreover, the final validation procedure proposed in our method verifies the validity of each detected peak and further improved the peak detection accuracy.

In Table 3, the performance of the proposed method on the MIT-BIH arrhythmia DB is compared with other existing methods. It shows that our proposed method has comparable accuracy to other conventionally used methods including wavelet transform techniques, the differential operation method, the Pan and Tompkins algorithm, SEHT, and the Shannon energy technique.

The previous four measures are related only to the R peaks. Recently, measures of HRV have been suggested, such as the mean of R-R intervals (MRR), standard deviation of

TABLE 2: Performance of the proposed R peak detection method using the MIT-BIH arrhythmia DB with length 30 min.

Record	Total beats (TP + FN)	TP (beats)	FN (beats)	FP (beats)	Se (%)	+P (%)	DER (%)	Acc (%)
100	2273	2272	1	0	99.96	100	0.044	99.956
101	1865	1864	1	3	99.95	99.84	0.215	99.786
102	2187	2187	0	0	100	100	0	100
103	2084	2084	0	0	100	100	0	100
104	2229	2228	1	5	99.96	99.78	0.269	99.731
105	2572	2568	4	17	99.84	99.34	0.818	99.189
106	2027	2024	3	5	99.85	99.75	0.395	99.606
107	2137	2136	1	1	99.95	99.95	0.094	99.906
108	1763	1759	4	5	99.77	99.72	0.512	99.491
109	2532	2532	0	1	100	99.96	0.039	99.961
111	2124	2123	1	0	99.95	100	0.047	99.953
112	2539	2539	0	0	100	100	0.000	100.000
113	1795	1794	1	0	99.94	100	0.056	99.944
114	1879	1878	1	2	99.95	99.89	0.160	99.841
115	1953	1953	0	3	100	99.85	0.154	99.847
116	2412	2400	12	2	99.50	99.92	0.583	99.420
117	1535	1532	3	1	99.80	99.93	0.261	99.740
118	2278	2278	0	2	100	99.91	0.088	99.912
119	1987	1986	1	2	99.95	99.90	0.151	99.849
121	1863	1862	1	1	99.95	99.95	0.107	99.893
122	2476	2476	0	0	100	100	0	100
123	1518	1518	0	2	100	99.87	0.132	99.868
124	1619	1619	0	4	100	99.75	0.247	99.754
200	2601	2600	1	2	99.96	99.92	0.115	99.885
201	1963	1960	3	0	99.85	100	0.153	99.847
202	2136	2135	1	0	99.95	100	0.047	99.953
203	2980	2959	21	8	99.30	99.73	0.980	99.029
205	2656	2655	1	0	99.96	100	0.038	99.962
207	1860	1858	2	5	99.89	99.73	0.377	99.625
208	2955	2946	9	3	99.70	99.90	0.407	99.594
209	3005	3005	0	1	100	99.97	0.033	99.967
210	2650	2649	1	2	99.96	99.92	0.113	99.887
212	2748	2748	0	0	100	100	0	100
213	3251	3251	0	0	100	100	0	100
214	2262	2262	0	1	100	99.96	0.044	99.956
215	3363	3362	1	0	99.97	100	0.030	99.970
217	2208	2208	0	1	100	99.95	0.045	99.955
219	2154	2154	0	0	100	100	0	100
220	2048	2048	0	0	100	100	0	100
221	2427	2427	0	1	100	99.96	0.041	99.959
222	2483	2483	0	1	100	99.96	0.040	99.960
223	2605	2605	0	0	100	100	0	100
228	2053	2052	1	12	99.95	99.42	0.634	99.370
230	2256	2256	0	2	100	99.91	0.089	99.911
231	1571	1571	0	0	100	100	0	100
232	1780	1780	0	2	100	99.89	0.112	99.888
233	3079	3076	3	2	99.90	100	0.163	99.838
234	2753	2753	0	0	100	100	0	100
Total	109,494	109,415	79	99	99.93	99.91	0.163	99.838

TABLE 3: Comparison of the proposed WTSEE method with other methods for detecting R peaks using the MIT-BIH database.

Method	Total beats ^a (TP + FN)	TP (beats)	FN (beats)	FP (beats)	Se (%) ^b	+P (%) ^b	DER (%) ^b	Acc (%) ^b
Proposed WTSEE	109,494	109,415	79	99	99.93	99.91	0.163	99.838
ISEE, 2016 [13]	109,532	109,474	58	116	99.95	99.89	0.159	99.841
PSEE, 2013 [23]	109,494	109,401	93	91	99.92	99.92	0.168	99.832
SEHT, 2012 [22]	109,496	109,417	79	140	99.93	99.87	0.200	99.800
DOM, 2008 [12]	109,809	109,751	58	166	99.95	99.85	0.204	99.796
STSE, 2014 [24]	108,494	108,323	171	97	99.84	99.91	0.247	99.753
WT, 2004 [16]	109,428	109,208	220	153	99.80	99.86	0.342	99.660
PT, 1985 [14]	109,809	109,302	507	277	99.54	99.75	0.717	99.288

^aTotal beats—the sum of total beats of all analyzed records in each study. ^bThese values are calculated by total beats.

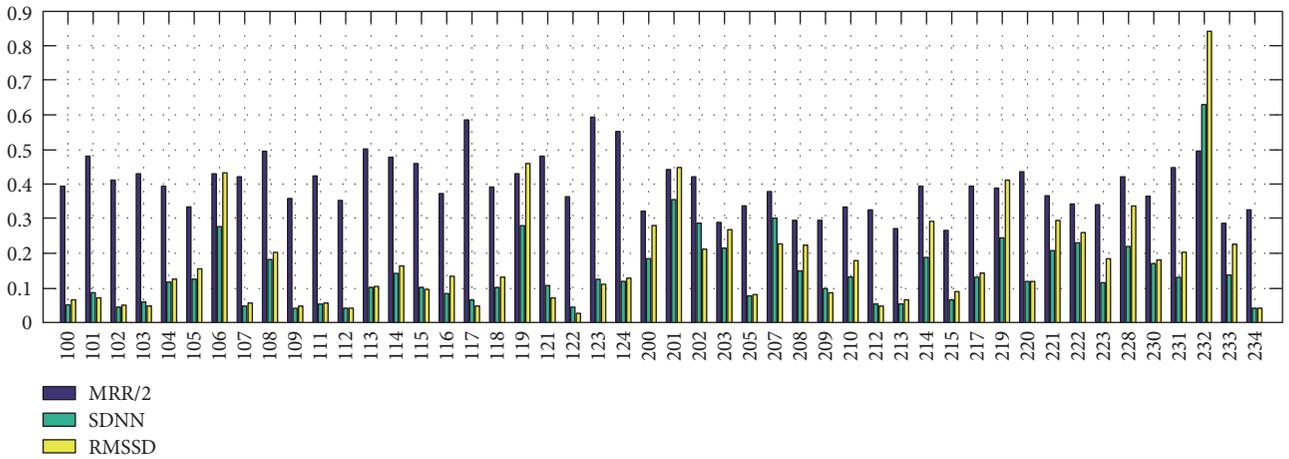


FIGURE 9: HRV measurements of the MIT-BIH DB. MRR: mean R-R intervals (ms), SDNN: standard deviation of normal to normal R-R intervals, RMSSD: root mean square of normal to normal R-R intervals.

normal to normal R-R intervals (SDNN), and root mean square of normal to normal RR intervals (RMSSD), where R is the peak of a QRS complex (heartbeat) [29].

(i) MRR is measured by

$$\text{MRR} = \bar{I} = \frac{1}{N-1} \sum_{n=2}^N I(n), \quad (21)$$

where N is the total heart beats ($N = \text{TP} + \text{FN}$), $I(n)$ is R-R interval between n th R peak and previous R peak, and \bar{I} is a mean of R-R intervals.

(ii) SDNN is defined by

$$\text{SDNN} = \sqrt{\frac{1}{N-1} \sum_{n=2}^N [I(n) - \bar{I}]^2}. \quad (22)$$

(iii) RMSSD is measured by

$$\text{RMSSD} = \sqrt{\frac{1}{N-2} \sum_{n=3}^N [I(n) - I(n-1)]^2}. \quad (23)$$

Figure 9 depicts the measured MRR, SDNN, and RMSSD from R-R intervals of the extracted R peaks. By investigating these measures, initial diagnosis of CVDs for patients can be achieved. In Figure 9, half of MRRs are illustrated to show three different kinds of measures together. The proposed method obtained MRR of 0.808, SDNN of 0.145, and RMSSD of 0.193, on average.

3.4. Experimental Environment and Real-Time Implementation. The proposed R peak detection method using WT and modified SEE has been implemented with MATLAB R2016b on a PC with a 3.3-GHz Intel Core i5-4590 CPU, 4GB of memory on Windows 7. The average processing time of our method is approximately 13.7 s on 48 ECG records in the MIT-BIH arrhythmia DB with a length of 30 minutes. As a result, the average throughput is 28.6 ms for an ECG signal. We showed exact parameter values used in our experiments around each of the equations throughout the manuscript to help researchers who want to implement our methods. It is recommended that researchers carefully follow the descriptions provided in Section 2.5 to obtain the same peak detection accuracies as those presented in this paper.

3.5. Discussions. This study presented our approach using WT and modified SEE for detecting R peaks of an ECG signal in the QRS complex. We used WT to take advantage of its representation power of temporal features at various resolutions. We applied WT to ECG signals to perform both down-sampling and noise reduction in a single step. This WT replaces the band-pass filtering used in a few representative ECG signal analysis methods [13, 25]. We also applied a soft-thresholding method for noise removal with keeping informative signals. The application of SEE and PEE can be considered as similar to the methods in [13]. However, we introduced additional postprocessing steps (Section 2.5) where the R-R peak intervals were inspected to determine whether to keep each detected R peak or detect additional R peaks in an R-R peak interval.

The novelty of our work is twofold: (1) Firstly, we proposed an ECG analysis technique that starts with WT method. There have been many WT-based methods, but their performances have been observed lower than those of non-WT-based methods [1, 4, 10, 13, 16–18, 25]. Our study demonstrates the possibility of obtaining high-peak detection accuracy in ECG signal analysis by using WT method. (2) Secondly, we proposed a postprocessing method that verifies the validity of each detected peak and indicates the probability of a missing peak.

The experimental results of our study show reasonable R peak detection performance with using less memory and processing time. This study has diverse applications in the initial diagnosis of CVDs for remote patients. The quantitative parameters (i.e., sensitivity, positive predictability, detection error rate, and accuracy) confirm acceptable performance of the proposed WTSEE detection method in practical applications.

4. Conclusion

In this paper, we presented a novel approach to detecting R peaks of an ECG signal and calculating the R-R interval of the R peaks via the wavelet transform (WT) method and a modified Shannon energy envelope (SEE) for rapid ECG analysis. The proposed method utilized WT and showed superior performance to other WT-based methods. We believe that the use of WT combined with subsequent processing steps properly analyzed the ECG signals to detect R peaks with high accuracy. Moreover, the proposed post-processing method effectively validated each detected peak and indicated miss detection to further improve the peak detection accuracy.

The proposed system was tested using the MIT-BIH arrhythmia DB. We analyzed the system performance for sensitivity, positive predictability, detection error rate, and accuracy. We compared the proposed approach with the existing techniques to validate the superior performance of the former. In the future, we would like to apply our analysis method to detecting other peak points in ECG signals. Detecting various types of peaks in ECG signal will provide more useful information for diagnosing CVDs and create a number of related applications.

Conflicts of Interest

The authors declare that there are no competing interests regarding the publication of this paper.

Acknowledgments

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (NRF-2012R1A1A3015620, NRF-2015R1D1A1A01060422) and by the Brain Research Program through the National Research Foundation of Korea funded by the Ministry of Science, ICT & Future Planning (NRF-2014M3C7A1046050).

References

- [1] Z. Zidelmal, A. Amirou, M. Adnane, and A. Belouchrani, "QRS detection based on wavelet coefficients," *Computer Methods and Programs in Biomedicine*, vol. 107, no. 3, pp. 490–496, 2012.
- [2] M. Merino, I. M. Gómez, and A. J. Molina, "Envelopment filter and K-means for the detection of QRS waveforms in electrocardiogram," *Medical Engineering and Physics*, vol. 37, no. 6, pp. 605–609, 2015.
- [3] R. B. K. Hoti and S. Khattak, "Automated noise removal and feature extraction in ECGs," *Arabian Journal for Science and Engineering*, vol. 39, no. 3, pp. 1937–1948, 2014.
- [4] H.-Y. Lin, S.-Y. Liang, Y.-L. Ho, Y.-H. Lin, and H.-P. Ma, "Discrete-wavelet-transform-based noise removal and feature extraction for ECG signals," *IRBM*, vol. 35, no. 6, pp. 351–361, 2014.
- [5] A. Ristovski, A. Guseva, M. Gusev, and S. Ristov, "Visualization in the ECG QRS detection algorithms," in *Proceedings of the 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO 2016)*, pp. 202–207, Opatija, Croatia, May–June 2016.
- [6] S. Yazdani and J.-M. Vesin, "Extraction of QRS fiducial points from the ECG using adaptive mathematical morphology," *Digital Signal Processing*, vol. 56, Issue C, pp. 100–109, 2016.
- [7] R. Rodríguez, A. Mexicano, J. Bila, S. Cervantes, and R. Ponce, "Feature extraction of electrocardiogram signals by applying adaptive threshold and principal component analysis," *Journal of Applied Research and Technology*, vol. 13, no. 2, pp. 261–269, 2015.
- [8] D. Ai, J. Yang, Z. Wang, J. Fan, C. Ai, and Y. Wang, "Fast multi-scale feature fusion for ECG heartbeat classification," *EURASIP Journal on Advances in Signal Processing*, vol. 2015, no. 46, 2015.
- [9] W.H.O, "World health statistics - part 6: SDG health and health-related targets," Tech. Rep., World Health Organization, 2016.
- [10] M. Merah, T. A. Abdelmalik, and B. H. Larbi, "R-peaks detection based on stationary wavelet transform," *Computer Methods and Programs in Biomedicine*, vol. 121, no. 3, pp. 149–160, 2015.
- [11] L. D. Sharma and R. K. Sunkaria, "A robust QRS detection using novel pre-processing techniques and kurtosis based enhanced efficiency," *Measurement*, vol. 87, pp. 194–204, 2016.

- [12] Y.-C. Yeh and W.-J. Wang, "QRS complexes detection for ECG signal: the difference operation method," *Computer Methods and Programs in Biomedicine*, vol. 91, pp. 245–254, 2008.
- [13] M. Rakshit, D. Panigrahy, and P. K. Sahu, "An improved method for R-peak detection by using Shannon energy envelope," *Sadhana. Academy Proceedings in Engineering Sciences*, vol. 41, no. 5, pp. 469–477, 2016.
- [14] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm," *IEEE Transactions on Biomedical Engineering (BME)*, vol. 32, no. 3, pp. 230–236, 1985.
- [15] F. Sufi, Q. Fang, and I. Cosic, "ECG R-R peak detection on mobile phones," in *Proceedings of 29th IEEE Conference on Engineering in Medicine and Biology Society (EMBS)*, pp. 3697–3700, Lyon, France, August 2007.
- [16] J. P. Martinez, S. Olmos, and P. Laguna, "Evaluation of a wavelet-based ECG waveform detector on the QT database," in *Computers in Cardiology*, pp. 81–84, Cambridge, MA, 2000.
- [17] P. Karthikeyan, M. Murugappan, and S. Yaacob, "ECG signal denoising using wavelet thresholding techniques in human stress assessment," *International Journal on Electrical Engineering and Informatics*, vol. 4, no. 2, pp. 306–319, 2012.
- [18] M. Yochum, C. Renaud, and S. Jacquir, "Automatic detection of P, QRS and T patterns in 12 leads ECG signal based on CWT," *Biomedical Signal Processing and Control*, vol. 25, pp. 46–52, 2016.
- [19] Y. Sun, K. L. Chan, and S. M. Krishnan, "Characteristic wave detection in ECG signal using morphological transform," *BMC Cardiovascular Disorders*, vol. 5, pp. 28–27, 2005.
- [20] J. A. Vav Alste and T. S. Schilder, "Removal of base-line wander and power-line interference from the ECG by an efficient FIR filter with a reduced number of taps," *IEEE Transactions on Biomedical Engineering (BME)*, vol. 32, no. 12, pp. 1052–1060, 1985.
- [21] P. Phukpattaranont, "QRS detection algorithm based on the quadratic filter," *Expert Systems with Applications*, vol. 42, no. 11, pp. 4867–4877, 2015.
- [22] M. S. Manikandan and K. P. Soman, "A novel method for detecting R-peaks in electrocardiogram (ECG) signal," *Biomedical Signal Processing and Control*, vol. 7, pp. 118–128, 2012.
- [23] H. Zhu and J. Dong, "An R-peak detection method based on peaks of Shannon energy envelope," *Biomedical Signal Processing and Control*, vol. 8, no. 5, pp. 466–474, 2013.
- [24] Z. Zidelmal, A. Amirou, D. Ould-Abdeslam, A. Moukadem, and A. Dieterlen, "QRS detection using S-transform and Shannon energy," *Computer Methods and Programs in Biomedicine*, vol. 116, no. 1, pp. 1–9, 2014.
- [25] H. Beyramienanlou and N. Lotfivand, "Shannon's energy based algorithm in ECG signal processing," *Computational and Mathematical Methods in Medicine*, vol. 2017, Article ID 8081361, 16 pages, 2017.
- [26] M. Baurner, "QT variability versus sympathetic cardiac activity in patients with major depression and patients with panic disorder," in *Proceedings of World Congress on Medical Physics and Biomedical Engineering (IFMBE)*, vol. 25, no. 4, pp. 1318–1320, Munich, Germany, September 2009.
- [27] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Engineering in Medicine and Biology*, vol. 20, no. 3, pp. 45–50, 2001, <https://physionet.org/physiobank/database/mitdb/>.
- [28] <http://wavelets.pybytes.com/wavelet/sym5/>.
- [29] H.-M. Wang and S.-C. Huang, "SDNN/RMSSD as a surrogate for LF/HF: a revised investigation," *Modelling and Simulation in Engineering*, vol. 2012, Article ID 931943, 8 pages, 2012.

Research Article

Hybrid Disease Diagnosis Using Multiobjective Optimization with Evolutionary Parameter Optimization

MadhuSudana Rao Nalluri,¹ Kannan K.,¹ Manisha M.,¹ and Diptendu Sinha Roy²

¹SASTRA University, Thanjavur, Tamil Nadu, India

²National Institute of Technology, Meghalaya, India

Correspondence should be addressed to Diptendu Sinha Roy; diptendu.sr@gmail.com

Received 10 December 2016; Revised 23 February 2017; Accepted 30 March 2017; Published 4 July 2017

Academic Editor: Ashish Khare

Copyright © 2017 MadhuSudana Rao Nalluri et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the widespread adoption of e-Healthcare and telemedicine applications, accurate, intelligent disease diagnosis systems have been profoundly coveted. In recent years, numerous individual machine learning-based classifiers have been proposed and tested, and the fact that a single classifier cannot effectively classify and diagnose all diseases has been almost accorded with. This has seen a number of recent research attempts to arrive at a consensus using ensemble classification techniques. In this paper, a hybrid system is proposed to diagnose ailments using optimizing individual classifier parameters for two classifier techniques, namely, support vector machine (SVM) and multilayer perceptron (MLP) technique. We employ three recent evolutionary algorithms to optimize the parameters of the classifiers above, leading to six alternative hybrid disease diagnosis systems, also referred to as hybrid intelligent systems (HISs). Multiple objectives, namely, prediction accuracy, sensitivity, and specificity, have been considered to assess the efficacy of the proposed hybrid systems with existing ones. The proposed model is evaluated on 11 benchmark datasets, and the obtained results demonstrate that our proposed hybrid diagnosis systems perform better in terms of disease prediction accuracy, sensitivity, and specificity. Pertinent statistical tests were carried out to substantiate the efficacy of the obtained results.

1. Introduction

The proliferations of computer usage across all aspects of life have resulted in accumulating a large number of systematic and related data. This has necessitated identifying useful patterns from raw datasets as the next logical step forward. Thus, data mining, a broad discipline encompassing classification, clustering, association, prediction, estimation, and visualization tasks [1], has emerged as a dynamic and significant field of research to address theoretical challenges as well as practical issues. Data mining and knowledge engineering techniques have been successfully applied to numerous areas, like education, pattern recognition, fraud detection, and medicine [2, 3].

The application of data mining and knowledge engineering techniques in the medical domain plays a prime role in the diagnosis of diseases and prognostication [4]. It assists healthcare professionals and doctors to analyze and predict

diseases [5] and is often commonly referred to as medical engineering. Numerous machine learning algorithms have been developed to extract useful patterns from raw medical data over the years [6]. These patterns have been utilized for disease prediction using classification and clustering strategies. Medical research focuses on employing data mining for prediction of a broad range of diseases, including breast cancer [7], heart diseases [8], Parkinson's disease [9], hepatitis, and diabetes, only to name a few.

Over the years, several supervised machine learning techniques such as classification as well as several unsupervised machine learning techniques like clustering have been applied to available medical information [10, 11]. Individual classifiers, ensembles thereof, and hybrid systems have often been used to diagnose various diseases. Several techniques have been applied on medical data to improve such diagnosing efficacy, regarding performance parameters such as prediction accuracy, sensitivity, and specificity [12, 13].

This paper presents a hybrid system for diagnosis and prediction of numerous diseases using optimized parameters for classifiers. The classifier parameters are optimized using evolutionary algorithms to enhance classification performance. By juxtaposing the proposed parameter optimization step within existing classifier mechanisms, our method provides improved prediction accuracy. In this paper, 16 classifiers are executed in which two basics are with and without resampling, 6 hybrid intelligent systems without resampling and 6 hybrid intelligent systems with resampling technique. In summary, this paper presents a comparative analysis of parameter optimized versions of two classifiers, namely, support vector machine (SVM) and multilayer perceptron (MLP) for medical data. It has been concluded from experimental results presented in this paper that our proposed hybrid system outperforms state of the art (single or ensemble) for classifying medical data. To contrive the parameter optimization, we have employed three popular evolutionary algorithms, namely, particle swarm optimization (PSO), gravitational search algorithm (GSA), and firefly algorithm (FA) for optimizing parameters of SVM and MLP classifiers. Accordingly, we study the performance of six alternative hybrid systems for classifying medical data towards a diagnosis of such diseases. The performance of the proposed hybrid intelligent techniques is compared with the recent literature results (both simple and ensemble classifiers [14–16]). This hybrid intelligent system shows better performance than the recently published ensemble classifiers on 11 benchmark datasets.

The rest of this paper is organized as follows: A brief exposition of existing researches has been dealt with in Section 2, specifically focusing on several machine learning algorithms employed for processing medical datasets. The problem formulation of our proposed weighted multiobjective optimization for the classifying problem dealt with has been presented in Section 3. Section 4 provides the rudimentary steps and key features of the evolutionary algorithms employed for the parameter optimization of SVM and MLP classifiers, namely, particle swarm optimization (PSO), gravitational search algorithm (GSA), and firefly algorithm (FA). A very basic introduction of the two classifiers employed, namely, SVM and MLP, has been discussed in Section 5. Section 6 elaborately explains the development of the proposed hybrid classification system for disease diagnosis along with their key components and design principles involved. The performance of the proposed hybrid scheme is tested over 11 benchmark medical datasets, and Section 7 provides a brief account of the experimental setup and the experiments conducted and summarizes the results obtained. This section also presents a statistical analysis of obtained results for validating the acceptability of obtained statistical results. The conclusions of the research have been presented in Section 8.

2. Related Work

There have been abundant attempts to analyze and diagnose ailments employing machine learning algorithms. This section gives a summary of the efforts in this field to put

the contribution of our work in perspective. These researches, however, vary considerably in terms of classifiers applied and nature of systems employed; for example, some are simple and others are hybrid whereas some others present ensemble systems. There are also major varieties in terms of objective functions chosen, single or multiobjective formulation, the number of datasets on which these methods have been applied, performance parameters employed for validating the efficacy, and so forth.

Among the different disease datasets that have been studied in the literature, heart disease diagnosis has been very prominent within medical engineering circles, and a wide variety of machine learning techniques have been explored towards diagnosing the same. References [17–38] include some prominent contributions towards diagnosing heart diseases from various aspects using myriad machine learning techniques, details of which are presented hereafter. Chitra and Seenivasagam [18] proposed a cascaded neural network (CNN) classifier and support vector machine (SVM) to diagnose heart diseases. The performance of CNN and SVM was compared based on the accuracy, sensitivity, and specificity. Pattekari and Parveen [19] suggested an intelligent system, which used a naive Bayes classifier that was further improved by developing ensemble-based classifiers. Das et al. [17] developed a neural network ensemble model for heart disease diagnosis. The proposed technique used Statistical Analysis System (SAS) enterprise guide 4.3 programs for data preprocessing and SAS Enterprise miner 5.2 programs for recognizing the heart disease by combining three neural networks ensemble. The technique was further improved by combining other neural networks and was also used for various datasets. Das et al. [37] described an SAS-based Software 9.1.3 for diagnosing valvular heart diseases. The proposed method used a neural network ensemble. Predicted values, posterior probabilities, and voting posterior probabilities were applied.

Masethe and Masethe [21] used J48, naive Bayes, REPTREE, CART, and Bayes Net for diagnosing the efficacy of heart diseases. High accuracy was obtained using a J48 tree. Shaikh et al. [22] evaluated the performance of three classifiers, namely, k-NN, naive Bayesian, and decision tree based on four parameters, namely, precision, recall, accuracy, and *F*-measure. k-NN produced higher accuracy than other methods. Bhatla and Jyoti [26] compared naive Bayes, decision tree, and neural networks for the said diagnosis. For the decision tree, genetic algorithm and fuzzy logic were employed, and results presented used TANAGRA tool.

Kavitha and Christopher [23] performed classification of heart rate using a hybrid particle swarm optimization and fuzzy C-means (PSO-FCM) clustering. The proposed method performed feature selection using PSO. The fuzzy C-means cluster and classifier are combined to enhance the accuracy. Enhanced SVM was used for classifying heart diseases. The hybrid system could be trained to shorten the implementation time. Alizadehsani et al. [24] evaluated sequential minimal optimization (SMO), naive Bayes, bagging with SMO, and neural networks. They employed rapid miner tool, and high accuracy was obtained using bagging with SMO. Abhishek [38] employed j48, naive Bayes, neural

networks with all attributes for diagnosing heart diseases with the WEKA machine learning software and concluded that j48 outperformed others regarding accuracy.

Jabbar et al. [20] used association mining and genetic algorithm in conjunction with heart disease prediction. The proposed method used Gini index statistics for association algorithm and crossover, the mutation for the genetic algorithm. They further employed a feature selection technique for improved accuracy. Ordonez et al. [36] presented an improved algorithm to determine constrained association rules by two techniques: mapping medical data and identifying constraints. The proposed method used mining attributes. Constrained association rules and parameters were used for the mapping. The technique produced interesting results by comparing this association rule with classification rule. Shenfield and Rostami [25] introduced a multiobjective approach to the evolutionary design of artificial neural networks for predicting heart disease.

Parthiban and Subramanian [27] developed a coactive neurofuzzy inference system (CANFIS) for prediction of heart diseases. The proposed model combined CANFIS, neural network, and fuzzy logic. It was then integrated with a genetic algorithm. Results showed that GA was useful for autotuning of the CANFIS parameters. Hedeshi and Abadeh [28] performed PSO algorithm with a boosting approach. The proposed method used fuzzy rule extraction with PSO and enhanced-particle swarm optimization 2 (En-PSO2). Karaolis et al. [35] used myocardial infarction (MI), percutaneous coronary intervention (PCI), and coronary artery bypass graft surgery (CABG) models. The proposed method used C4.5 decision tree algorithms. Results were compared based on false positive (FP), precision, and so forth. By further investigation with various datasets and employing extraction rule algorithms further, better results were obtained.

Kim et al. [30] proposed a fuzzy rule-based adaptive coronary heart disease prediction support model. The proposed method had three parts, namely, introducing fuzzy membership functions, a decision-tree rule induction technique, and fuzzy inference based on Mamdani's method. Outcomes were compared with neural network, logistic regression, decision tree, and Bayes Net. Chaurasia and Pal [31] offered three popular data mining algorithms: CART (classification and regression tree), ID3 (iterative dichotomized 3), and decision table (DT) for diagnosing heart diseases, and the results presented demonstrated that CART obtained higher accuracy within less time.

Olaniyi et al. [29] used neural network and support vector machine for heart diseases. Their proposed method used multilayer perceptron and demonstrated that SVM produced high accuracy. Yan et al. [32] proposed that multilayer perception with hidden layers is found by a cascade process. For the inductive reasoning of the methods, the proposed method used three assessment procedures, namely, cross-validation, hold out, and five bootstrapping samples for five intervals. Yan et al. [33] utilized multilayer perception for the diagnosis of five different cases of heart disease. The method employed a cascade learning process to find hidden layers and used back propagation for training the datasets.

Further improvements to the accuracy were achieved by parameter adjustments. Shouman et al. [34] identified gaps in the research work for heart disease diagnosis. The proposed method applied both single and hybrid data mining techniques to establish baseline accuracy and compared. Based on the research, hybrid classifier produced higher accuracy than a single classifier.

Sartakhti et al. [39] presented a method for diagnosis of hepatitis by novel machine learning methods that hybridize support vector machine and simulated annealing process. The proposed method used two hyperparameters for radial basis function (RBF) kernel: C and γ . For all potential combinations of C and γ interval, k -fold cross-validation score had been calculated. Results demonstrated that tuning SVM parameters by simulated annealing increased the accuracy. Çalişir et al. [40] developed the principle component analysis and least square support vector machine (PS-LLSVM). The suggested method was carried out in two steps: (1) the feature extraction from hepatitis disease database and feature reduction by PCA and (2) the reduced features are fed to the LSSVM classifier. Li and Wong [41] proposed C4.5 and PCL classifier. The outcomes were compared between C4.5 (bagging, boosting, and single tree) and PCL, and it was concluded that PCL produced higher accuracy than C4.5 based on their observations.

Weng et al. [42] investigated the performance of different classifiers which predicts Parkinson's disease. The proposed method used an ANN classifier based on the evaluation criteria. Jane et al. [43] proposed a Q-back propagated time delay neural network (Q-BTDNN) classifier. It developed temporal classification models that performed the task of classification and prognostication in clinical decision-making system. It used to feed forward time-delay neural network (TDNN) where training was imparted by a Q-learning-induced back propagation (Q-BP) technique. A 10-fold-cross-validation was employed for assessing the classification model. The results obtained were considered for comparative analysis, and it produced high accuracy. Gürüler [44] described a combination of the k -means clustering-based feature weighting (KMCFW) method and a complex-valued artificial neural network (CVANN). The suggested method considered five different evaluation methods. The cluster centers were estimated using the KMC. Results obtained showed very high accuracy.

Bashir et al. [45] presented an ensemble framework for predicting people with diabetes with multilayer classification using enhanced bagging and optimized weighting. The proposed HM-BagMOOV method used KNN approach for missing data imputation and had three layers, namely, layer 1 containing naive Bayes (NB), quadratic discriminant analysis (QDA), linear regression (LR), instance-based learning (IBL), and SVM; layer 2 included ANN and RF; and layer 3 used multilayer weighted bagging prediction. The outcome showed that it produced good accuracy for all datasets. Iyer et al. [46] prescribed a method to diagnose the disease using decision tree and naive Bayes. The proposed method used 10-fold cross-validation. The technique had been further enhanced by using other classifiers and neural network

techniques. Choubey and Sanchita [47] used genetic algorithm and multilayer perceptron techniques for the diagnosis of diabetics. The suggested methodology was implemented in two levels where genetic algorithm (GA) was used for feature selection and multilayer perceptron neural network (MLP NN) was used for classification of the selected characteristics. The results produced excellent accuracy that was further increased by considering receiver operating characteristic (ROC).

Kharya [48] used various data mining techniques for the diagnosis and prognosis of cancer. The proposed method used neural network, association rule mining, naïve Bayes, C4.5 decision tree algorithm, and Bayesian networks. The results showed that decision tree produced better accuracy than other classifiers. Chaurasia and Pal [49] investigated the performance of different classification techniques on breast cancer data. The proposed method used three classification techniques, namely, SMO, k -nearest neighbor algorithm (IBK), and best first (BF) tree. The results demonstrated that SMO produced higher accuracy than the other two techniques. In this article [50], an expert system (ES) is proposed for clinical diagnosis which is helpful for decision making in primary health care. The ES proposed used a rule-based system to identify several diseases based on clinical test reports.

Alzubaidi et al. studied ovarian cancer well [51]. In this work, features are selected using a hybrid global optimization technique. The hybridization process has involved mutual information, linear discriminate analysis, and genetic algorithm. The performance of the proposed hybrid technique is compared with support vector machine. This hybrid technique has shown significant performance improvements than support vector machine.

Gwak et al. [52] have proposed an ensemble framework for combining various crossover strategies using probability. The performance of this context had tested over 27 benchmark functions. It showed outperformance on eight tough benchmark functions. This ensemble framework further can be efficiently used for feature selection of big datasets.

Hsieh et al. [53] have developed an ensemble machine learning model for diagnosing breast cancer. In this model, information-gain has been adopted for feature selection. The list of classifiers used for developing ensemble classifier is neural fuzzy (NF), k -nearest neighbor (KNN), and the quadratic classifier (QC). The performance of ensemble framework is compared with individual classifier performance. The results demonstrate that ensemble framework has shown better performance than single classifier.

Review of existing literature for disease diagnosis techniques with machine learning indicates that there exists a plethora of individual classifiers as well as ensemble techniques. However, from such studies, it was also been conclusively evident that no individual classifier gives high prediction accuracy for different disease datasets. This has led to abundant ensemble classifiers for disease diagnosis, compromising the simplicity that an individual classifier offers. To this end, this paper indulges in designing a hybrid system that focuses on providing generalized

performance across a broad range of benchmark datasets. The most significant contribution of the proposed hybrid disease classifiers is that unlike most research works mentioned before that targets a specific disease, this paper validates the efficacy of the proposed hybrid classifiers across six different diseases collected over eleven datasets. For instance, among all heart disease, related diagnosis systems only [33] consider five different datasets for the said disease. Also, there are very few attempts in validating diagnosis efficacy over multiple diseases. Shen et al. [54] and Bashir et al. [14] are few exceptions that validate their results for four and five different diseases, respectively. The proposed classifiers employ novel parameter optimization approaches using a few recent evolutionary algorithms, detailed design of which has been presented in subsequent sections.

3. Problem Formulation

In this paper, we deal with classifying data from different disease datasets using a hybrid technique that optimizes the parameters of SVM and MLP classifiers for improved disease prediction. The list of objective functions to be targeted while solving the said classification problem include (i) prediction accuracy, (ii) specificity, and (iii) sensitivity, which has been considered very commonly for this problem in existing literature [55–57]. Each of these objective functions captures some aspect of quality of disease classification. In this sense, the problem studied in this paper is a multiobjective optimization problem.

All the aforementioned measures are computed in terms of the following values: true positive (TP), true negative (TN), false positive (FP), and false negative (FN), and their significance is defined as follows: TP: total number of positives that are correctly identified as positive; TN: total number of negatives that are identified as negatives; FP: total number of negatives that are incorrectly identified as positives; and FN: total number of positives that are wrongly identified as negatives.

The objective functions considered for optimization in this work are prediction accuracy (PAC), specificity (SPY), and sensitivity (SEY). To model these functions, two random indicator variables are introduced for all the data objects to compute TP, TN, FP, and FN. These are X_{i1} and X_{i2} , where these are defined as follows:

$$\begin{aligned} X_{i1} &= I\{CL_i = PC_i = C_+\}; \\ X_{i2} &= I\{CL_i = PC_i = C_-\}, \end{aligned} \quad (1)$$

where C_+ represents the actual class label is positive (+), C_- represents the actual class label is negative (-), PC_i represents predicted class label of i th data object, and CL_i represents the actual class label of the i th data object. At any point of the time, the sum of the entire indicator random variable values is equal to 1; that is, $\sum_{j=1}^2 X_{ij} = 1, \forall i$.

Let the classifier being developed for classifying a given dataset be a binary classifier and the dataset has N instances with m_1 positive and m_2 negative instances. Therefore,

$$\begin{aligned}
TP &= \sum_{i=1}^N X_{i1}, \\
TN &= \sum_{i=1}^N X_{i2}, \\
FN &= m_1 - TP = m_1 - \sum_{i=1}^N X_{i1}, \\
FP &= m_2 - TN = m_2 - \sum_{i=1}^N X_{i2}.
\end{aligned} \tag{2}$$

The performance parameters for the classifiers can thus be obtained using the following three equations:

$$\begin{aligned}
\text{Prediction accuracy (PAC)} &= \frac{TP + TN}{TP + TN + FP + FN} \\
&= \frac{\sum_{i=1}^N X_{i1} + \sum_{i=1}^N X_{i2}}{m_1 + m_2},
\end{aligned} \tag{3}$$

$$\text{Specificity (SPY)} = \frac{TN}{TN + FP} = \frac{\sum_{i=1}^N X_{i2}}{m_2}, \tag{4}$$

$$\text{Sensitivity (SEY)} = \frac{TP}{TP + FN} = \frac{\sum_{i=1}^N X_{i1}}{m_1}. \tag{5}$$

The aim of this research is to arrive at optimal values of classifier parameters through evolution such that some maxima are attained for PAC, SPY, and SEY. It is worthwhile to mention that even different sets of classifier parameter values with same PAC can have different values for SPY and SEY. Thus, there exist tradeoffs among (3), (4), and (5).

Any multiobjective optimization problem can then be solved either by converting the objective functions into a single linear or nonlinear objective function or by computing Pareto fronts using the concept of nondominance [58].

In this paper, a linear combination of objective functions has been taken to form a single linear compound objective function due to the requirement of additional computational effort for finding Pareto fronts in every iteration.

$$\begin{aligned}
&\text{Maximize } Z = W_1 * \text{PAC} + W_2 * \text{SPY} + W_3 * \text{SEY}, \\
&\text{Maximize } Z = W_1 * \frac{\sum_{i=1}^N X_{i1} + \sum_{i=1}^N X_{i2}}{m_1 + m_2} + W_2 * \frac{\sum_{i=1}^N X_{i2}}{m_2} \\
&\quad + W_3 * \frac{\sum_{i=1}^N X_{i1}}{m_1}
\end{aligned} \tag{6}$$

subject to the constraints

$$W_1 + W_2 + W_3 = 1, \tag{7}$$

$$1 \geq W_i \geq 0 \quad \forall i, \tag{8}$$

$$U_i \geq \text{CLASSIFIER_PAR}_i \geq L_i \quad \forall i, \tag{9}$$

where CLASSIFIER_PAR_{*i*} is the *i*th sensitive parameter of the considered classifier, (7) represents the totality condition

of the weights, (8) guarantees the nonnegativity condition, and (9) checks that the *i*th classifier parameter values is within the specified bounds.

4. Evolutionary Algorithms

In this section, we present a summary of the three evolutionary algorithms employed to optimize the parameters of SVM and MLP for classifying medical datasets for disease diagnosis. The discussions are restricted only to provide a brief overview. Detailed information and possible variations of these algorithms are beyond the scope of this paper.

4.1. Gravitational Search Algorithm (GSA). Gravitational search algorithm (GSA) is one of the population-based stochastic search methods initially developed by Rashedi et al. in the year 2009 [59]. GSA is inspired by Newton's gravitational law in physics, where every particle in the universe attracts every other particle with a force that is directly proportional to the product of their masses and inversely proportional to the square of the distance between them. GSA has been successfully applied to solve several engineering optimization problems [60, 61].

In GSA, several masses are considered on a *d*-dimensional space. The position of each mass resembles a point in the solution space of the problem to be solved. The fitness values of the agent, worst (*t*), and best (*t*) are used to compute the force (*F*) of mass. Equations corresponding to these parameters are provided in

$$q_i(t) = \frac{\text{fit}_i(t) - \text{worst}(t)}{\text{best}(t) - \text{worst}(t)}, \tag{10}$$

$$M_i(t) = \frac{q_i(t)}{\sum_{j=1}^s q_j(t)}, \tag{11}$$

$$\text{best}(t) = \min\{\text{fit}_k(t) : \forall k\}, \tag{12}$$

$$\text{worst}(t) = \max\{\text{fit}_k(t) : \forall k\}. \tag{13}$$

To update the position of mass ($x_i^d(t+1)$), velocity ($v_i^d(t)$) needs to be updated first. The velocity of the mass at the time (*t* + 1) majorly depends on the values of velocity and acceleration at that time instant *t*. Acceleration of the *i*th mass at instant *t* is ($a_i^d(t)$) depending on forces of all other heavy masses based on (14). The equation corresponding to the acceleration is given in (15). Equations corresponding to updating process of mass position and mass velocity are provided in (16) and (17).

$$F_i^d(t) = \sum_{j \in \text{kbest}, j \neq i} \text{rand}_j \left(G(t) \frac{M_j(t)M_i(t)}{R_{ij}(t) + \epsilon} (x_j^d(t) - x_i^d(t)) \right), \tag{14}$$

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)}, \tag{15}$$

$$V_i^d(t+1) = \text{rand}_i \times V_i^d(t) + a_i^d(t), \tag{16}$$

$$X_i^d(t+1) = X_i^d(t) + V_i^d(t+1), \quad (17)$$

where rand_i and rand_j lie between 0 and 1. “ ϵ ” is a small value. The distance between agents i and j is denoted by $R_{ij}(t)$. The best k agents are denoted with $k\text{best}$. G is a gravitational constant which is initialized with G_0 at the beginning, and with the progress in time, the value of G decreases.

4.2. Particle Swarm Optimization (PSO). In 1995, Dr. Kennedy and Dr. Eberhart developed a population-based speculative computational optimization procedure called particle swarm optimization based on the social behavior of living organisms like fish schools and bird flocks [62]. In PSO, the particles are randomly initialized. Position and velocity of the particles are represented as X_i and V_i , respectively. The fitness function is computed for each particle. Personal best (pBest) and global best (gBest) are the two important factors in PSO. Each particle has its own personal best, which is the particles’ individual best so far achieved until a time instant t . Global best is the overall best of all particles upto the time instant t . The algorithm is executed for a certain number of iterations. At each iteration, velocity is updated for all particles using a velocity updating scheme [63] as depicted in

$$V_{id}(t) = w * V_{id}(t-1) + c_1 * \text{rand}() * (\text{pBest}_{id} - X_{id}(t-1)) + c_2 * \text{rand}() * (\text{gBest}_{id} - X_{id}(t-1)), \quad (18)$$

where w represents the inertia weight, c_1 and c_2 are the personal and global learning factors, and $\text{rand}()$ is a random number between [0,1].

The following equation updates the new position of the particle:

$$X_i^d(t) = X_i^d(t-1) + V_i^d(t). \quad (19)$$

The basic steps of PSO are given in Algorithm 1.

4.3. Firefly Algorithm (FA). The firefly algorithm is a recently proposed bioinspired, evolutionary metaheuristic that mimics the social behavior of firefly species. Fireflies produce short and rhythmic flashes, the pattern of which characterizes particular species. The artificial, firefly-inspired algorithm makes certain assumptions regarding its functioning, such as unisexual fireflies for ensuring that all artificial fireflies attract each other and that the attractiveness is proportional to their brightness to define the potential of relative firefly movements. The brightness of a firefly is defined based on the problem at hand that it needs to optimize. For the minimization problem, the brightness may be the reciprocal of the objective function value. The pseudocode of the basic firefly algorithm as given by Yang in [64] has been depicted in Algorithm 2. The list of equations used in firefly algorithm is given as follows:

$$X_i^{t+1} = X_i^t + V_i^t, \quad (20)$$

$$V_i^t = \beta_0 e^{-\gamma r^m} d(X_i - X_j) + \alpha(\text{rand} - 0.5), \quad (21)$$

```

For each particle
  Initialize position and velocity randomly
End
t = 1
Do
  For each particle
    Calculate fitness function
    If fitness value > pBest Then
      Set current fitness value as pBest
  End
  Update particle with best fitness value as gBest
  For each particle
    Calculate new velocity using equation (18)
    Update position using equation (19)
  End
  t = t + 1
While (t < maximum iterations)
Post process the result.

```

ALGORITHM 1: Pseudocode for particle swarm optimization.

$$r_{ij} = d(X_i - X_j) = \|X_i - X_j\| = \text{Euclidean distance between } X_i \text{ and } X_j, \quad (22)$$

$$\beta(r) = \beta_0 e^{-\gamma r^m} \text{ where } m \geq 1,$$

where $\beta_0, \gamma, \alpha \in [0, 1]$.

Each firefly position is updated based on (20). The velocity of i th firefly is based on a fraction of attractiveness in the distance between fireflies X_i and X_j in an m -dimensional space and also on α , a small random value in the range 0 to 0.2; the equations related to velocity are given in (21) and (22).

5. Classification Techniques

Two classification techniques are used, and the basic details of these techniques are discussed in the subsequent sections.

5.1. Multilayer Perceptron. Multilayer perceptron (MLP) is the most commonly used supervised feed forward artificial neural network. It is a modification of the linear perceptron algorithm. It consists of many nodes that are arranged in several layers. In general, MLP contains three or more processing layers: one input layer for receiving input features, one or more hidden layers, and an output layer for producing classification results based on the classes [65].

Each node is represented as an artificial neuron which converts the input features into output using a weighted sum of inputs and activation function.

The weighted input is given by

$$V_i = \sum W_{ij} X_j + \Theta_i, \quad (23)$$

$$Y_i = f_i(V_i),$$

```

Objective function  $f(X)$ ,  $X = (x_1, \dots, x_d)$ 
Randomly generate initial fireflies' positions as  $X_i$  ( $i = 1, 2, \dots, n$ )
Light intensity  $I_i$  at  $x_i$  is determined by  $f(x_i)$ 
Define light absorption coefficient,  $\gamma$ 
Old_Solutions =  $\{X_1, X_2, \dots, X_N\}$ 
While ( $t < \text{MaxGeneration}$ )
  New_sol = {};
  for  $i = 1 : n$ 
    for  $j = 1 : i$ 
      if ( $f(X_j) > f(X_i)$ ) then
         $X = \text{Move}_d(X_i, X_j)$ ;
        New_sol = New_sol  $\cup$   $\{X\}$ ;
      End if
    Update( $r, e^{-\gamma r}$ )
    Evaluate new solutions, update light intensity
  End for  $j$ 
End for  $i$ 
 $Z = \text{Rank}(\text{Old\_Solutions}, \text{New\_sol})$ 
Old_Solutions =  $Z[1..n]$ 
End while
Post process results and visualization

```

ALGORITHM 2: Pseudocode representation of the basic firefly algorithm.

where V is the weighted sum of input features, W represents weights, X represents the input features, and Θ is the bias based on the classes.

The activation function is denoted by $f(x)$. The most frequently used activation functions are sigmoids. They are as follows:

$$\begin{aligned} f(v_i) &= \tanh(v_i), \\ f(v_i) &= (1 + e^{-v_i})^{-1}. \end{aligned} \quad (24)$$

The multilayer perceptron is trained using back propagation (BP). The weight update equation used in BP is given in

$$w_{ji} \leftarrow w_{ji} + \eta \delta_j x_{ji} + \alpha \Delta w_{ij} (n-1) \quad \text{where } 1 \geq \eta, \alpha \geq 0. \quad (25)$$

The parameter's learning rate (η) and momentum (α) are evolved using evolutionary algorithms presented in Section 4. A very basic MLP algorithm is provided in Algorithm 3 [66, 67].

There is a chance of being caught at a local minimum during the process of back propagation learning, and hence, to overcome it in this research article, learning rate and momentum values are evolved using 3 evolutionary search algorithms (CSO, IWO, and FF). A 3-layer neural network has been executed with an input layer, one hidden layer, and an output layer. The size of the input layer is equal to the number features of the data, and also, the size of the output layer is nothing but the number classes. The size of the hidden layer is the average of input and output layer sizes. Moreover, the performance of these three algorithms is compared in the simulation and is discussed in the Results and Discussion of this paper.

```

 $W = 0$  and  $\Theta = 0$ 
Repeat
For  $i = 1$  to  $L$  layers do
  If  $Y_i(W_{ij}X_i + \Theta_i) \leq \beta$  then
    For  $i = 1$  to  $N$  nodes do
      If  $|V_i \cdot X_i + d_i| \leq 1$  then
         $V_i = V_i + \lambda Y_i X_i$ 
         $d_i = d_i + \lambda Y_i$ 
      End if
    End for
     $\Theta = \Theta + \lambda Y_i$ 
  End if
End for
Until termination expression

```

ALGORITHM 3: MLP training algorithm.

5.2. Support Vector Machine. Support vector machine (SVM) is one of the supervised machine learning algorithms, which are often used for binary classifications. It was originally developed by Vapnik in 1979 [68]. The training data is in the form of instance-value pairs (x_i, y_i) . The SVM classifier finds an optimal hyperplane to separate negative and positive classes, and it is represented by $F(x) = w^t \cdot x + b = 0$.

Based on the class labels, two hyperplanes are formed, which are as follows:

$F(x) = w^t \cdot x + b \geq 0$ for positive instances ($y_j = +1$) and $F(x) = w^t \cdot x + b \leq 0$ for negative instances ($y_j = -1$), where w is the weight vector, x is input vector, and b is bias. Classifications are made on the hyperplanes thus formed.

The optimization problem formed during the development of soft margin classifier is as follows:

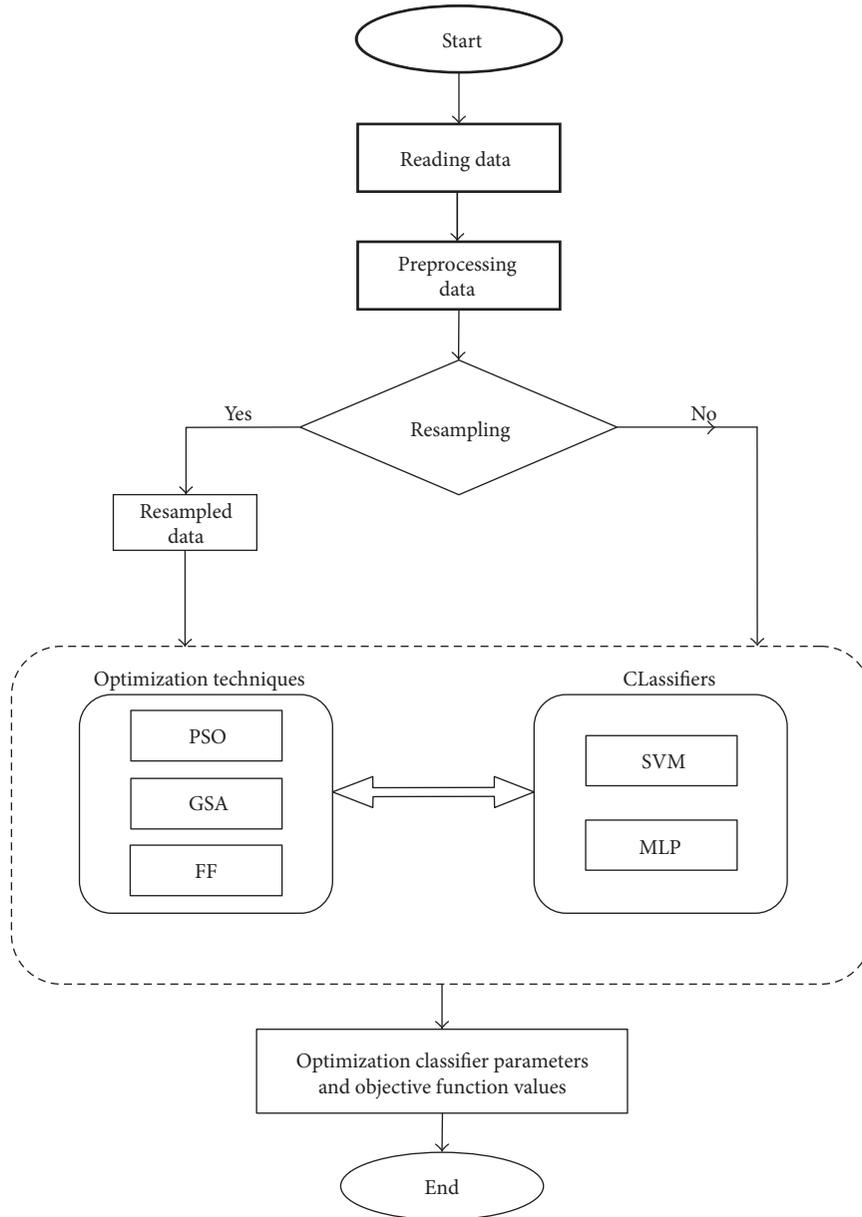


FIGURE 1: Schematic representing the flow of steps in the proposed hybrid disease diagnosis system.

$$\begin{aligned} \text{Minimize } Z &= \frac{1}{2} \langle w, w \rangle + C \sum_i \xi_i, \\ \text{subject to } Y_i (\langle w_i, x_i \rangle + b) &\geq (1 - \xi_i). \end{aligned} \quad (26)$$

The parameter cost (C) mentioned in (26) will be evolved using the evolutionary algorithms mentioned in Section 4.

6. Hybrid Intelligent System for Diagnosing Diseases

Diagnosing diseases from data collected from many patients with a varied degree of a specific disease is a classification problem. In medical information systems, single classifiers as well as ensemble classifiers have been studied for the disease diagnosis problem. In this section, we present the design

of hybrid systems that employ evolutionary algorithms as well as classification techniques to classify diseases based on data. A few hybrid systems have been developed to optimize the parameters of the classifiers [69, 70]; however, the premises of such classifiers are different application domains.

The performance of any classifier broadly depends on three factors, namely, the technique used for the classification; data statistics (regression, entropy, kurtosis, standard deviation, number of features considered for training, size of the training data, etc.); and parameters of the classifier (learning rate, depth of the tree, maximum number of child nodes allowed for a parent node in the decision tree, pruning, fuzzy membership functions, activation functions, etc.). In this paper, we focus on optimizing the parameter classifiers using evolutionary algorithms, and thus, our designed system qualifies as a hybrid system. Figure 1 illustrates a schematic

1. *Interpreting Data.*
2. *Pre-processing the data.*
3. *Select the classifier used for classification.*
4. *Identify the parameters to be optimized for the classifier chosen in step3.*
5. *Develop the objective function (Single/Multi).*
6. *Pick out the optimization technique to be used for parameter optimization of the classifier selected in step-3 based on the objective function constructed in step4.*
7. *Do the chosen optimization technique over the selected classifier for a finite iteration.*
8. *Post process the result obtained in step 7.*

ALGORITHM 4: Basic steps of hybrid intelligent system.

block diagram of the proposed hybrid system depicting the major steps to be carried out to arrive at disease diagnosis. The rectangle with dotted border illustrates the main emphasis of this paper. It represents that in this paper, we have studied how two classifiers, namely, SVM and MLP, perform as far as disease diagnosis is concerned. The parameters of these two classifiers have been optimized using three evolutionary algorithms, namely, PSO, GSA, and FA, with the goal of maximizing quality of diagnosis in terms of PAC, SPY, and SEY; or simply said, the goal is to optimize the three objectives as has been explained in Section 3. This has been depicted in the left half of the dotted rectangle in Figure 1. The basic steps involved in the hybrid system are summarized in Algorithm 4.

Preprocessing stage handles missing data of a feature by inserting most popular data or interval estimated data for that feature. As a part of preprocessing, features have also been normalized using min–max norm with the goal of reducing training phase time of classifiers, which takes quite some time due to the varied range of the feature values. In step 3, we have employed two classifiers (SVM and MLP). In step 4, the parameters selected for evolving in SVM are COST whereas for MLP, two parameters, namely, learning rate and momentum, have been selected for evolution. The range of these three parameters (cost, learning rate, and momentum) has been set as [0, 1]. In step 5, the objective function selected is either a single objective or multiobjective. If the method of optimization is multiobjective optimization, then for the sake of simplicity or uniformity, convert all the objective functions into either maximization or minimization. The multiple objectives considered for multiobjective optimization are given in (3)–(9). In step 6, three evolutionary algorithms (cat swarm optimization, gravitational search algorithm, and firefly algorithm) are selected as optimization techniques to find the optimum parameter values for the considered classifiers with respect to the multiple objectives: prediction accuracy, sensitivity, and specificity. Equations corresponding to multiobjective optimization are given in (3)–(9). In step 8, postprocessing of the results found in step 7 has to be done based on the optimization model selected in step 5. If the optimization model is single objective optimization, then to check the performance of the evolutionary algorithm, several statistical values like max, minimum, mean, median, and so forth have to be computed. If the selected optimization model is multiobjective (or weighted multiobjective) optimization model, the quality of

nondominated solutions must found using the metrics like spacing, generational distance, and so forth.

The hybridization process ensures that the population of the evolutionary algorithms is constructed based on the classifier parameters by satisfying parameter bounds. During the execution of evolutionary algorithms, population fitness is computed by substituting the performance parameter values of the classifier executed on the dataset in step 4.

Once all the three EAs are executed individually, optimal parameter values for each of the two classifiers (SVM and MLP) are found, and subsequently, these six HISs are compared based on their fitness values. That HIS having the best fitness value for a particular dataset is considered as the proposed HIS for that particular dataset. The objective function and parameter values of the best hybrid intelligent system are treated as final optimal values.

By combining the two classifiers and the three evolutionary optimization techniques for optimizing chosen classifier parameters, a number of hybrid intelligent systems have been obtained as possible alternatives. These alternative hybrid intelligent systems (HISs) have been termed as GSA-based SVM (GSVM), FA-based SVM (FSVM), PSO-based SVM (PSVM), GSA-based MLP (GMLP), FA-based MLP (FMLP), and PSO-based MLP (PMLP). These six HISs are tested on all the eleven benchmark datasets considered in this work, once without employing resampling and then using resampling technique. Hence, these HISs produce a set of sixteen results for each of the disease datasets, eight for SVM and eight for MLP. Out of these eight results, one is for the basic classifier (only SVM and only MLP) without data resampling, another for the same with resampling data, and the remaining six are for the three evolutionary algorithms each, once with original data and again with resampling data. The benchmark datasets are tested with ADABOOST version of SVM and MLP. However, on average, the ADABOOST results are not competitive with the instance-based supervised resampling technique in Weka, and the corresponding performances are given in Table 1. Moreover, we continued our experiments using instance-based supervised resampling technique.

7. Simulations and Results

To check the performance of the proposed hybrid system, 11 medical datasets of various diseases are considered. These data have been collected from the UCI repository [71], and

TABLE 1: Comparison of basic, resampling, and ADABOOST versions of SVM and MLP.

Dataset	MLP	SVM	RMLP	RSVM	ADA-MLP	ADA-SVM
Cleveland	79.2079	82.8383	93.72	85.0886	76.23	82.5083
Statlog	77.4074	84.07	86.6667	83.7037	77.777	84.07
Spect	79.4	81.65	88.38	88.764	79.4007	80.8989
Spectf	76.03	79.40	90.2622	82.397	76.03	77.9026
Eric	77.99	78.95	88.51	83.73	77.9904	77.9904
WBC	95.28	96.85	97.1388	96.5665	95.5651	96.7096
Hepatitis	81.94	85.16	90.3226	85.8065	78.7097	78.9097
Thyroid	96.28	89.77	98.1395	78.6047	97.2093	85.1163
Parkinson	91.28	86.15	96.4103	90.2564	92.3077	87.6923
Pima Indian diabetics	75.13	77.47	79.2969	76.0417	73.9583	77.3438
BUPA liver	71.59	70.14	68.1159	54.4928	71.3043	62.029

R: filter-based supervised instance resampling; ADA: ADABOOST.

TABLE 2: Summary of datasets used.

S. number	Dataset	Size
1	Cleveland	303×14
2	Statlog	270×14
3	Spect	267×23
4	Spectf	267×45
5	Eric	209×8
6	WBC	699×10
7	Hepatitis	155×20
8	Thyroid	215×6
9	Parkinson	195×23
10	Pima Indian diabetics	768×9
11	BUPA	345×7

the same form the basis of almost all performance evaluations in disease diagnosis. A detailed account of the datasets employed in this paper has been summarized in Table 2. All the six hybrid system alternatives and basic classifier technique have been executed on each of the 11 datasets, once without resampling of the dataset and then repeated with resampled dataset.

All the three evolutionary algorithms are executed for 50 iterations by considering 20 agents per iteration. These algorithms have been implemented in Java. Weka 3.7.4 tool class libraries have been used for the implementation of SVM and MLP. Instance-based resampling, which is available in Weka, has been used for resampling purposes. For experimentation purposes, the datasets considered are divided into testing and training sets, and a 10-fold cross-validation is used to that effect. To compare the performance of our proposed hybrid system for the datasets employed, we have compared results we have obtained with the results presented in three very recent papers that use the same datasets (not all 11 datasets, only a subset is utilized by these papers). References [14] through [16] are recent literature, and they have been referred to in our work as the *base papers* for every dataset (as has been earmarked in legends in Figure 2). The results of datasets corresponding to diseases like breast

cancer, hepatitis, BUPA liver, Pima, Cleveland, and Parkinson have been compared with those of [14], whereas the results of Statlog, Spect, Spectf, and Eric have been compared with those of BagMOOV [15]. Thyroid disease results alone are compared with those of [16]. In this work, the highest priority is given in favor of prediction accuracy. Hence, w_1 , w_2 , and w_3 in (6) correspond to 0.95, 0.05/2, and 0.05/2, respectively.

7.1. Statistical Analysis. In this section, we present a number of statistical analyses for the results obtained from our proposed hybrid system. The following subsections provide details about how these analyses are done.

7.1.1. Signed-Rank Test. The statistical analysis was done using Wilcoxon signed-rank test [72]. It tests the performance of all the techniques. The null hypothesis and alternative null hypothesis are set as follows:

$$H_0: \text{median}(X) \text{ is equal to } \text{median}(Y).$$

$$H_1: \text{median}(X) \text{ is not equal to } \text{median}(Y).$$

The objective values corresponding to FMLP overall disease datasets are tested over rest of the five techniques on all disease datasets, once with and next without resampling for each of the hybrid system alternatives, namely, FSVM, GSVM, PSVM, GMLP, and PMLP.

The Wilcoxon signed-rank test was executed with the level of significances 0.01 and 0.05. The Matlab function “signrank()” was used to perform the statistical analysis and the conclusions arrived upon has been presented in Tables 3, 4, 5, and 6.

7.1.2. Student’s t-Test. Student’s *t*-test is used to test whether the sample X derived from a normal distribution can have the mean m without knowing standard deviation [73]. We execute FMLP for 20 times, and we also noted the best performance in each iteration. Student’s *t*-test is executed on the three objectives: prediction accuracy (PAC), sensitivity

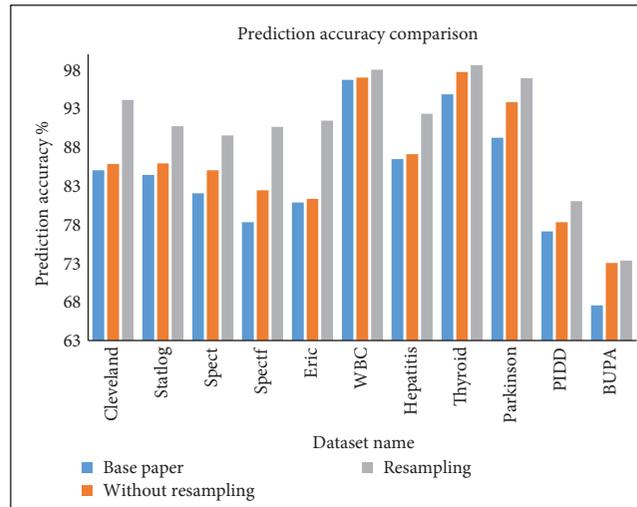


FIGURE 2: Comparison of all techniques based on prediction accuracy.

(SEN), and specificity (SPE). Null hypothesis and alternative hypothesis are set as follows:

$$H_0: \mu_X = m, H_1: \mu_X \neq m.$$

Student's t -test is performed by using `ttest()` function available in MATLAB. The performance of this test for various parameter values has been summarized in Tables 7 and 8.

7.2. Performance Metrics. The important distinct goals of multiobjective optimization are (1) finding solutions as close to the Pareto-optimal solutions as possible and (2) finding solutions as diverse as possible in the obtained nondominated front. In this work, to test the first goal is tested using generational distance (GD) and the second target is tested by computing spacing [58]. In the metric computation, two sets are used, namely, Q and P^* , where Q is the Pareto front found by test algorithm and P^* is the subset of true Pareto-optimal members. Before computing these metrics, the data in Q is to be normalized since various objective functions will have different ranges.

Generational distance (GD): Veldhuizen introduces this metric in the year 1990 [74]. This metric finds an average distance between the members of Q and P^* as follows:

$$GD = \left(\sum_{i=1}^{|Q|} d_i^p \right)^{1/p} / |Q|.$$

For $p=2$, the parameter d_i is the Euclidean distance between the members of Q and the

nearest member of P^* : $d_i = \min_{k=1}^{|P^*|} \sqrt{\sum_{m=1}^M (f_m^i - f_m^{*(k)})^2}$, where $f_m^{*(k)}$ is the m th objective function value of the k th member of P^* . An algorithm having a small value of GD is better. The members in P^* are having a maximum value for at least one objective function.

Spacing (SP): Schott introduces this metric in the year 1995 [75]. This metric finds the standard deviation of different d_i values. It can be calculated as follows: $S = \sqrt{(1/|Q|) \sum_{i=1}^{|Q|} (d_i - \bar{d})^2}$, where $d_i = \min_{k \in Q \cap k \neq i} \sum_{m=1}^M |f_m^i - f_m^k|$

and \bar{d} is the mean value of d_i 's. A good algorithm will be having a minimal SP value. The set Q is caught by executing FMLP for 50 iterations with each iteration having 20 agents. In every iteration, the Pareto fronts are stored in external memory. The metrics for GD and SP for all the three objectives with and without resampling are given in Table 9.

7.3. Results and Discussion. The best values found in all the hybrid systems are discussed as follows.

7.3.1. Cleveland Dataset. The performance of all the 8 techniques (2 basic machine learning and six hybrid systems) over Cleveland dataset is depicted in Table 10. PMLP shows best sensitivity (84.79%), whereas FMLP shows better results for all the other performance parameters, like accuracy (85.8%), specificity (87.5%), F -measure (85.74%), recall (85.8%), and precision (85.91%) without resampling. On the contrary, with resampling, PMLP shows the best accuracy (94.1%), but for all the other parameters, like sensitivity (93.49%), specificity (94.77%), F -measure (94.05%), recall (94.05%), and precision (94.07%), PMLP (GMLP, FMLP) shows best results. A comparison of Cleveland result with the state-of-the-art result is given in Table 11. Table 10 summarizes the performance of the proposed hybrid alternatives for the Cleveland dataset, and Table 11 compares this performance with best results obtained in recent literature.

7.3.2. Statlog Dataset. The performance of all the 8 techniques (2 machine learning and six hybrid systems) over Statlog dataset with and without resampling is given in Table 12. The highest accuracy, sensitivity, specificity, F -measure, recall, and precision without resampling are achieved by FMLP, PMLP, FMLP, FMLP, FMLP, and FMLP, respectively; best values found have been bolded for easy identification in Table 12. The highest accuracy, sensitivity, specificity, F -measure, recall, and precision with resampling are achieved by GMLP, GMLP, FMLP, GMLP, GMLP, and GMLP, respectively; best values found have been bolded for

TABLE 3: Signed-rank test at LOS 0.01 on resampled data.

Dataset	Objectives	PSVM		GSVM		FSVM		PMLP		GMLP	
		<i>P</i> value	<i>H</i>								
Cleveland	PAC	3.69E-05	1	3.69E-05	1	3.69E-05	1	8.75E-05	1	8.125E-5	1
	SEN	4.4E-05	1	2.98E-05	1	3.69E-05	1	6.875E-5	1	1.13E-5	1
	SPE	3.69E-05	1	4.4E-05	1	4.4E-05	1	8.125E-5	1	4.375E-5	1
Statlog	PAC	3.69E-05	1	2.98E-05	1	3.69E-05	1	6.11E-05	1	2.31E-05	1
	SEN	3.69E-05	1	3.69E-05	1	4.4E-05	1	3.69E-05	1	4.4E-05	1
	SPE	2.98E-05	1	4.4E-05	1	3.69E-05	1	4.4E-05	1	4.4E-05	1
Spect	PAC	4.4E-05	1	4.4E-05	1	4.4E-05	1	0.000963	1	5E-05	1
	SEN	4.4E-05	1	7.23E-05	1	4.4E-05	1	0.007533	1	0.009141	1
	SPE	2.98E-05	1	3.69E-05	1	3.69E-05	1	2.98E-05	1	3.69E-05	1
Spectf	PAC	2.98E-05	1	4.4E-05	1	3.69E-05	1	5.17E-05	1	0.000726	1
	SEN	4.4E-05	1	4.4E-05	1	8.51E-05	1	0.000629	1	0.00082	1
	SPE	4.4E-05	1	4.4E-05	1	3.69E-05	1	2.98E-05	1	0.000627	1
Eric	PAC	4.4E-05	1	4.4E-05	1	4.4E-05	1	0.875	0	3.69E-05	1
	SEN	4.4E-05	1	4.4E-05	1	4.4E-05	1	5.2E-05	1	3.69E-05	1
	SPE	3.69E-05	1	2.98E-05	1	2.98E-05	1	3.69E-05	1	3.69E-05	1
WBC	PAC	4.37E-05	1	3.55E-05	1	4.37E-05	1	1	0	5E-05	1
	SEN	4.4E-05	1	3.69E-05	1	4.4E-05	1	0.5625	0	0.006127	1
	SPE	4.4E-05	1	4.4E-05	1	4.4E-05	1	0.4375	0	6.11E-05	1
Hepatitis	PAC	4.4E-05	1	3.69E-05	1	3.69E-05	1	0.4375	0	8.4375	1
	SEN	4.37E-05	1	5.2E-05	1	3.69E-05	1	8.75E-6	1	5.7126E-5	1
	SPE	4.4E-05	1	3.69E-05	1	3.69E-05	1	1.15E-5	1	2.3123E-5	1
Thyroid	PAC	4.4E-05	1	3.69E-05	1	3.69E-05	1	4.4E-05	1	4.4E-05	1
	SEN	4.4E-05	1	4.4E-05	1	2.98E-05	1	4.4E-05	1	2.98E-05	1
	SPE	4.4E-05	1	3.69E-05	1	4.4E-05	1	6.577E-5	1	0.007533	1
Parkinson	PAC	4.4E-05	1	4.4E-05	1	3.69E-05	1	3.69E-05	1	4.37E-05	1
	SEN	4.4E-05	1	2.98E-05	1	4.4E-05	1	0.006855	1	3.2366E-5	1
	SPE	4.4E-05	1	4.4E-05	1	3.69E-05	1	2.98E-05	1	3.69E-05	1
Pima Indian diabetics	PAC	4.4E-05	1	4.4E-05	1	4.4E-05	1	3.69E-05	1	1E-04	1
	SEN	3.69E-05	1	3.69E-05	1	2.98E-05	1	3.69E-05	1	2.98E-05	1
	SPE	4.4E-05	1	4.4E-05	1	3.69E-05	1	3.69E-05	1	3.69E-05	1
BUPA liver disease	PAC	4.4E-05	1	3.69E-05	1	4.4E-05	1	4.4E-05	1	3.69E-05	1
	SEN	4.4E-05	1	3.69E-05	1	4.4E-05	1	4.4E-05	1	4.4E-05	1
	SPE	3.69E-05	1	3.69E-05	1	3.69E-05	1	2.31E-05	1	2.31E-05	1

easy identification in Table 12. A comparison of Statlog result with the state-of-the-art result is given in Table 13. The highest prediction accuracy for Statlog is 85.9% (without resampling) and 90.7 (with resampling). The performance of all the considered techniques over Statlog dataset with resampling is better than without resampling. Table 12 summarizes the performance of proposed hybrid alternatives for this dataset, and Table 13 compares this performance with best results obtained in recent literature.

7.3.3. Spect Dataset. The performance of all the 8 techniques (2 machine learning and six hybrid systems) over Spect dataset with and without resampling is given in Table 14. The highest accuracy, sensitivity, specificity, *F*-measure, recall, and precision without resampling are achieved by

FMLP, FSVM, FMLP, FMLP, FMLP, and FMLP, respectively, with the values 85%, 88.4%, 74.2%, 83.3%, 85%, and 83.9%. The highest accuracy, sensitivity, specificity, *F*-measure, recall, and precision with resampling are achieved by GMLP (PMLP), GMLP (FMLP, PMLP), GMLP (PMLP), PMLP (GMLP), GMLP (PMLP), and GMLP (PMLP), respectively, with the values 89.5%, 91.9%, 77.3%, 89.2%, 89.5%, and 89.1%. A comparison of Spect result with the state-of-the-art result is given in Table 15. The highest prediction accuracy for Spect is 85% (without resampling) and 89.5 (with resampling). The performance of all the considered techniques over Spect dataset with resampling is better than without resampling. Table 14 summarizes the performance of proposed hybrid alternatives for this dataset, and Table 15 compares this performance with best results obtained in recent literature.

TABLE 4: Signed-rank test at LOS 0.05 on resampled data.

Dataset	Objectives	PSVM		GSVM		FSVM		PMLP		GMLP	
		<i>P</i> value	<i>H</i>								
Cleveland	PAC	3.69E-05	1	3.69E-05	1	3.69E-05	1	8.75E-5	1	8.125E-5	1
	SEN	4.4E-05	1	2.98E-05	1	3.69E-05	1	6.875E-5	1	1.542E-4	1
	SPE	3.69E-05	1	4.4E-05	1	4.4E-05	1	8.125E-5	1	4.375	1
Statlog	PAC	3.69E-05	1	2.98E-05	1	3.69E-05	1	6.11E-05	1	2.31E-05	1
	SEN	3.69E-05	1	3.69E-05	1	4.4E-05	1	3.69E-05	1	4.4E-05	1
	SPE	2.98E-05	1	4.4E-05	1	3.69E-05	1	4.4E-05	1	4.4E-05	1
Spect	PAC	4.4E-05	1	4.4E-05	1	4.4E-05	1	0.000963	1	5E-05	1
	SEN	4.4E-05	1	7.23E-05	1	4.4E-05	1	0.007533	1	0.009141	1
	SPE	2.98E-05	1	3.69E-05	1	3.69E-05	1	2.98E-05	1	3.69E-05	1
Spectf	PAC	2.98E-05	1	4.4E-05	1	3.69E-05	1	5.17E-05	1	0.000726	1
	SEN	4.4E-05	1	4.4E-05	1	8.51E-05	1	0.000629	1	0.00082	1
	SPE	4.4E-05	1	4.4E-05	1	3.69E-05	1	2.98E-05	1	0.000627	1
Eric	PAC	4.4E-05	1	4.4E-05	1	4.4E-05	1	8.75E-5	1	3.69E-05	1
	SEN	4.4E-05	1	4.4E-05	1	4.4E-05	1	5.2E-05	1	3.69E-05	1
	SPE	3.69E-05	1	2.98E-05	1	2.98E-05	1	3.69E-05	1	3.69E-05	1
WBC	PAC	4.37E-05	1	3.55E-05	1	4.37E-05	1	1.3982E-5	1	5.38E-05	1
	SEN	4.4E-05	1	3.69E-05	1	4.4E-05	1	5.625E-5	1	0.6127	0
	SPE	4.4E-05	1	4.4E-05	1	4.4E-05	1	4.235E-5	1	6.11E-05	1
Hepatitis	PAC	4.4E-05	1	3.69E-05	1	3.69E-05	1	4.375E-5	1	8.475E-5	1
	SEN	4.37E-05	1	5.2E-05	1	3.69E-05	1	0.875E-5	1	1.873E-5	1
	SPE	4.4E-05	1	3.69E-05	1	3.69E-05	1	1.321E-5	1	1.098E-5	1
Thyroid	PAC	4.4E-05	1	3.69E-05	1	3.69E-05	1	4.4E-05	1	4.4E-05	1
	SEN	4.4E-05	1	4.4E-05	1	2.98E-05	1	4.4E-05	1	2.98E-05	1
	SPE	4.4E-05	1	3.69E-05	1	4.4E-05	1	0.026577	1	0.007533	1
Parkinson	PAC	4.4E-05	1	4.4E-05	1	3.69E-05	1	3.69E-05	1	4.37E-05	1
	SEN	4.4E-05	1	2.98E-05	1	4.4E-05	1	0.006855	1	0.032366	1
	SPE	4.4E-05	1	4.4E-05	1	3.69E-05	1	2.98E-05	1	3.69E-05	1
Pima Indian diabetics	PAC	4.4E-05	1	4.4E-05	1	4.4E-05	1	3.69E-05	1	1E-04	1
	SEN	3.69E-05	1	3.69E-05	1	2.98E-05	1	3.69E-05	1	2.98E-05	1
	SPE	4.4E-05	1	4.4E-05	1	3.69E-05	1	3.69E-05	1	3.69E-05	1
BUPA liver disease	PAC	4.4E-05	1	3.69E-05	1	4.4E-05	1	4.4E-05	1	3.69E-05	1
	SEN	4.4E-05	1	3.69E-05	1	4.4E-05	1	4.4E-05	1	4.4E-05	1
	SPE	3.69E-05	1	3.69E-05	1	3.69E-05	1	2.31E-05	1	2.31E-05	1

7.3.4. Spectf Dataset. The performance of all the 8 techniques (2 machine learning and six hybrid systems) over Spectf dataset with and without resampling is given in Table 16. The highest accuracy, sensitivity, specificity, *F*-measure, recall, and precision without resampling are achieved by FMLP, FSVM, FMLP, PSVM, FMLP, and FMLP, respectively, with the values 82.4%, 88%, 83.3%, 80.6%, 82.4%, and 82.6%. The highest accuracy, sensitivity, specificity, *F*-measure, recall, and precision with resampling are achieved by PMLP, GSVM, PMLP, PMLP, PMLP, and PMLP, respectively; best values found are bolded for easy identification in Table 16. A comparison of Spectf result with the state-of-the-art result is given in Table 17. The highest prediction accuracy for Spectf is 82.4% (without resampling) and 90.6% (with resampling). The performance of all the considered techniques over Spectf dataset with resampling

is better than without resampling except in specificity. Table 16 summarizes the performance of proposed hybrid alternatives for this dataset, and Table 17 compares this performance with best results obtained in recent literature.

7.3.5. Eric Dataset. The performance of all the 8 techniques (2 basic machine learning and six hybrid systems) over ERIC dataset is depicted in Table 18. FMLP shows best results for parameters like accuracy (81.34%), specificity (79.1%), *F*-measure (81.02%), and recall (81.34%), whereas GMLP shows better results for sensitivity (88.41%) and precision (82.5%) without resampling. On the contrary, with resampling, GMLP shows best results for parameters like accuracy (91.39%), sensitivity (88.78%), specificity (93.69%), *F*-measure (91.40%), recall (91.39%), and precision (91.48%). A comparison of ERIC result with the state-of-the-

TABLE 5: Signed-rank test at LOS 0.01 on without resampled data.

Dataset	Objectives	PSVM		GSVM		FSVM		PMLP		GMLP	
		<i>P</i> value	<i>H</i>								
Cleveland	PAC	3.69E-05	1	3.69E-05	1	3.69E-05	1	2.98E-05	1	2.98E-05	1
	SEN	4.4E-05	1	2.98E-05	1	4.4E-05	1	0.004662	1	8.51E-05	1
	SPE	4.4E-05	1	4.4E-05	1	4.4E-05	1	4.4E-05	1	3.69E-05	1
Statlog	PAC	2.98E-05	1	3.69E-05	1	3.69E-05	1	3.69E-05	1	4.4E-05	1
	SEN	0.035645	0	3.69E-05	1	6.14E-05	1	4.4E-05	1	2.98E-05	1
	SPE	4.4E-05	1	4.4E-05	1	4.4E-05	1	4.4E-05	1	3.69E-05	1
Spect	PAC	3.69E-05	1	4.4E-05	1	4.4E-05	1	4.4E-05	1	3.69E-05	1
	SEN	8.5E-05	1	3.69E-05	1	3.69E-05	1	4.4E-05	1	5.17E-05	1
	SPE	4.4E-05	1	4.4E-05	1	3.69E-05	1	3.69E-05	1	3.69E-05	1
Spectf	PAC	4.4E-05	1	4.4E-05	1	3.69E-05	1	4.4E-05	1	3.69E-05	1
	SEN	2.98E-05	1	4.4E-05	1	4.4E-05	1	4.4E-05	1	4.4E-05	1
	SPE	4.4E-05	1	3.69E-05	1	4.4E-05	1	2.98E-05	1	3.69E-05	1
Eric	PAC	4.4E-05	1	4.4E-05	1	4.4E-05	1	3.69E-05	1	4.375E-05	1
	SEN	4.4E-05	1	4.4E-05	1	6.14E-05	1	0.000542	1	4.4E-05	1
	SPE	4.4E-05	1	3.69E-05	1	3.69E-05	1	4.37E-05	1	3.69E-05	1
WBC	PAC	7.23E-05	1	0.000117	1	0.000943	1	0.005267	1	1E-04	1
	SEN	3.69E-05	1	4.4E-05	1	3.69E-05	1	0.000544	1	8.51E-05	1
	SPE	0.007263	1	0.032366	0	0.000826	1	0.00003	1	4.37E-05	1
Hepatitis	PAC	4.4E-05	1	3.69E-05	1	4.4E-05	1	4.4E-05	1	4.4E-05	1
	SEN	3.69E-05	1								
	SPE	5E-05	1	4.4E-05	1	6.11E-05	1	2.98E-05	1	3.69E-05	1
Thyroid	PAC	4.4E-05	1	4.4E-05	1	3.69E-05	1	4.4E-05	1	4.4E-05	1
	SEN	0.005738	1	0.000725	1	0.000726	1	3.69E-05	1	4.4E-05	1
	SPE	4.4E-05	1	4.4E-05	1	3.69E-05	1	4.37E-05	1	3.69E-05	1
Parkinson	PAC	4.4E-05	1	3.69E-05	1	4.4E-05	1	3.69E-05	1	4.4E-05	1
	SEN	3.69E-05	1	3.69E-05	1	3.69E-05	1	6.14E-05	1	4.359E-05	1
	SPE	4.4E-05	1	3.69E-05	1	4.4E-05	1	4.4E-05	1	3.69E-05	1
Pima Indian diabetics	PAC	4.4E-05	1	3.69E-05	1	3.69E-05	1	3.69E-05	1	4.4E-05	1
	SEN	3.69E-05	1								
	SPE	3.69E-05	1	3.69E-05	1	3.69E-05	1	3.69E-05	1	3.55E-05	1
BUPA liver disease	PAC	4.4E-05	1	3.69E-05	1	4.4E-05	1	4.4E-05	1	3.69E-05	1
	SEN	4.4E-05	1	0.00509	1	0.00646	1	4.4E-05	1	4.4E-05	1
	SPE	3.69E-05	1	3.69E-05	1	3.69E-05	1	6.11E-05	1	3.69E-05	1

art result is given in Table 19. Table 18 summarizes the performance of the proposed hybrid alternatives for the ERIC dataset, and Table 19 compares this performance with best performance in recent literature.

7.3.6. Wisconsin Breast Cancer (WBC) Dataset. The performance of all the 8 techniques (2 basic machine learning and six hybrid systems) over breast cancer dataset is depicted in Table 20. GMLP shows best accuracy (97%) and precision (97.04%), whereas PSVM shows better results for the parameters like sensitivity (95.08%), specificity (98.02%), and *F*-measure (97%), and GMLP and PSVM together show the best result for recall (97%) without resampling. On the contrary, with resampling, FMLP shows the best accuracy (98%), but for all the other parameters, like sensitivity (96.61%), *F*-measure (98%),

recall (98%), and precision (98%), PMLP (FMLP) shows best results and PSVM (GSVM and FSVM) shows best results for specificity (99.55%). A comparison of breast cancer result with the state-of-the-art result is given in Table 21. Table 20 summarizes the performance of the proposed hybrid alternatives for the breast cancer dataset, and Table 21 compares this performance with best results obtained in recent literature.

7.3.7. Hepatitis Dataset. The performance of all the 8 techniques (2 basic machine learning and six hybrid systems) over Hepatitis dataset is depicted in Table 22. PMLP shows best results for specificity (90.55%), *F*-measure (86.77%), recall (87.1%), and precision (86.6%), whereas GSVM (PSVM and FSVM) shows better results for the parameters like accuracy (87.1%) and sensitivity (73.08%) without

TABLE 6: Signed-rank test at LOS 0.05 on without resampled data.

Dataset	Objectives	PSVM		GSVM		FSVM		PMLP		GMLP	
		<i>P</i> value	<i>H</i>								
Cleveland	PAC	3.69E-05	1	3.69E-05	1	3.69E-05	1	2.98E-05	1	2.98E-05	1
	SEN	4.4E-05	1	2.98E-05	1	4.4E-05	1	0.004662	1	8.51E-05	1
	SPE	4.4E-05	1	4.4E-05	1	4.4E-05	1	4.4E-05	1	3.69E-05	1
Statlog	PAC	2.98E-05	1	3.69E-05	1	3.69E-05	1	3.69E-05	1	4.4E-05	1
	SEN	0.035645	1	3.69E-05	1	6.14E-05	1	4.4E-05	1	2.98E-05	1
	SPE	4.4E-05	1	4.4E-05	1	4.4E-05	1	4.4E-05	1	3.69E-05	1
Spect	PAC	3.69E-05	1	4.4E-05	1	4.4E-05	1	4.4E-05	1	3.69E-05	1
	SEN	8.5E-05	1	3.69E-05	1	3.69E-05	1	4.4E-05	1	5.17E-05	1
	SPE	4.4E-05	1	4.4E-05	1	3.69E-05	1	3.69E-05	1	3.69E-05	1
Spectf	PAC	4.4E-05	1	4.4E-05	1	3.69E-05	1	4.4E-05	1	3.69E-05	1
	SEN	2.98E-05	1	4.4E-05	1	4.4E-05	1	4.4E-05	1	4.4E-05	1
	SPE	4.4E-05	1	3.69E-05	1	4.4E-05	1	2.98E-05	1	3.69E-05	1
Eric	PAC	4.4E-05	1	4.4E-05	1	4.4E-05	1	3.69E-05	1	0.4375	0
	SEN	4.4E-05	1	4.4E-05	1	6.14E-05	1	0.000542	1	4.4E-05	1
	SPE	4.4E-05	1	3.69E-05	1	3.69E-05	1	4.37E-05	1	3.69E-05	1
WBC	PAC	7.23E-05	1	0.000117	1	0.000943	1	0.005267	1	1E-04	1
	SEN	3.69E-05	1	4.4E-05	1	3.69E-05	1	0.000544	1	8.51E-05	1
	SPE	0.007263	1	0.032366	1	0.033826	1	0.039203	1	4.37E-05	1
Hepatitis	PAC	4.4E-05	1	3.69E-05	1	4.4E-05	1	4.4E-05	1	4.4E-05	1
	SEN	3.69E-05	1								
	SPE	5E-05	1	4.4E-05	1	6.11E-05	1	2.98E-05	1	3.69E-05	1
Thyroid	PAC	4.4E-05	1	4.4E-05	1	3.69E-05	1	4.4E-05	1	4.4E-05	1
	SEN	0.005738	1	0.000725	1	0.000726	1	3.69E-05	1	4.4E-05	1
	SPE	4.4E-05	1	4.4E-05	1	3.69E-05	1	4.37E-05	1	3.69E-05	1
Parkinson	PAC	4.4E-05	1	3.69E-05	1	4.4E-05	1	3.69E-05	1	4.4E-05	1
	SEN	3.69E-05	1	3.69E-05	1	3.69E-05	1	6.14E-05	1	0.043059	1
	SPE	4.4E-05	1	3.69E-05	1	4.4E-05	1	4.4E-05	1	3.69E-05	1
Pima Indian diabetics	PAC	4.4E-05	1	3.69E-05	1	3.69E-05	1	3.69E-05	1	4.4E-05	1
	SEN	3.69E-05	1								
	SPE	3.69E-05	1	3.69E-05	1	3.69E-05	1	3.69E-05	1	3.55E-05	1
BUPA liver disease	PAC	4.4E-05	1	3.69E-05	1	4.4E-05	1	4.4E-05	1	3.69E-05	1
	SEN	4.4E-05	1	0.00509	1	0.00646	1	4.4E-05	1	4.4E-05	1
	SPE	3.69E-05	1	3.69E-05	1	3.69E-05	1	6.11E-05	1	3.69E-05	1

resampling. On the contrary, with resampling, FMLP (PMLP and GMLP) shows best results for parameters like accuracy (92.26%), sensitivity (80.77%), specificity (94.57%), *F*-measure (92.14%), recall (92.26%), and precision (92.08%). A comparison of Hepatitis result with the state-of-the-art result is given in Table 23. Table 22 summarizes the performance of the proposed hybrid alternatives for the Hepatitis dataset, and Table 23 compares this performance with best results obtained in recent literature.

7.3.8. Thyroid Dataset. The performance of all the 8 techniques (2 machine learning and 6 hybrid systems) over thyroid dataset with and without resampling is given in Table 24. The highest accuracy, sensitivity, specificity, *F*-measure, recall, and precision without resampling is achieved by FMLP, FMLP (PMLP), FMLP, FMLP, FMLP (PMLP), and

FMLP (PMLP), respectively, best values found have been bolded for easy identification in Table 24. The highest accuracy, sensitivity, specificity, *F*-measure, recall, and precision with resampling are achieved by PMLP (fMLP), FMLP (PMLP), PMLP (FMLP), PMLP (FMLP), PMLP (FMLP), and PMLP (FMLP), respectively, with the values 98.6%, 98.2%, 98.74%, 98.6%, 98.6%, and 98.6%. A comparison of thyroid result with the state-of-the-art result is given in Table 25. The highest prediction accuracy for thyroid is 97.7% (without resampling) and 98.6% (with resampling). The performance of all the considered techniques over thyroid dataset with resampling is better than without resampling. Table 24 summarizes the performance of proposed hybrid alternatives for this dataset, and Table 25 compares this performance with best results obtained in recent literature.

TABLE 7: Results of Student's t -test on without resampled data.

Dataset name	Obj	Alpha = 0.01				Alpha = 0.05			
		H	P value	Lower bound	Upper bound	H	P value	Lower bound	Upper bound
Cleveland	PAC	0	0.107409014	93.63760552	94.15613048	0	0.107409014	93.70719481	94.08654119
	SEN	0	0.119440215	93.37174786	93.52239194	0	0.119440215	93.39196524	93.50217457
	SPE	0	0.110406058	94.4269811	94.85972715	0	0.110406058	94.48505832	94.80164993
Statlog	PAC	0	0.198452296	89.40087856	89.71350963	0	0.198452296	89.4428356	89.67155259
	SEN	0	0.113684201	87.8202939	88.13645358	0	0.113684201	87.8627245	88.09402298
	SPE	0	0.085172965	90.21689475	90.78345951	0	0.085172965	90.29293128	90.70742299
Spect	PAC	0	0.09899382	88.79823642	89.22375777	0	0.09899382	88.85534404	89.16665015
	SEN	0	0.102037465	91.52271275	91.98167028	0	0.102037465	91.58430772	91.92007531
	SPE	0	0.094031375	75.19513272	75.64666638	0	0.094031375	75.25573136	75.58606775
Spectf	PAC	0	0.091447714	89.90632221	90.34258368	0	0.091447714	89.96487122	90.28403467
	SEN	0	0.085170746	93.2108793	93.62575965	0	0.085170746	93.26655883	93.57008011
	SPE	0	0.086020283	75.50805229	76.1105871	0	0.086020283	75.58891622	76.02972317
Eric	PAC	0	0.123534674	89.66246147	90.03031881	0	0.123534674	89.71183022	89.98095006
	SEN	0	0.127198304	87.50689214	87.66497294	0	0.127198304	87.52810757	87.64375751
	SPE	0	0.166768736	91.88301746	91.98548668	0	0.166768736	91.89676947	91.97173467
WBC	PAC	0	0.104809721	97.89345297	98.02702302	0	0.104809721	97.91137891	98.00909708
	SEN	0	0.103386039	96.30307823	96.68720924	0	0.103386039	96.35463101	96.63565646
	SPE	0	0.108706462	98.37053699	98.78539395	0	0.108706462	98.42621338	98.72971756
Hepatitis	PAC	0	0.131198082	92.0423242	92.32294757	0	0.131198082	92.07998561	92.28528616
	SEN	0	0.270421552	80.5671699	80.85759447	0	0.270421552	80.60614669	80.81861768
	SPE	0	0.141041289	94.26694606	94.66132599	0	0.141041289	94.31987431	94.60839774
Thyroid	PAC	0	0.171022773	97.15734101	97.84211907	0	0.171022773	97.2492425	97.75021758
	SEN	0	0.189226475	94.54448673	94.93137205	0	0.189226475	94.59640916	94.87944963
	SPE	0	0.209391475	98.4352079	98.84065802	0	0.209391475	98.48962183	98.78624408
Parkinson	PAC	0	0.100403912	96.04769583	96.49955298	0	0.100403912	96.10833788	96.43891093
	SEN	0	0.118064733	97.14392547	97.46967293	0	0.118064733	97.18764281	97.42595559
	SPE	0	0.132191436	92.30860837	92.78786405	0	0.132191436	92.37292747	92.72354494
Pima Indian diabetics	PAC	0	0.092109816	80.171399	80.70052364	0	0.092109816	80.24241083	80.62951182
	SEN	0	0.085031456	76.12631776	76.69348033	0	0.085031456	76.20243452	76.61736358
	SPE	0	0.165042763	81.90646042	82.11053899	0	0.165042763	81.93384904	82.08315037
BUPA liver disease	PAC	0	0.155271161	70.23491732	70.49206105	0	0.155271161	70.26942761	70.45755076
	SEN	0	0.114021708	67.20708253	67.52742545	0	0.114021708	67.25007455	67.48443343
	SPE	0	0.145476728	73.12029125	73.35511283	0	0.145476728	73.15180577	73.32359831

7.3.9. *Parkinson Dataset.* The performance of all the 8 techniques (2 machine learning and six hybrid systems) over Parkinson dataset with and without resampling is given in Table 26. The highest accuracy, sensitivity, specificity, F -measure, recall, and precision without resampling are achieved by FMLP, FMLP, PSVM (FSVM, GSVM), FMLP, FMLP, and FMLP, respectively, with the values 93.8%, 96.6%, 96.2%, 93.9%, 93.8%, and 94%. The highest accuracy, sensitivity, specificity, F -measure, recall, and precision with resampling are achieved by GMLP, GMLP (FMLP, PMLP), PSVM (GSVM, FSVM), PMLP (GMLP), PMLP (GMLP), and PMLP (GMLP), respectively, with the values 96.9%, 97.4%, 100%, 96.9%, 96.9%, and 96.9%. A comparison of Parkinson result with the state-of-the-art result is given in Table 27. The highest prediction accuracy for Parkinson is 93.8% (without resampling) and 96.9% (with resampling).

The performance of all the considered techniques over Parkinson dataset with resampling is better than without resampling. Table 26 summarizes the performance of proposed hybrid alternatives for this dataset, and Table 27 compares this performance with best results obtained in recent literature.

7.3.10. *Pima Indian Diabetic Dataset.* The performance of all the 8 techniques (2 machine learning and six hybrid systems) over Pima dataset with and without resampling is given in Table 28. The highest accuracy, sensitivity, specificity, F -measure, recall, and precision without resampling are achieved by FMLP, FMLP, PSVM (FSVM, GSVM), FMLP, FMLP, and FMLP, respectively; best values found have been bolded for easy identification in Table 28. The highest accuracy, sensitivity, specificity, F -measure, recall and precision

TABLE 8: Results of Student's *t*-test on resampled data.

Dataset name	Obj	Alpha = 0.01					Alpha = 0.05				
		<i>H</i>	<i>P</i> value	Lower bound	Upper bound	<i>H</i>	<i>P</i> value	Lower bound	Upper bound		
Cleveland	PAC	0	0.107409014	93.63760552	94.15613048	0	0.107409014	93.70719481	94.08654119		
	SEN	0	0.119440215	93.37174786	93.52239194	0	0.119440215	93.39196524	93.50217457		
	SPE	0	0.110406058	94.4269811	94.85972715	0	0.110406058	94.48505832	94.80164993		
Statlog	PAC	0	0.198452296	89.40087856	89.71350963	0	0.198452296	89.4428356	89.67155259		
	SEN	0	0.113684201	87.8202939	88.13645358	0	0.113684201	87.8627245	88.09402298		
	SPE	0	0.085172965	90.21689475	90.78345951	0	0.085172965	90.29293128	90.70742299		
Spect	PAC	0	0.09899382	88.79823642	89.22375777	0	0.09899382	88.85534404	89.16665015		
	SEN	0	0.102037465	91.52271275	91.98167028	0	0.102037465	91.58430772	91.92007531		
	SPE	0	0.094031375	75.19513272	75.64666638	0	0.094031375	75.25573136	75.58606775		
Spectf	PAC	0	0.091447714	89.90632221	90.34258368	0	0.091447714	89.96487122	90.28403467		
	SEN	0	0.085170746	93.2108793	93.62575965	0	0.085170746	93.26655883	93.57008011		
	SPE	0	0.086020283	75.50805229	76.1105871	0	0.086020283	75.58891622	76.02972317		
Eric	PAC	0	0.123534674	89.66246147	90.03031881	0	0.123534674	89.71183022	89.98095006		
	SEN	0	0.127198304	87.50689214	87.66497294	0	0.127198304	87.52810757	87.64375751		
	SPE	0	0.166768736	91.88301746	91.98548668	0	0.166768736	91.89676947	91.97173467		
WBC	PAC	0	0.104809721	97.89345297	98.02702302	0	0.104809721	97.91137891	98.00909708		
	SEN	0	0.103386039	96.30307823	96.68720924	0	0.103386039	96.35463101	96.63565646		
	SPE	0	0.108706462	98.37053699	98.78539395	0	0.108706462	98.42621338	98.72971756		
Hepatitis	PAC	0	0.131198082	92.0423242	92.32294757	0	0.131198082	92.07998561	92.28528616		
	SEN	0	0.270421552	80.5671699	80.85759447	0	0.270421552	80.60614669	80.81861768		
	SPE	0	0.141041289	94.26694606	94.66132599	0	0.141041289	94.31987431	94.60839774		
Thyroid	PAC	0	0.171022773	97.15734101	97.84211907	0	0.171022773	97.2492425	97.75021758		
	SEN	0	0.189226475	94.54448673	94.93137205	0	0.189226475	94.59640916	94.87944963		
	SPE	0	0.209391475	98.4352079	98.84065802	0	0.209391475	98.48962183	98.78624408		
Parkinson	PAC	0	0.100403912	96.04769583	96.49955298	0	0.100403912	96.10833788	96.43891093		
	SEN	0	0.118064733	97.14392547	97.46967293	0	0.118064733	97.18764281	97.42595559		
	SPE	0	0.132191436	92.30860837	92.78786405	0	0.132191436	92.37292747	92.72354494		
Pima Indian diabetics	PAC	0	0.092109816	80.171399	80.70052364	0	0.092109816	80.24241083	80.62951182		
	SEN	0	0.085031456	76.12631776	76.69348033	0	0.085031456	76.20243452	76.61736358		
	SPE	0	0.165042763	81.90646042	82.11053899	0	0.165042763	81.93384904	82.08315037		
BUPA liver disease	PAC	0	0.155271161	70.23491732	70.49206105	0	0.155271161	70.26942761	70.45755076		
	SEN	0	0.114021708	67.20708253	67.52742545	0	0.114021708	67.25007455	67.48443343		
	SPE	0	0.145476728	73.12029125	73.35511283	0	0.145476728	73.15180577	73.32359831		

TABLE 9: Performance metric values on all datasets with and without resampling.

Dataset name	Without resampling		Resampling		Dataset name	Without resampling		Resampling	
	GD	SP	GD	SP		GD	SP	GD	SP
Cleveland	0.21	0.8	0.19	0.9	WBC	0.39	1.25	0.25	2.21
Statlog	0.39	1.02	0.16	1.91	Hepatitis	0.24	3.21	0.27	1.29
Spect	0.11	1.87	0.14	1.23	Thyroid	0.1	1.34	0.26	1.76
Spectf	0.18	1.98	0.12	1.2	Parkinson	0.22	1.8	0.18	2.92
Eric	0.29	2.02	0.13	1.92	Pima Indian diabetics	0.2	0.98	0.13	1.05
					BUPA	0.18	2.2	0.14	1.9

TABLE 10: Performance of hybrid systems on Cleveland dataset.

		Cleveland						
	Resampling	Without resampling	Resampling			Without resampling		
			PSO	GSA	FA	PSO	GSA	FA
		Basic SVM			Parameter optimized SVM			
PAC	85.14	82.83	86.79	86.13	86.13	83.49	82.83	83.49
Sensitivity	83.33	82.18	84.91	84.74	84.35	82.38	82.18	82.38
Specificity	87.80	83.72	89.51	88.09	88.70	85.03	83.72	85.03
<i>F</i> -measure	85.04	82.77	86.71	86.06	86.05	83.41	82.77	83.41
Recall	85.14	82.83	86.79	86.13	86.13	83.49	82.83	83.49
Precision	85.36	82.8	87.01	86.27	86.33	83.60	82.88	83.60
		Basic MLP			Parameter optimized MLP			
PAC	93.72	79.20	94.05	94.05	94.05	85.14	84.15	85.80
Sensitivity	93.45	80.23	93.49	93.49	93.49	84.79	84.11	84.57
Specificity	94.07	77.94	94.77	94.77	94.77	85.60	84.21	87.5
<i>F</i> -measure	93.72	79.18	94.05	94.05	94.05	85.11	84.12	85.74
Recall	93.72	79.20	94.05	94.05	94.05	85.14	84.15	85.80
Precision	93.73	79.18	94.07	94.07	94.07	85.16	84.16	85.91

TABLE 11: Comparison of hybrid systems with HMV [14] for Cleveland dataset.

	Method	Accuracy	Sensitivity	Specificity	<i>F</i> -measure
Base paper	Ensemble	85	83.82	88.41	82.15
Without resampling	FMLP	85.8	87.5	84.6	87.5
Resampling	FMLP (PMLP)	94.1	94.8	93.5	94.1

TABLE 12: Performance of hybrid systems on Statlog dataset.

		Statlog						
	Resampling	Without resampling	Resampling			Without resampling		
			PSO	GSA	FA	PSO	GSA	FA
		Basic SVM			Parameter optimized SVM			
PAC	85.19	84.44	85.56	85.56	85.56	84.44	84.81	84.44
Sensitivity	83.81	83.62	82.73	83.96	83.96	83.62	84.35	84.21
Specificity	86.06	85.06	87.50	86.59	86.59	85.06	85.16	84.62
<i>F</i> -measure	85.12	84.41	85.55	85.50	85.50	84.41	84.78	84.40
Recall	85.19	84.44	85.56	85.56	85.56	84.44	84.81	84.44
Precision	85.14	84.42	85.54	85.51	85.51	84.42	84.80	84.44
		Basic MLP			Parameter optimized MLP			
PAC	86.67	77.41	90.37	90.74	89.63	84.07	81.85	85.93
Sensitivity	82.61	73.60	91.26	92.16	88.07	85.32	79.34	83.61
Specificity	89.68	80.69	89.82	89.88	90.68	83.23	83.89	87.84
<i>F</i> -measure	86.70	77.45	90.31	90.67	89.61	83.97	81.86	85.94
Recall	86.67	77.41	90.37	90.74	89.63	84.07	81.85	85.93
Precision	86.77	77.54	90.41	90.82	89.61	84.16	81.87	85.96

TABLE 13: Comparison of hybrid systems with BagMOOV [15] for Statlog dataset.

	Method	Accuracy	Sensitivity	Specificity	<i>F</i> -measure
Base paper	Ensemble	84.4	86	86	86
Without resampling	FMLP	85.9	83.6	87.8	85.9
Resampling	GMLP	90.74	92.2	89.9	90.7

TABLE 14: Performance of hybrid systems on Spect dataset.

	Resampling	Spect						
		Without resampling	PSO	Resampling GSA	FA	Without resampling PSO	GSA	FA
		Basic SVM			Parameter optimized SVM			
PAC	88.39	81.65	88.39	88.39	88.39	82.77	82.40	82.77
Sensitivity	91.07	88.26	91.07	91.44	91.07	87.05	88.02	88.43
Specificity	74.42	55.56	74.42	73.33	74.42	60.47	58.00	58.82
<i>F</i> -measure	87.96	81.59	87.96	88.06	87.96	81.95	82.08	82.53
Recall	88.39	81.65	88.39	88.39	88.39	82.77	82.40	82.77
Precision	87.83	81.53	87.83	87.91	87.83	81.58	81.83	82.33
		Basic MLP			Parameter optimized MLP			
PAC	88.39	79.40	89.51	89.51	89.14	82.40	83.52	85.02
Sensitivity	91.44	87.56	91.93	91.93	91.89	87.33	87.17	86.44
Specificity	73.33	50.00	77.27	77.27	75.56	58.70	63.41	74.19
<i>F</i> -measure	88.06	79.60	89.17	89.17	88.83	81.80	82.58	83.33
Recall	88.39	79.40	89.51	89.51	89.14	82.40	83.52	85.02
Precision	87.91	79.82	89.07	89.07	88.71	81.43	82.28	83.92

TABLE 15: Comparison of hybrid systems with BagMOOV [15] for Spect dataset.

	Method	Accuracy	Sensitivity	Specificity	<i>F</i> -measure
Base paper	Ensemble	82.02	27.27	96.23	42.50
Without resampling	FMLP	85	86.4	74.2	83.3
Resampling	FMLP	89.5	91.9	77.3	89.2

TABLE 16: Performance of hybrid systems on Spectf dataset.

	Resampling	Spectf						
		Without resampling	PSO	Resampling GSA	FA	Without resampling PSO	GSA	FA
		Basic SVM			Parameter optimized SVM			
PAC	87.64	79.40	89.14	89.14	88.01	80.90	79.40	80.15
Sensitivity	92.92	87.56	94.71	94.71	92.96	87.10	87.56	88.04
Specificity	67.27	50.00	69.49	69.49	68.52	54.00	50.00	51.72
<i>F</i> -measure	87.77	79.60	89.39	89.39	88.10	80.56	79.60	80.34
Recall	87.64	79.40	89.14	89.14	88.01	80.90	79.40	80.15
Precision	87.93	79.82	89.80	89.80	88.20	80.28	79.82	80.56
		Basic MLP			Parameter optimized MLP			
PAC	90.26	76.03	90.64	89.89	90.26	80.52	80.90	82.40
Sensitivity	93.15	85.24	93.58	93.12	93.55	87.74	84.85	82.35
Specificity	77.08	42.11	77.55	75.51	76.00	52.73	55.56	83.33
<i>F</i> -measure	90.11	76.19	90.53	89.77	90.19	80.52	79.31	77.56
Recall	90.26	76.03	90.64	89.89	90.26	80.52	80.90	82.40
Precision	90.02	76.35	90.46	89.69	90.13	80.52	78.81	82.55

TABLE 17: Comparison of hybrid systems with BagMOOV [15] for Spectf dataset.

	Method	Accuracy	Sensitivity	Specificity	<i>F</i> -measure
Base paper	Ensemble	78.28	7.27	96.70	13.53
Without resampling	FMLP	82.4	82.4	83.3	77.6
Resampling	PMLP	90.6	93.6	77.6	90.5

TABLE 18: Performance of hybrid systems on Eric dataset.

	Resampling	Eric						
		Without resampling	PSO	Resampling GSA	FA	Without resampling PSO	GSA	FA
		Basic SVM			Parameter optimized SVM			
PAC	84.21	78.95	85.17	85.65	85.17	80.86	78.95	80.86
Sensitivity	80.20	83.33	80.00	81.37	81.19	86.11	83.33	86.11
Specificity	87.96	76.64	90.38	89.72	88.89	78.10	76.64	78.10
<i>F</i> -measure	84.25	78.49	85.20	85.68	85.20	80.45	78.49	80.45
Recall	84.21	78.95	85.17	85.65	85.17	80.86	78.95	80.86
Precision	84.47	79.59	85.71	85.97	85.43	81.63	79.59	81.63
		Basic MLP			Parameter optimized MLP			
PAC	88.52	77.99	89.95	91.39	89.95	80.86	81.34	81.34
Sensitivity	85.00	77.38	86.87	88.78	87.63	85.14	88.41	85.33
Specificity	91.74	78.40	92.73	93.69	91.96	78.52	77.86	79.10
<i>F</i> -measure	88.54	77.85	89.97	91.40	89.96	80.51	80.84	81.02
Recall	88.52	77.99	89.95	91.39	89.95	80.86	81.34	81.34
Precision	88.71	77.95	90.09	91.48	90.01	81.43	82.50	81.85

TABLE 19: Comparison of hybrid systems with BagMOOV [15] for Eric dataset.

	Method	Accuracy	Sensitivity	Specificity	<i>F</i> -measure
Base paper	Ensemble	80.86	86.32	73.91	79.64
Without resampling	FMLP	81.3	88.4	77.9	80.8
Resampling	GMLP	91.4	88.8	93.7	91.4

TABLE 20: Performance of hybrid systems on WBC dataset.

	Resampling	Breast cancer						
		Without resampling	PSO	Resampling GSA	FA	Without resampling PSO	GSA	FA
		Basic SVM			Parameter optimized SVM			
PAC	96.14	96.85	97.00	97.00	97.00	97.00	96.85	96.85
Sensitivity	92.59	94.69	92.43	92.43	92.43	95.08	94.69	94.69
Specificity	98.03	98.02	99.55	99.55	99.55	98.02	98.02	98.02
<i>F</i> -measure	96.15	96.86	97.02	97.02	97.02	97.00	96.86	96.86
Recall	96.14	96.85	97.00	97.00	97.00	97.00	96.85	96.85
Precision	96.21	96.87	97.17	97.17	97.17	97.01	96.87	96.87
		Basic MLP			Parameter optimized MLP			
PAC	97.14	95.28	98.00	97.71	98.00	96.57	97.00	96.42
Sensitivity	95.34	92.62	96.61	96.58	96.61	93.93	94.35	93.55
Specificity	98.06	96.70	98.70	98.28	98.70	98.01	98.45	98.00
<i>F</i> -measure	97.14	95.29	98.00	97.71	98.00	96.58	97.01	96.44
Recall	97.14	95.28	98.00	97.71	98.00	96.57	97.00	96.42
Precision	97.15	95.30	98.00	97.71	98.00	96.60	97.04	96.47

TABLE 21: Comparison of hybrid systems with HMV [14] for WBC dataset.

	Method	Accuracy	Sensitivity	Specificity	<i>F</i> -measure
Base paper	Ensemble	96.71	98.01	96.94	97.48
Without resampling	PSVM	97	95.1	98	97
Resampling	PSVM	98	96.6	98.7	98

TABLE 22: Performance of hybrid systems on hepatitis dataset.

	Hepatitis							
	Resampling	Without resampling	PSO	Resampling GSA	FA	PSO	Without resampling GSA	FA
		Basic SVM				Parameter optimized SVM		
PAC	89.03	85.16	89.68	89.68	89.68	87.10	87.10	87.10
Sensitivity	73.91	65.52	80.00	80.00	80.00	73.08	73.08	73.08
Specificity	91.67	89.68	91.11	91.11	91.11	89.92	89.92	89.92
<i>F</i> -measure	88.60	84.89	88.97	88.97	88.97	86.58	86.58	86.58
Recall	89.03	85.16	89.68	89.68	89.68	87.10	87.10	87.10
Precision	88.46	84.69	89.10	89.10	89.10	86.44	86.44	86.44
		Basic MLP				Parameter optimized MLP		
PAC	90.32	81.94	92.26	92.26	92.26	87.10	85.16	83.23
Sensitivity	72.41	56.67	80.77	80.77	80.77	71.43	71.43	59.38
Specificity	94.44	88.00	94.57	94.57	94.57	90.55	87.31	89.43
<i>F</i> -measure	90.39	81.72	92.14	92.14	92.14	86.77	83.94	83.23
Recall	90.32	81.94	92.26	92.26	92.26	87.10	85.16	83.23
Precision	90.46	81.53	92.08	92.08	92.08	86.60	84.03	83.23

TABLE 23: Comparison of hybrid systems with HMV [14] for hepatitis dataset.

	Method	Accuracy	Specificity	Sensitivity	<i>F</i> -measure
Base paper	Ensemble	86.45	90.48	92.68	91.57
Without resampling	PMLP	87.1	90.6	71.4	86.8
Resampling	PMLP	92.3	80.8	94.6	92.1

TABLE 24: Performance of hybrid systems on thyroid dataset.

	Thyroid							
	Resampling	Without resampling	PSO	Resampling GSA	FA	PSO	Without resampling GSA	FA
		Basic SVM				Parameter optimized SVM		
PAC	89.77	89.77	91.16	91.16	91.16	89.77	89.77	89.77
Sensitivity	90.70	93.88	97.50	97.50	97.50	93.88	93.88	93.88
Specificity	89.53	88.55	89.71	89.71	89.71	88.55	88.55	88.55
<i>F</i> -measure	89.27	89.31	90.61	90.61	90.61	89.31	89.31	89.31
Recall	89.77	89.77	91.16	91.16	91.16	89.77	89.77	89.77
Precision	89.84	90.16	91.78	91.78	91.78	90.16	90.16	90.16
		Basic MLP				Parameter optimized MLP		
PAC	98.14	96.28	98.60	97.67	98.6	97.67	96.74	97.67
Sensitivity	98.18	93.85	98.21	94.83	98.21	95.45	93.94	95.45
Specificity	98.13	97.33	98.74	98.73	98.74	98.66	97.99	98.66
<i>F</i> -measure	98.13	96.28	98.60	97.68	98.6	97.68	96.75	97.68
Recall	98.14	96.28	98.60	97.67	98.6	97.67	96.74	97.67
Precision	98.14	96.28	98.60	97.69	98.6	97.69	96.76	97.69

TABLE 25: Comparison of hybrid systems with neural network [16] for thyroid dataset.

	Method	Accuracy	Sensitivity	Specificity	<i>F</i> -measure
Base paper	Neural networks	94.81	NIL	NIL	NIL
Without resampling	FMLP	97.7	95.5	98.7	97.7
Resampling	FMLP	98.6	98.2	98.7	98.6

TABLE 26: Performance of hybrid systems on Parkinson dataset.

Parkinson's disease								
Resampling	Without resampling		Resampling			Without resampling		
		PSO	GSA	FA	PSO	GSA	FA	
	Basic SVM				Parameter optimized SVM			
PAC	91.28	86.15	91.28	91.28	91.28	87.69	87.69	87.69
Sensitivity	90.00	87.04	90.00	90.00	90.00	86.39	86.39	86.39
Specificity	100.00	81.82	100.00	100.00	100.00	96.15	96.15	96.15
F-measure	90.41	85.21	90.41	90.41	90.41	86.29	86.29	86.29
Recall	91.28	86.15	91.28	91.28	91.28	87.69	87.69	87.69
Precision	92.15	85.75	92.15	92.15	92.15	88.79	88.79	88.79
	Basic MLP				Parameter optimized MLP			
PAC	96.41	91.28	96.92	96.92	96.41	92.31	92.31	93.85
Sensitivity	97.40	94.52	97.42	97.42	97.40	95.83	96.48	96.55
Specificity	92.68	81.63	95.00	95.00	92.68	82.35	81.13	86.00
F-measure	96.39	91.31	96.90	96.90	96.39	92.38	92.43	93.89
Recall	96.41	91.28	96.92	96.92	96.41	92.31	92.31	93.85
Precision	96.39	91.35	96.90	96.90	96.39	92.52	92.70	93.95

TABLE 27: Comparison of hybrid systems with HMV [14] for Parkinson dataset.

	Method	Accuracy	Sensitivity	Specificity	F-measure
Base paper	Ensemble	89.23	91.45	94.56	92.98
Without resampling	FMLP	93.8	96.6	86	93.9
Resampling	GMLP	96.9	97.4	95	96.9

TABLE 28: Performance of hybrid systems on Pima Indian diabetics' dataset.

Pima Indian diabetics								
Resampling	Without resampling		Resampling			Without resampling		
		PSO	GSA	FA	PSO	GSA	FA	
	Basic SVM				Parameter optimized SVM			
PAC	77.08	77.47	77.08	77.21	77.60	78.26	78.26	78.13
Sensitivity	72.78	72.51	72.53	72.93	73.63	74.63	74.63	74.51
Specificity	78.40	79.35	78.50	78.53	78.84	79.57	79.57	79.43
F-measure	75.86	76.74	75.90	76.01	76.45	77.45	77.45	77.30
Recall	77.08	77.47	77.08	77.21	77.60	78.26	78.26	78.13
Precision	76.51	76.97	76.49	76.65	77.09	77.85	77.85	77.71
	Basic MLP				Parameter optimized MLP			
PAC	79.30	75.13	77.47	80.60	80.99	74.48	77.60	76.04
Sensitivity	70.71	65.34	69.41	76.59	73.14	64.06	70.87	65.79
Specificity	83.18	79.88	80.69	82.06	84.60	79.69	80.48	81.47
F-measure	79.09	74.93	76.97	79.97	80.83	74.34	77.16	76.02
Recall	79.30	75.13	77.47	80.60	80.99	74.48	77.60	76.04
Precision	78.99	74.81	76.90	80.22	80.75	74.24	77.13	76.00

TABLE 29: Comparison of hybrid systems with HMV [14] for Pima Indian diabetics' dataset.

	Method	Accuracy	Sensitivity	Specificity	F-measure
Base paper	Ensemble	77.08	78.93	88.4	83.4
Without resampling	PSVM	78.3	74.6	79.6	77.5
Resampling	FMLP	81	73.1	84.6	80.8

TABLE 30: Performance of hybrid systems on BUPA liver disease dataset.

	BUPA liver disease								
	Resampling	Without resampling	PSO	Resampling GSA	FA	PSO	Without resampling GSA	FA	
		Basic SVM			Parameter optimized SVM				
PAC	63.48	70.14	66.09	66.67	63.48	70.72	70.14	70.14	
Sensitivity	61.44	68.10	64.86	65.97	61.44	72.00	68.10	68.10	
Specificity	65.10	71.18	67.01	67.16	65.10	70.20	71.18	71.18	
<i>F</i> -measure	63.40	69.50	65.95	66.47	63.40	69.52	69.50	69.50	
Recall	63.48	70.14	66.09	66.67	63.48	70.72	70.14	70.14	
Precision	63.39	69.89	66.01	66.61	63.39	70.96	69.89	69.89	
		Basic MLP			Parameter optimized MLP				
PAC	68.12	71.59	65.80	73.33	70.43	72.75	73.04	71.59	
Sensitivity	63.93	69.75	61.88	68.85	67.46	70.73	70.00	68.50	
Specificity	72.84	72.57	70.12	78.40	73.30	73.87	74.88	73.39	
<i>F</i> -measure	68.12	71.06	65.81	73.34	70.46	72.35	72.80	71.27	
Recall	68.12	71.59	65.80	73.33	70.43	72.75	73.04	71.59	
Precision	68.68	71.38	66.27	73.94	70.57	72.55	72.83	71.34	

TABLE 31: Comparison of hybrid systems with HMV [14] for BUPA liver disease dataset.

	Method	Accuracy	Sensitivity	Specificity	<i>F</i> -measure
Base paper	Ensemble	67.54	68.54	42.07	52.14
Without resampling	GMLP	73	70	74.9	72.8
Resampling	GMLP	73.3	68.9	78.4	73.3

with resampling are achieved by FMLP, GMLP, FMLP, FMLP, FMLP, and FMLP, respectively; best values found have been bolded for easy identification in Table 28. A comparison of Pima result with the state-of-the-art result is given in Table 29. The highest prediction accuracy for Pima is 78.3% (without resampling) and 81% (with resampling). The performance of all the considered techniques over Pima dataset with resampling is better than without resampling. Table 28 summarizes the performance of proposed hybrid alternatives for this dataset, and Table 29 compares this performance with best results obtained in recent literature.

7.3.11. BUPA Liver Disease Dataset. The performance of all the 8 techniques (2 machine learning and six hybrid systems) over BUPA dataset with and without resampling is given in Table 30. The highest accuracy, sensitivity, specificity, *F*-measure, recall, and precision without resampling are achieved by GMLP, PSVM, FMLP, FMLP, FMLP, and FMLP, respectively, with the values 73%, 72%, 74.9%, 72.8%, 73%, and 72.8%. The highest accuracy, sensitivity, specificity, *F*-measure, recall, and precision with resampling are achieved by GMLP, FMLP, GMLP, GMLP, GMLP, and GMLP, respectively, with the values 73.3%, 67.5%, 78.4%, 73.3%, 73.3%, and 73.9%. A comparison of BUPA result with the state-of-the-art result is given in Table 31. The highest prediction accuracy for BUPA is 73% (without resampling) and 73.3% (with resampling). The performance of all the

considered techniques over BUPA dataset with resampling is better than without resampling except in sensitivity. Table 30 summarizes the performance of proposed hybrid alternatives for this dataset, and Table 31 compares this performance with best results obtained in recent literature.

In GSA updating of an agent, the position is learned from all other agents, whereas in PSO updating of an agent position is based on two parameters called gBEST and pBEST. In each iteration of these two algorithms at most n , new solutions are brought forth. However, in FA in the worst case, each agent develops $O(n)$ new solution by moving towards all other best solutions. Therefore, in the worst case of FA, space is managed more efficiently than the other two algorithms (GSA and PSO). The same is demonstrated over the 11 medical datasets.

From the previous observations, it is concluded that MLP without resampling shows improvement in all datasets when compared with latest literature results and the same is depicted in Figure 2. As mentioned earlier, in Figure 2, the blue bar represents best performance in literature. Best results obtained by any of the six HISs proposed in this paper has been depicted in Figure 2 alongside, once without resampling (orange bar) and then with resampled data (gray bar). Sensitivity and specificity values for all systems have been presented in Tables 32 and 33, and it can be observed that our proposed hybrid system performs very well across all the datasets, in particular, the parameter optimized MLP. Table 34 summarizes

TABLE 32: Sensitivity improvement with hybrid systems.

S. number	Set	Base paper	Sensitivity		Resampling	
			Without resampling Technique	Percentage	Technique	Percentage
1	Cleveland	83.82	FMLP	87.5	FMLP	94.8
2	Statlog	86	PMLP	85.3	GMLP	92.2
3	Spect	27.27	FSVM	88.4	FMLP	91.9
4	Spectf	7.27	FSVM	88	GSVM	94.7
5	Eric	86.32	GMLP	88.4	GMLP	88.8
6	WBC	98.01	PSVM	95.1	FMLP	96.6
7	Hepatitis	90.48	FSVM	73.1	FMLP	80.8
8	Thyroid	NIL	GMLP	95.5	GMLP	98.2
9	Parkinson	91.45	FMLP	96.6	FMLP	97.4
10	Pima Indian diabetics	78.93	PSVM	74.6	FMLP	76.6
11	BUPA	68.54	PSVM	72	FMLP	67.5

TABLE 33: Specificity improvement with hybrid systems.

S. number	Dataset	Base paper	Specificity		Resampling	
			Without resampling Technique	Percentage	Technique	Percentage
1	Cleveland	88.41	PMLP	84.8	FMLP	93.5
2	Statlog	86	FMLP	87.8	FMLP	90.7
3	Spect	96.23	FMLP	74.2	GMLP	77.3
4	Spectf	96.7	FMLP	83.3	PMLP	77.6
5	Eric	73.91	FMLP	79.1	GMLP	93.7
6	WBC	96.94	FMLP	98	FMLP	98.7
7	Hepatitis	92.68	PMLP	90.6	FMLP	94.6
8	Thyroid	NIL	GMLP	98.7	FMLP	98.7
9	Parkinson	94.56	FSVM	96.2	FSVM	100
10	Pima Indian diabetics	88.4	FMLP	81.5	GMLP	84.6
11	BUPA	42.07	GMLP	74.9	GMLP	78.4

TABLE 34: Parameters to be used in MLP for all datasets.

Dataset	Learning rate	Momentum	Hybrid MLP accuracy	Base paper accuracy
Cleveland	0.4410246832765716	0.945131728055943	85.8	85
Statlog	0.24687115044065697	0.7512112614957723	85.9	84.4
Spect	0.001620407197768992	0.5458467309532906	85	82.02
Spectf	0.0037254964036241137	0.6064034495456784	82.4	78.28
Eric	0.6953073769599724	0.9167657941184544	81.3	80.86
Breast cancer	0.5272364248697747	0.9288899224295802	96.99	96.71
Hepatitis	0.6376255545427609	0.9250563419048221	87.1	86.45
Thyroid	0.1516498076389815	0.48805304429332785	97.7	94.81
Parkinson	0.8486064853474067	0.3499016503919223	93.8	89.23
Pima Indian diabetics	0.03218577681226653	0.06466339445401592	77.60	77.08
BUPA	0.8329619224653821	0.014749643317800043	73	67.54

the optimal parameter values of MLP. Hence, in comparison with the ensemble techniques, parameter optimized MLP gives a better result.

Table 3 gives the outcomes of the rank test for the results of with resampling at the level of significance (LOS) 0.01. If h value is zero, then H_0 is true otherwise H_1 is true. In

Table 3, 161 ones are there out of a total of 165. It means that 161 times null hypothesis is false and four times null hypothesis is true.

Table 4 gives the outcomes of the rank test for the results of with resampling at the level of significance (LOS) 0.05. In Table 4, 163 ones are there out of 165. It means that 163 times null-hypothesis is false and four times null hypothesis is true.

Table 5 gives the outcomes of the rank test for the results of without resampling at the level of significance (LOS) 0.01. In Table 5, 164 ones are there out of 165. It means that 164 times null hypothesis is false and one time null hypothesis is true.

Table 6 gives the outcomes of the rank test for the results of without resampling at the level of significance (LOS) 0.05. In Table 6, 164 ones are there out of 165. It means that 164 times null hypothesis is false and one-time null hypothesis is true.

The outcomes of FMLP are taken as “ m ” value. Tables 7 and 8 give the results of t -test for both resampling and without resampling techniques with LOS 0.01 and 0.05. In these tables, h value is zero for all datasets at 0.01 and 0.05 LOS. Hence, null hypothesis is accepted for all datasets.

8. Conclusion

Due to the complex framework of ensemble approach and the moderate performance of the individual classifier, hybrid systems have a lot of promise in the diagnosis and prognosis of diseases. To overcome these, we proposed a disease diagnosis system by juxtaposing three evolutionary algorithms and SVM and MLP classifiers. Three evolutionary algorithms optimize the parameters of the two classifiers and such enhanced classifiers have been used to train and diagnose diseases. Accordingly, six hybrid diagnosis alternatives have been obtained by working out the combinations of classifiers and evolutionary algorithms. Based on results presented in this paper, it can be concluded that our hybridization approach provides high prediction accuracy than other methods in literature across a wide variety of disease datasets. Even among the six alternative parameter optimized classifier systems proposed, FMLP was found to be the relatively best across the majority of the 11 datasets considered. On average, MLP shows 2.2% and 6.814% improvement in prediction accuracy on the 11 datasets with and without resampling. The ranges of improvements shown by MLP in the objective sensitivity are -2.9 to 75.13 and -9.68 to 86.33 without and with resampling, respectively. The ranges of improvement shown by MLP in the objective specificity are -9.68 to 86.33 and -18.93 to 36.33 without and with resampling, respectively. From the experimental results, it is concluded that FMLP shows outperformance than recently developed ensemble classifiers ([14, 15]). As a part of the continuation of this research, we intend to process a very higher dimensional dataset with the major phases of feature selection and parameter evolution of the classifier. For feature selection, similarity metric-based hypergraph will be constructed and then by using hypergraph special properties, important topological and geometrical features will be identified. In phase 2, competitive and co-operative parallel hybrid intelligent systems will be employed for incorporating direct and indirect

communication among the different systems at guaranteed run times that would allow the entire HISs to converge to a single value. This work is presently ongoing.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

Two of the authors (MadhuSudana Rao Nalluri and Kannan K.) of this paper wish to thank the Department of Science and Technology, Government of India, for the financial sanction towards this work under FIST programme: SR/FST/MSI-107/2015.

References

- [1] S.-H. Liao, P.-H. Chu, and P.-Y. Hsiao, “Data mining techniques and applications—a decade review from 2000 to 2011,” *Expert Systems with Applications*, vol. 39, no. 12, pp. 11303–11311, 2012.
- [2] E. W. T. Ngai, L. Xiu, and D. C. K. Chau, “Application of data mining techniques in customer relationship management: a literature review and classification,” *Expert Systems with Applications*, vol. 36, no. 2, pp. 2592–2602, 2009.
- [3] E. W. Ngai, Y. Hu, Y. H. Wong, Y. Chen, and X. Sun, “The application of data mining techniques in financial fraud detection: a classification framework and an academic review of literature,” *Decision Support Systems*, vol. 50, no. 3, pp. 559–569, 2011.
- [4] N. Esfandiari, M. R. Babavalian, A. M. Moghadam, and V. K. Tabar, “Knowledge discovery in medicine: current issue and future trend,” *Expert Systems with Applications*, vol. 41, no. 9, pp. 4434–4463, 2014.
- [5] Y. Li, C. Bai, and C. K. Reddy, “A distributed ensemble approach for mining healthcare data under privacy constraints,” *Information Sciences*, vol. 330, pp. 245–259, 2016.
- [6] R. Ramos-Pollán, M. Á. Guevara-López, and E. Oliveira, “A software framework for building biomedical machine learning classifiers through grid computing resources,” *Journal of Medical Systems*, vol. 36, no. 4, pp. 2245–2257, 2012.
- [7] A. Malik and J. Iqbal, “Extreme learning machine based approach for diagnosis and analysis of breast cancer,” *Journal of the Chinese Institute of Engineers*, vol. 39, no. 1, pp. 74–78, 2016.
- [8] S. Palaniappan and R. Awang, “Intelligent heart disease prediction system using data mining techniques,” in *2008 IEEE/ACS International Conference on Computer Systems and Applications*, pp. 108–115, Doha, 2008.
- [9] M. Hariharan, K. Polat, and R. Sindhu, “A new hybrid intelligent system for accurate detection of Parkinson’s disease,” *Computer Methods and Programs in Biomedicine*, vol. 113, no. 3, pp. 904–913, 2014.
- [10] I. Castelli and E. Trentin, “Combination of supervised and unsupervised learning for training the activation functions of neural networks,” *Pattern Recognition Letters*, vol. 37, pp. 178–191, 2014.
- [11] B. Xie, Y. Liu, H. Zhang, and J. Yu, “A novel supervised approach to learning efficient kernel descriptors for high

- accuracy object recognition,” *Neurocomputing*, vol. 182, pp. 94–101, 2016.
- [12] K. Morris and P. D. McNicholas, “Clustering, classification, discriminant analysis, and dimension reduction via generalized hyperbolic mixtures,” *Computational Statistics & Data Analysis*, vol. 97, pp. 133–150, 2016.
- [13] V. Elyasigomari, M. S. Mirjafari, H. R. Screen, and M. H. Shaheed, “Cancer classification using a novel gene selection approach by means of shuffling based on data clustering with optimization,” *Applied Soft Computing*, vol. 35, pp. 43–51, 2015.
- [14] S. Bashir, U. Qamar, F. H. Khan, and L. Naseem, “HMV: a medical decision support framework using multi-layer classifiers for disease prediction,” *Journal of Computational Science*, vol. 13, pp. 10–25, 2016.
- [15] S. Bashir, U. Qamar, and F. H. Khan, “BagMOOV: a novel ensemble for heart disease prediction bootstrap aggregation with multi-objective optimized voting,” *Australasian Physical & Engineering Sciences in Medicine*, vol. 38, no. 2, pp. 305–323, 2015.
- [16] F. Temurtas, “A comparative study on thyroid disease diagnosis using neural networks,” *Expert Systems with Applications*, vol. 36, no. 1, pp. 944–949, 2009.
- [17] R. Das, I. Turkoglu, and A. Sengur, “Effective diagnosis of heart disease through neural networks ensembles,” *Expert Systems with Applications*, vol. 36, no. 4, pp. 7675–7680, 2009.
- [18] R. Chitra and V. Seenivasagam, “Heart disease prediction system using supervised learning classifier,” *Bonfring International Journal of Software Engineering and Soft Computing*, vol. 3, no. 1, p. 1, 2013.
- [19] S. A. Pattekari and A. Parveen, “Prediction system for heart disease using Naïve Bayes,” *International Journal of Advanced Computer and Mathematical Sciences*, vol. 3, no. 3, pp. 290–294, 2012.
- [20] M. A. Jabbar, B. L. Deekshatulu, and P. Chandra, “Heart disease prediction system using associative classification and genetic algorithm,” 2012, <http://arxiv.org/abs/1303.5919>.
- [21] H. D. Masethe and M. A. Masethe, “Prediction of heart disease using classification algorithms,” in *Proceedings of the World Congress on Engineering and Computer Science*, vol. 2, pp. 22–24, San Francisco, USA, October 2014.
- [22] A. Shaikh, N. Mahoto, F. Khuhawar, and M. Memon, “Performance evaluation of classification methods for heart disease dataset,” *Sindh University Research Journal-SURJ (Science Series)*, vol. 47, no. 3, 2015.
- [23] R. Kavitha and T. Christopher, “An effective classification of heart rate data using PSO-FCM clustering and enhanced support vector machine,” *Indian Journal of Science and Technology*, vol. 8, no. 30, 2015.
- [24] R. Alizadehsani, J. Habibi, M. J. Hosseini et al., “A data mining approach for diagnosis of coronary artery disease,” *Computer Methods and Programs in Biomedicine*, vol. 111, no. 1, pp. 52–61, 2013.
- [25] A. Shenfield and S. Rostami, “A multi objective approach to evolving artificial neural networks for coronary heart disease classification,” in *2015 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pp. 1–8, Niagara Falls, ON, 2015.
- [26] N. Bhatla and K. Jyoti, “An analysis of heart disease prediction using different data mining techniques,” *International Journal of Engineering Research and Technology*, vol. 1, no. 8, pp. 1–4, 2012.
- [27] L. Parthiban and R. Subramanian, “Intelligent heart disease prediction system using CANFIS and genetic algorithm,” *International Journal of Biological, Biomedical and Medical Sciences*, vol. 3, no. 3, 2008.
- [28] N. G. Hedeshi and M. S. Abadeh, “Coronary artery disease detection using a fuzzy-boosting PSO approach,” *Computational Intelligence and Neuroscience*, vol. 2014, Article ID 783734, 12 pages, 2014.
- [29] E. O. Olaniyi, O. K. Oyedotun, and K. Adnan, “Heart diseases diagnosis using neural networks arbitration,” *International Journal of Intelligent Systems and Applications (IJISA)*, vol. 7, no. 12, p. 75, 2015.
- [30] J. K. Kim, J. S. Lee, D. K. Park, Y. S. Lim, Y. H. Lee, and E. Y. Jung, “Adaptive mining prediction model for content recommendation to coronary heart disease patients,” *Cluster Computing*, vol. 17, no. 3, pp. 881–891, 2014.
- [31] V. Chauraisa and S. Pal, “Early prediction of heart diseases using data mining techniques,” *Caribbean Journal of Science and Technology*, vol. 1, pp. 208–217, 2013.
- [32] H. Yan, Y. Jiang, J. Zheng, C. Peng, and Q. Li, “A multilayer perceptron-based medical decision support system for heart disease diagnosis,” *Expert Systems with Applications*, vol. 30, no. 2, pp. 272–281, 2006.
- [33] H. Yan, J. Zheng, Y. Jiang, C. Peng, and Q. Li, “Development of a decision support system for heart disease diagnosis using multilayer perceptron,” in *Proceedings of the 2003 International Symposium on Circuits and Systems (ISCAS '03)*, vol. 5, pp. V-709–V-712, 2003.
- [34] M. Shouman, T. Turner, and R. Stocker, “Using data mining techniques in heart disease diagnosis and treatment,” in *2012 Japan-Egypt Conference on Electronics, Communications and Computers*, pp. 173–177, Alexandria, 2012.
- [35] M. Karaolis, J. A. Moutiris, and C. S. Pattichis, “Assessment of the risk of coronary heart event based on data mining,” in *2008 8th IEEE International Conference on BioInformatics and BioEngineering*, pp. 1–5, Athens, 2008.
- [36] C. Ordóñez, E. Omiecinski, L. De Braal et al., “Mining constrained association rules to predict heart disease,” in *Proceedings 2001 IEEE International Conference on Data Mining*, pp. 433–440, San Jose, CA, USA, 2001.
- [37] R. Das, I. Turkoglu, and A. Sengur, “Diagnosis of valvular heart disease through neural networks ensembles,” *Computer Methods and Programs in Biomedicine*, vol. 93, no. 2, pp. 185–191, 2009.
- [38] A. Taneja, “Heart disease prediction system using data mining techniques,” *Oriental Journal of Computer Science and Technology*, vol. 6, no. 4, pp. 457–466, 2013.
- [39] J. S. Sartakhti, M. H. Zangoeei, and K. Mozafari, “Hepatitis disease diagnosis using a novel hybrid method based on support vector machine and simulated annealing (SVM-SA),” *Computer Methods and Programs in Biomedicine*, vol. 108, no. 2, pp. 570–579, 2012, ISSN 0169-2607.
- [40] D. Çalişır and E. Dogantekin, “A new intelligent hepatitis diagnosis system: PCA–LSSVM,” *Expert Systems with Applications*, vol. 38, no. 8, pp. 10705–10708, 2011.
- [41] J. Li and L. Wong, “Using rules to analyse bio-medical data: a comparison between C4.5 and PCL,” in *Advances in Web-Age Information Management*, pp. 254–265, Springer, Berlin Heidelberg, 2003.

- [42] C.-H. Weng, T. C.-K. Huang, and R.-P. Han, "Disease prediction with different types of neural network classifiers," *Telematics and Informatics*, vol. 33, no. 2, pp. 277–292, 2016.
- [43] Y. N. Jane, H. K. Nehemiah, and K. Arputharaj, "A Q-backpropagated time delay neural network for diagnosing severity of gait disturbances in Parkinson's disease," *Journal of Biomedical Informatics*, vol. 60, pp. 169–176, 2016.
- [44] H. Gürüler, "A novel diagnosis system for Parkinson's disease using complex-valued artificial neural network with k-means clustering feature weighting method," in *Neural Computing and Applications*, pp. 1–10, Springer, London, 2016.
- [45] S. Bashir, U. Qamar, and F. H. Khan, "IntelliHealth: a medical decision support application using a novel weighted multi-layer classifier ensemble framework," *Journal of Biomedical Informatics*, vol. 59, pp. 185–200, 2016.
- [46] A. Iyer, S. Jeyalatha, and R. Sumbaly, "Diagnosis of diabetes using classification mining techniques," 2015, <http://arxiv.org/abs/1502.03774>.
- [47] D. K. Choubey and P. Sanchita, "GA_MLP NN: a hybrid intelligent system for diabetes disease diagnosis," *International Journal of Intelligent Systems and Applications*, vol. 8, no. 1, p. 49, 2016.
- [48] S. Kharya, "Using data mining techniques for diagnosis and prognosis of cancer disease," 2012, <http://arxiv.org/abs/1205.1923>.
- [49] V. Chaurasia and S. Pal, "A novel approach for breast cancer detection using data mining techniques," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, no. 1, pp. 2456–2465, 2014.
- [50] R. Fernandez-Millan, J. A. Medina-Merodio, R. B. Plata, J. J. Martinez-Herraiz, and J. M. Gutierrez-Martinez, "A laboratory test expert system for clinical diagnosis support in primary health care," *Applied Sciences*, vol. 5, no. 3, pp. 222–240, 2015.
- [51] A. Alzubaidi, G. Cosma, D. Brown, and A. G. Pockley, "A new hybrid global optimization approach for selecting clinical and biological features that are relevant to the effective diagnosis of ovarian cancer," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–8, Athens, December 2016.
- [52] J. Gwak, M. Jeon, and W. Pedrycz, "Bolstering efficient SSGAs based on an ensemble of probabilistic variable-wise crossover strategies," *Soft Computing*, vol. 20, no. 6, pp. 2149–2176, 2016.
- [53] S. L. Hsieh, S. H. Hsieh, P. H. Cheng et al., "Design ensemble machine learning model for breast cancer diagnosis," *Journal of Medical Systems*, vol. 36, no. 5, pp. 2841–2847, 2012.
- [54] L. Shen, H. Chen, Z. Yu et al., "Evolving support vector machines using fruit fly optimization for medical data classification," *Knowledge-Based Systems*, vol. 96, pp. 61–75, 2016.
- [55] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.
- [56] C. Phua, V. Lee, K. Smith, and R. Gayler, "A comprehensive survey of data mining-based fraud detection research," 2010, <http://arxiv.org/abs/1009.6119>.
- [57] A. A. Freitas, "A critical review of multi-objective optimization in data mining: a position paper," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 2, pp. 77–86, 2004.
- [58] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, 2001.
- [59] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: a gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.
- [60] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "Filter modeling using gravitational search algorithm," *Engineering Applications of Artificial Intelligence*, vol. 24, no. 1, pp. 117–122, 2011.
- [61] C. Li and J. Zhou, "Parameters identification of hydraulic turbine governing system using improved gravitational search algorithm," *Energy Conversion and Management*, vol. 52, no. 1, pp. 374–381, 2011.
- [62] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, pp. 1942–1948, Perth, WA, 1995.
- [63] C.-L. Huang and J.-F. Dun, "A distributed PSO-SVM hybrid system with feature selection and parameter optimization," *Applied Soft Computing*, vol. 8, no. 4, pp. 1381–1391, 2008.
- [64] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *International Journal of bio-Inspired Computation*, vol. 2, no. 2, pp. 78–84, 2010.
- [65] F. Rosenblatt, "Principles of neurodynamics. Perceptrons and the theory of brain mechanisms," Cornell Aeronautical Lab Inc., Buffalo, NY, 1961.
- [66] S. K. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets, and classification," *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 683–697, 1992.
- [67] R. Collobert and S. Bengio, "Links between perceptrons, MLPs and SVMs," in *Proceedings of the Twenty-First International Conference on Machine Learning (ICML '04)*, Banff, Alberta, Canada, 2004.
- [68] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [69] S. W. Lin, K. C. Ying, S. C. Chen, and Z. J. Lee, "Particle swarm optimization for parameter determination and feature selection of support vector machines," *Expert Systems with Applications*, vol. 35, no. 4, pp. 1817–1824, 2008.
- [70] S. Chatterjee, S. Sarkar, S. Hore, N. Dey, A. S. Ashour, and V. E. Balas, "Particle swarm optimization trained neural network for structural failure prediction of multistoried RC buildings," in *Neural Computing and Applications*, pp. 1–12, Springer, London, 2016.
- [71] <http://archive.ics.uci.edu/ml/datasets.html>.
- [72] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.
- [73] W. L. Martinez and A. R. Martinez, *Computational Statistics Handbook with MATLAB*, vol. 22, CRC press, 2007.
- [74] D. A. Van Veldhuizen, "Multiobjective evolutionary algorithms: classifications, analyses, and new innovations," Air Force Institute of Technology Wright-Patterson AFB OH School of Engineering, 1999.
- [75] J. R. Schott, "Fault tolerant design using single and multicriteria genetic algorithm optimization," Air Force Institute of Technology Wright-Patterson AFB OH, 1995.

Research Article

Diagnosis of Alzheimer's Disease Based on Structural MRI Images Using a Regularized Extreme Learning Machine and PCA Features

Ramesh Kumar Lama,^{1,2} Jeonghwan Gwak,^{1,3} Jeong-Seon Park,⁴ and Sang-Woong Lee^{1,2}

¹National Research Center for Dementia, Gwangju, Republic of Korea

²Department of Software, Gachon University, 1342 Seongnamdaero, Sujeonggu, Seongnam, Gyeonggido 13120, Republic of Korea

³School of Electrical Engineering and Computer Science (EECS), Gwangju Institute of Science and Technology (GIST), Gwangju 61005, Republic of Korea

⁴Department of Multimedia, Chonnam National University, 50 Daehakro, Yeosu, Jeollanamdo 59626, Republic of Korea

Correspondence should be addressed to Sang-Woong Lee; slee@gachon.ac.kr

Received 4 December 2016; Revised 30 March 2017; Accepted 13 April 2017; Published 18 June 2017

Academic Editor: Ashish Khare

Copyright © 2017 Ramesh Kumar Lama et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Alzheimer's disease (AD) is a progressive, neurodegenerative brain disorder that attacks neurotransmitters, brain cells, and nerves, affecting brain functions, memory, and behaviors and then finally causing dementia on elderly people. Despite its significance, there is currently no cure for it. However, there are medicines available on prescription that can help delay the progress of the condition. Thus, early diagnosis of AD is essential for patient care and relevant researches. Major challenges in proper diagnosis of AD using existing classification schemes are the availability of a smaller number of training samples and the larger number of possible feature representations. In this paper, we present and compare AD diagnosis approaches using structural magnetic resonance (sMR) images to discriminate AD, mild cognitive impairment (MCI), and healthy control (HC) subjects using a support vector machine (SVM), an import vector machine (IVM), and a regularized extreme learning machine (RELM). The greedy score-based feature selection technique is employed to select important feature vectors. In addition, a kernel-based discriminative approach is adopted to deal with complex data distributions. We compare the performance of these classifiers for volumetric sMR image data from Alzheimer's disease neuroimaging initiative (ADNI) datasets. Experiments on the ADNI datasets showed that RELM with the feature selection approach can significantly improve classification accuracy of AD from MCI and HC subjects.

1. Introduction

Alzheimer's disease (AD) is a slow fatal neurodegenerative disease affecting people over the age of 65 years [1], while early-onset AD is also diagnosed before 65. The deposition of two abnormal protein fragments known as plaques and tangles in the brain causes the death of neuron cells. The hippocampus, where the memories are first formed, is the initially affected region by AD, and thus early symptoms of AD include memory problems resulting difficulties in word finding and thinking processes [2]. AD patients suffer from a lack of initiative, changes in personality or behavior in day-to-day functions at home, or at work, and in taking care of oneself, eventually, leading to death. The brain volume

reduces dramatically through time and affects most of its functions with the progression of AD.

With the increase in the population of elderly people in developed countries, AD is going to be a major problem in socioeconomic implications. According to the recent report [3], it is expected that the number of affected people will be doubled in the next 20 years and one in two aged above 85 years will suffer from AD by 2050. Thus, accurate diagnosis of AD is very important, especially, at its early stage. Conventionally, the diagnosis of AD is performed by a neuropsychological examination in support of structural imaging. It is reported in [4] that (1) in the early stage of AD, degeneration of neurons takes place in the medial temporal lobe, (2) gradually affecting the entorhinal cortex, the hippocampus, and the

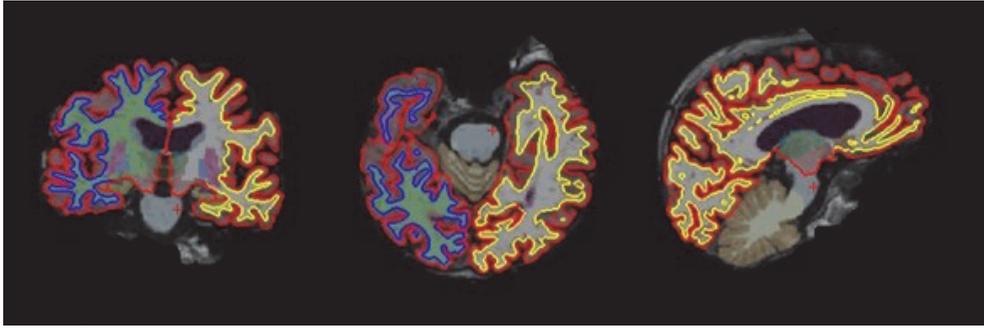


FIGURE 1: Segmentation of brain MR images for volumetric study.

limbic system, and (3) neocortical areas are affected at the final stage. Therefore, the study of medial temporal lobe atrophy (MTA), particularly in the hippocampus, the entorhinal cortex, and the amygdala provides the evidence of the progression of AD. Generally, MTA is measured in terms of voxel-based [5], vertex-based [6], and ROI-based [7] approaches. However, as the disease progresses, other regions in the brain are also affected. In such cases, whole-brain methods are preferred rather than a specific region-based method; then, the characterization of brain atrophy for differentiating AD and MCI patients can be performed more efficiently.

In recent years, major advances in neuroimaging have provided opportunities to study neurological-related diseases, resulting improvements in early and accurate detection of AD [5, 6, 8]. Magnetic resonance imaging (MRI) is more widely used in AD-related studies because of its noninvasive nature and lack of pain to patients. In addition, MRI provides an excellent spatial resolution and good contrast [5–7, 9]. Thus, several studies have used structural MRI- (sMRI-) based biomarkers to classify AD [10–19], which describes brain atrophy and change in the size of brain tissues. Similarly, functional MRI (fMRI) [20] can be utilized to characterize the hemodynamic response relevant to neural activity and functional/structural connectivity [21–23], which can be used to describe neurological disorders in the whole brain at the connectivity level. In this paper, we focused only on AD classification using sMRI. The intensity and stage of the neurodegeneration can be identified by the help of atrophy measured by sMRI [24]. Thus, sMRI-based feature extraction has attracted the attention for researchers of AD classification. These studies include morphometric methods such as region of interest (ROI)/volume of interest (VOI) grey matter voxels in the automatic segmentation of images [25] and the sMRI measurement of the hippocampus and the medial temporal lobe [26].

Several machine learning techniques have been used to distinguish AD subjects from elderly control subjects using different biomarkers. The commonly used classifiers include support vector machine (SVM), artificial neural network (ANN), and other ensemble classifiers. Among them, SVM and the variants have been widely studied due to its relatively good accuracy and ability to deal with high-dimensional data. A SVM-type classifier (e.g., Magnin et al. [27]) begins with a learning stage from the training dataset consisting of well-characterized subjects with known states (i.e., labels for

the subjects are given). Then, the classifier aims to maximize the margin of the training data by constructing the optimal separating hyperplane or a set of hyperplanes in a single- or higher-dimensional space. At a testing stage, classification is performed for test dataset based on the learned hyperplane(s). In general, three-dimensional (3D) T1-weighted MR images of each subject were automatically parcellated into ROIs. Grey matter from each ROI is extracted, as shown in Figure 1, as a feature for classification.

Zhang et al. [10] proposed a multimodal classification approach by utilizing multiple-kernel SVM based on the biomarkers including sMRI [18, 19], positron emission tomography (PET) [6], and cerebrospinal fluid (CSF) [28] to discriminate AD (or MCI) and normal control (NC) subjects. From the binary classification (i.e., AD versus NC and MCI versus NC) results, their proposed model could obtain a good accuracy for AD classification and an encouraging accuracy for MCI classification. Liu et al. [29, 30] proposed deep learning-based multiclass classification among normal controls (NC), MCI nonconverters (ncMCI), MCI converters (cMCI), and AD subjects based on 83 ROIs of sMRI images and the corresponding registered PET images. Stacked auto-encoders (SAE) were used as unsupervised learning to obtain high-level features, and then softmax logistic regression was adopted as the classifier. While the experimental results showed reasonably good performance, it is still arguing that the denoising nature of SAE can increase the difficulty of suitable feature learning and thus it may be difficult for practical use. Li et al. [31] proposed fine-grained new features based on principle component analysis (PCA), stability selection, dropout, and multitask learning, where restricted Boltzmann machine (RBM) model was used as the deep learning architecture. 93 ROIs of MRI and PET images, together with CSF biomarkers, are used. Ye et al. [32] introduced conceptual machine learning-based multimodal data fusion approach using MRI, PET, genetic, CSF, demographic for AD-related research, and functional connectivity analysis. Recently, Rama et al. [33] proposed IVM-based classification approach for multiclass classification. In this method, only the subset of features from structural MRI was used as input to kernel logistic regression thus reducing the computational cost. This method used total 65 ROIs as features for training and testing and achieved the accuracy of up to 70% while classifying AD, MCI, and HC and 76.9% for binary classification of HC and AD [33].

TABLE 1: Summary of subject's demographic status.

	NC	MCI	AD
Number of subjects	70	74	70
Average age	76.3	74.5	76.0
Average education points	16.19	15.96	15.53
MMSE	29.2 ± 1.0	27.2 ± 1.7	23.2 ± 2.0

While several approaches have been proposed for classification of different AD stages, with relatively small dataset, it is very difficult to extract effective information. This work focuses on comparing and presenting efficient classification approaches working robustly for a relatively small dataset. To this end, we present and compare three representative classifiers, with an efficient feature selection approach, including SVM, an import vector machine (IVM) and a regularized extreme learning machine (RELM) for the multiclass classification of different stages of AD progression.

2. Materials for Study

2.1. sMRI Dataset. Data used in preparation of this paper were obtained from the Alzheimer's disease neuroimaging initiative database (ADNI) (<http://adni.loni.usc.edu/>). The ADNI database was launched in 2003 as a public-private partnership. The primary goal of ADNI has been to test whether the serial MRI, PET, other biological markers, and clinical and neurophysical assessment can be combined to measure the progression of midcognitive impairment and the early AD.

2.2. Subjects. The ADNI dataset consists of more than 6000 subjects aged from 18 to 96. From it, we selected 214 subjects aged between 65 and 96. The selected participants met the criteria defined in the ADNI protocol. We constructed balanced dataset consisting of 214 subjects as follows:

- (1) 70 NC subjects: 33 males, 37 females; age \pm SD = 76.3 \pm 5.4 years, range = 60–90 years; mini-mental state estimation (MMSE) score = 29.2 \pm 1.0, range = 25–30.
- (2) 74 MCI subjects who had not converted to AD within 18 months: 38 males, 36 females; age \pm SD = 74.5 \pm 7.2 years, range = 58–88 years; MMSE score = 27.2 \pm 1.7, range = 24–30.
- (3) 70 AD subjects: 39 males, 31 females; age \pm SD = 76.0 \pm 7.3 years, range = 55–91 years; MMSE = 23.2 \pm 2.0, range = 18–27.

Table 1 shows a summary of demographic status of the selected subjects.

All structural MR (sMR) scans used in this work were acquired from 3T scanners. The main focus of this work was to elaborate the supervised multiclass classification among NC, MCI, and AD based on different classifiers. Thus, to obtain unbiased estimates of the classifier performance, the selected subjects were randomly split up into two groups of the training dataset and the testing dataset. The algorithms were trained on a training set, and the performances of the

diagnostic sensitivity and specificity together with accuracy were evaluated on an independent test dataset. The division process considers balanced age and sex distributions.

2.3. Preprocessing of sMRI Data. We used a fully automated pipeline of the FreeSurfer 5.3.0 software package for reconstruction and volumetric segmentation from all the sMRI images and extracted the pattern of useful data. The software performs a series of preprocessing operations with the FreeSurfer's recon-all processing pipeline on the original sMRI data as shown in Figure 2. The preprocessing steps include motion correction, T1-weighted image averaging, registration of volume to the Talairach space, skull stripping with a deformable template model. The white surface and the pial surface are generated for each hemisphere using encoding the shape of the corpus callosum and pons in the Talairach space and following the intensity gradients from the white matter. The accurate matching of the morphologically homologous cortical locations across subjects was estimated using the mapping of the atlas based on a cortical surface to a sphere aligning the cortical patterns. Cortical thickness at each vertex of the cortex is denoted by the average shortest distance between white and pial surfaces. The area of every triangle in a standardized spherical surface tessellation provides the surface area. Similarly, the registration surface based on the folding pattern was used to compute the local curvature. The method developed by Schaer [34] was used to measure the folding index over the whole cortical surface. All the extracted features are explained in terms of feature measures as in Table 2.

2.4. Details of the sMRI Data. We perform binary classification for NC versus AD and multiclass classification using the one-versus-all (OVA) class setting for NC, MCI, and AD. For the subjects and groups chosen as in Table 2, volumetric features, fM5, in Table 2, were used for the study, and for each feature, we computed the grey matter tissue volume from the individual subject's sMRI image. Block brain regions selected for the classification are shown in Figure 3. Each tissue is discriminated from other tissues by using color code defined by FreeSurfer software package. The left column presents the coronal view followed by the saggital view in the middle column and the axial view at the rightmost column. We followed neurological convention for the view. All sMR scans used in this paper were acquired from 3T scanners.

3. Proposed Methods: Classification of Stages of AD Progression

We used the three representative machine-learning classification algorithms, SVM, IVM, and RELM. The stepwise block diagram of the classification of stages of AD progression is shown in Figure 4.

3.1. Efficient Feature Selection. In neuroimaging analysis, the number of features per subject can be very high compared to the number of subjects, which is commonly referred to as the curse of dimensionality. We perform an efficient feature selection method based on PCA which is a method widely used to reduce the dimensionality of a high-

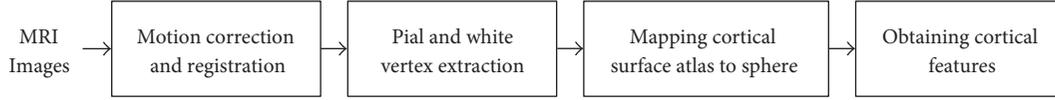


FIGURE 2: Preprocessing steps of sMRI images.

TABLE 2: Feature measures and cortical feature index information.

Feature measure (fM)	Feature measure type	Indices of cortical feature
fM1	Mean cortical thickness	1–64
fM2	Surface area	65–128
fM3	Folding indices	193–256
fM4	Mean curvature indices	193–256
fM5	Volume	257–320

dimensional (imaging) data [25]. As the result, the most information representative dimensions are kept while the least important ones are excluded. PCA generates new features which are a linear combination of the initial features and maps each instance of the given dataset present in a d -dimensional space to a k -dimensional subspace such that $k < d$. The set of k new dimensions generated are called the principal components (PCs), and each PC is directed towards maximum variance excluding the variance already accounted for in all its preceding components. Subsequently, the first component covers the maximum variance, and each component that follows it covers a lesser value of variance. PCs can be represented as

$$PC_i = a_1X_1 + a_2X_2 + \dots + a_dX_d, \quad (1)$$

where PC_i is the i th PC, X_j is the original feature j , and a_j represents the numerical coefficient for X_j .

3.2. SVM Classifier. SVM [35] is basically a binary classifier which is useful for the classification of both separable and nonseparable data. It has been used in the neuroimaging field and considered as one of the most popular machine learning tools in the neuroscience domain in the last decade. It is a supervised classification algorithm and finds the optimal hyperplane that separates both classes with maximum margin from support vectors during the training phase. For the testing of new data points, the classifier's decision is based on the estimated hyperplane. For the linearly separable patterns, linear SVM is used. However, linear SVM cannot guarantee better performance in complex cases with nonseparable patterns. In such scenario, linear SVM is extended using kernel trick. The input patterns are mapped into a higher dimensional space using linear and nonlinear functions known as kernels. Linear and nonlinear radial basis function (RBF) kernels are widely used SVM kernels.

3.3. IVM Classifier. The fundamental principle of IVM proposed by Zhu and Hastie [36] is built on kernel logistic

regression (KLR). It has not merely performed well in the binary classification as SVM, and it can be naturally generalized to the multiclass classification. Thus, we begin with the explanation of logistic regression. Let $x_i = (x_1, \dots, x_n)^T$ represent observed samples with class labels $y_j \in C \{j = 1, \dots, K\}$ pattern classes. The training set is represented as $(x_i, y_j), i = 1, \dots, n$. For the binary class problem, where input samples x_i are independent and identically distributed, the conditional class posterior probability $P_i(y_i/x_i; w)$ is estimated using the following logistic regression model:

$$P_i\left(\frac{y_i}{x_i; w}\right) = \frac{1}{1 + \exp(w^T x_i)}. \quad (2)$$

The logistic regression predicts the class based on probabilities which are either p for $y_i = 1$ or $1 - p$ for $y_i = 0$. Thus, we can express the cost function of logistic regression as

$$Q_0(w) = \prod_{i=1}^n p(x_i)^{y_i} (1 - p(x_i))^{1-y_i}. \quad (3)$$

In order to fit the parameters for the given model by training the given data points, we try to find the parameter u that minimizes Q_0 . As a result, u is selected, which is most likely to generate the labels as the same as in the training set. The minimization can be obtained by using the gradient and the Hessian. In order to prevent overfitting, one may introduce a prior over the parameters and optimize

$$Q(w) = Q_0(w) + \frac{\lambda}{2} w^T L w. \quad (4)$$

Therefore, the iteration scheme could simply be formulated with the Newton-Raphson iteration method. To extend the linear model to a nonlinear one, the original features x_n are transformed into the higher dimensional space k_n using a kernel function

$$k_{nn} = k(x_n, x_n). \quad (5)$$

The model of kernel logistic regression now presumes the a posteriori probabilities are given by

$$P_{nc}(w) = \frac{\exp(w_c^T k_n)}{\sum_c \exp(w_c^T k_n)}, \quad (6)$$

with k_n as the n th column of the kernel matrix K , and the unknown parameter $w = [\dots, w_c, \dots]$ refers to c classes. The parameters are determined in an iterative way by optimizing the regularized objective function. One of the limitations of the standard KLR is that all possible training samples are used to evaluate the kernel function, thus increasing the computational complexity and the memory

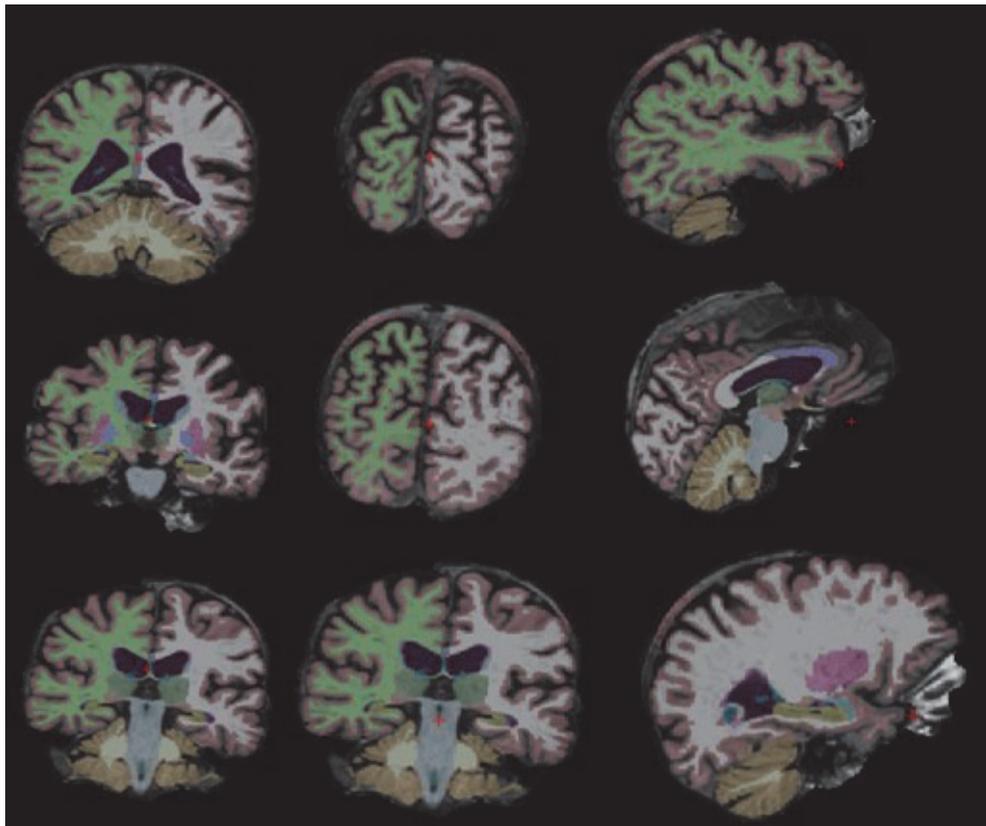


FIGURE 3: Block brain regions selected for AD classification using sMRI images.

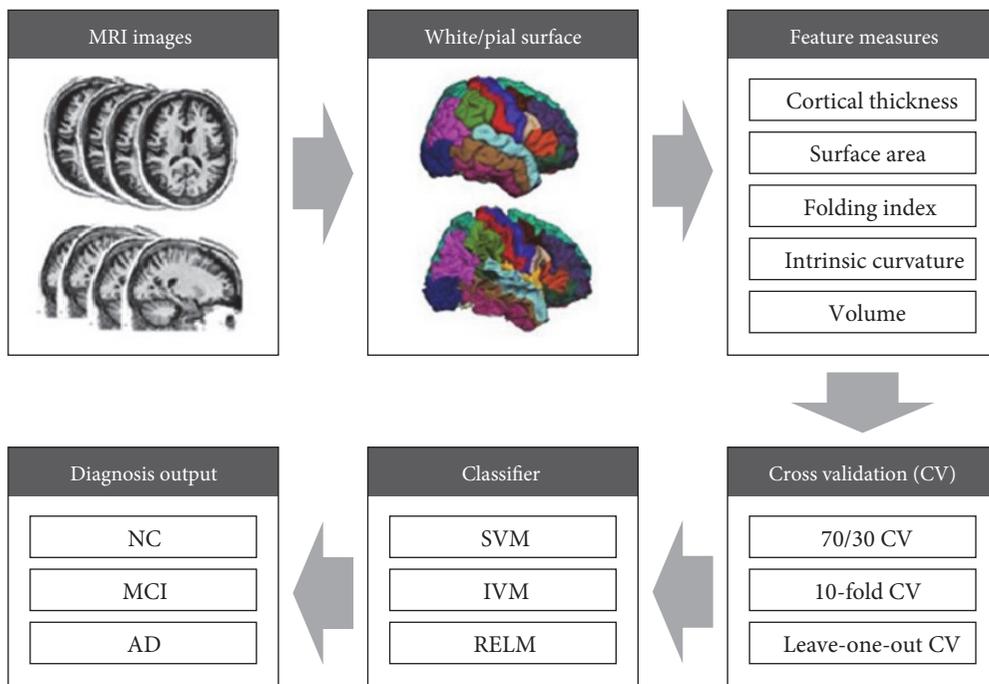


FIGURE 4: Block diagram of automatic diagnosis system.

requirement for large datasets. Meanwhile, the complexity of the classifier can be controlled by enforcing the sparseness in the learning model. The sparse kernel machine uses only the kernel function evaluated at a subset of the training data points for prediction of new inputs. The most common methods to implement sparseness are by introducing a suitable prior or by a subset selection. One of the popular examples for sparse kernel machine is SVM, which only supports that vectors are used to predict new inputs. The main idea of incorporating sparseness into KLR is to select a subset ν of V feature vectors out of the training set T . Thus, the kernel matrix only consists of the selected a subset ν of important kernels k_ν from all samples T . IVM uses a smaller fraction of training data to realize the sparse KLR. The subset is determined by a greedy manner. This method begins with empty set ν and then constructs the set of import vectors by successively adding data samples. The construction process of sets stops once the convergence criterion is reached. The convergence criterion is used by the ratio $\varepsilon = |Q_t - Q_{t-\Delta t}|/|Q_t|$ with a small integer Δt such as the regularization, and the kernel parameter ε defines the threshold for excluding import vectors. Consequently, this criterion influences the sparseness of the model.

3.4. RELM Classifier. Single hidden-layer feed forward neural networks (SLFNs), such as the back propagation (BP) learning algorithm, are widely used machine learning techniques for research in various fields. These methods minimize the cost function to maintain the accuracy within an acceptable range by searching the specific input weights and hidden layer biases, which leads to increase in computational cost. Extreme learning machine (ELM) is a learning algorithm implemented without iteratively tuning the artificial hidden nodes, thus decreasing the computation time [37]. ELM is an effective solution for SLFNs. The SLFN with L hidden nodes and an activation function $g(x)$ is expressed as

$$Y_L(x) = \sum_{i=1}^L \beta_i h_i(x) = h(x)\beta, \quad (7)$$

where $\beta = [\beta_1, \dots, \beta_L]^T$ is an output weight matrix between the hidden nodes and output nodes. $h_i(x)$ is the hidden node output. Unlike SVM and other BP-based methods, the parameters of the hidden layer such as the input weight w_i and the hidden layer biases b_i need not to be tuned and can be generated randomly before the training samples are acquired. Given N training samples $\{(x_j, t_j)\}_{j=1}^N$, ELM solves the learning problem by minimizing the error between t_j and Y_j :

$$\left\| H(w_1, \dots, w_N, b_1, \dots, b_N)\hat{\beta} - T \right\| = \min_{\beta} \left\| H\hat{\beta} - T \right\|, \quad (8)$$

where

$$\begin{aligned} H(w_1, \dots, w_N, b_1, \dots, b_N) &= \begin{bmatrix} g(w_1 \cdot x_1 + b_1) & \cdots & g(w_L \cdot x_1 + b_L) \\ \vdots & \cdots & \vdots \\ g(w_1 \cdot x_N + b_1) & \cdots & g(w_L \cdot x_N + b_L) \end{bmatrix}, \\ \beta &= \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}, \\ T &= \begin{bmatrix} t_1^T \\ \vdots \\ t_L^T \end{bmatrix}. \end{aligned} \quad (9)$$

Here, H is called the hidden layer output matrix. The output weights β can be calculated as

$$\beta = H^+ T, \quad (10)$$

where H^+ is the Moore-Penrose generalized inverse of the matrix H with the advantage of speed. ELM is well-suited for the tasks related to neuroimaging and big data for the classification of binary and multiclass settings. However, the decrease in computation time increases the error in the output, thus decreasing the accuracy. To increase the accuracy, ELM is combined with sparse representation. This hybrid algorithm performs classification in two fundamental steps [38–40]. In the first stage, the ELM network is trained with the conventional training approach. However, in the testing stage, reliability-based classification is used. In reliability-based classification, the ELM classifier is employed if the test data is correctly classified; otherwise, the sparse representation-based classification is used [41]. Additionally, a regularization term is added to improve generalization performance and make the solution more robust. Finally, the output weight of the RELM can be expressed as

$$\beta = \left(\frac{I}{C} + H^T H \right)^{-1} H^T T. \quad (11)$$

4. Experimental Results and Analysis

4.1. Permutation Testing. Permutation testing can be applied to assess the statistical significance of the classifier [42]. The assessment proceeds with the selection of the test statistic of the classifier and assigns random labels to the classifier by permuting the class labels for the training dataset. Permutation testing involves performing cross validation (CV) on data for which the diagnostic label has been randomly permuted. This leads to a distribution of classification results under the null hypothesis that the classifier cannot accurately predict the clinical labels from the data. The p value of the permuted prediction rate against the prediction rate with the original data labels indicates the significance of the classifier. In this work, we used 70/30 CV, 10-fold CV, and leave-one-out (LOO) CV methods. Experiments for both binary and multiclass classification were carried out with the same setup.

TABLE 3: Confusion matrix.

True class	Predicted class	
	S1	S2
S1	TP	FN
S2	FP	TN

4.2. Performance Evaluation Methods. We evaluated the performance of the proposed algorithm with the IVM, SVM, and RELM classifiers for each specific test including binary and multiclass classification tasks. The performance of the binary classification for the two subjects S1 and S2 can be visualized in a form of a confusion matrix as shown in Table 3. Diagonal elements of the matrix indicate the number of correct predictions by the classifier. The elements can be further divided into true positive (TP) and true negative (TN), which represent correctly identified controls. Similarly, the number of wrongly classified subjects may be represented by false positive (FP) and false negative (FN).

The accuracy measures the proportion of examples that are correctly labeled by the classifier.

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \quad (12)$$

However, for dataset with very unbalanced class distribution, accuracy in (12) may be a misleading performance metric. Thus, two performance metrics known as sensitivity and specificity are also used.

$$\text{SEN} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (13)$$

$$\text{SPE} = \frac{\text{TN}}{\text{TN} + \text{FP}}. \quad (14)$$

The sensitivity in (13) measures the rate of true positives while the specificity in (14) measures the rate of true negatives. The performance metrics for the multiclass classification are easily extended as the averaged ones on the OVA setting.

4.3. Binary Classification: Results and Analysis. The experimental results of binary classification (NC versus AD) are shown in Table 4, and those with feature selection are in Table 5. 141 subjects were randomly selected for the binary classification. Initially, we randomly segregated the training and testing dataset and used the first 111 randomly chosen subjects from each group for training and the remaining 30 subjects for testing the classifier. Similarly, in 10-fold cross-validation, all 141 subjects were randomly divided into equally sized subsets, that is, 10% testing subjects and 90% training subjects, for each of the 10-fold sets of the CV. In addition, for the nested validation, we repeated the classification experiment 10 times in the case of the 10-fold CV and leave-one-out CV, and 100 times in the case of the conventional 70/30 CV, to ensure the robustness of the classification results. The mean accuracy of all the repetitions was calculated by the final results.

TABLE 4: Performance of binary classification.

CV method	Classifier	Performance metrics		
		ACC (%)	SEN (%)	SPEC (%)
10-fold CV	SVM	60.10	74.63	88.81
	IVM	59.50	62.30	62.85
	RELM	77.30 ($p < 0.0001$)	62.12	79.85
LOO CV	SVM	78.01	75.81	79.12
	IVM	73.36	70.97	75.95
	RELM	75.66 ($p < 0.0001$)	72.13	77.22

TABLE 5: Performance of binary classification with feature selection.

CV method	Classifier	Performance metrics		
		ACC (%)	SEN (%)	SPEC (%)
10-fold CV	SVM	75.33	77.51	61.20
	IVM	60.20	62.50	81.10
	RELM	76.61 ($p < 0.0001$)	61.70	90.63
LOO CV	SVM	80.32	83.37	78.82
	IVM	74.47	87.10	64.56
	RELM	77.88 ($p < 0.0001$)	68.85	83.54

In Table 4 showing the baseline performance of different classifiers, all classifiers except IVM obtained good performance. There was no substantial difference, in terms of accuracy, between the results obtained with IVM and SVM, and RELM is better than the others in 10-fold CV; however, SVM is better than RELM in LOO CV. For the feature selection, the datasets of size $n \times d$ were mapped to the given k principal component framework and transformed into the dataset of size $n \times k$, where n is the number of subjects and d is the original number of features. The dataset originally consists of total 54 features. The number of PCs represented as k ranging from 2 to 20, with an incremental offset of two, was checked and the best one was selected for each classifier. From repeated simulations, we achieved the generally good accuracy when the value of k is set to 10. As shown in Table 5, by adopting feature selection, the similar performance characteristic was observed in terms of accuracy. From Figure 5, it is easily observed the effectiveness of the feature selection approach in 10-fold CV and LOO CV cases. It is meaningful that from our repeated simulations, we found that the results of 70/30 CV case, which is a widely used setting, are not stable (i.e., it has large variance with different trials) mainly due to overfitting problems, and thus, the results were not listed in this work.

4.4. Multiclass Classification: Results and Analysis. For multiclass classification, we adopted all labeled 214 subjects in Table 1. The same subjects were used in binary classification, and we adopted three CV methods. From Tables 6 and 7, it is easily observed that RELM outperforms SVM and IVM in terms of accuracy. From Figure 6, we also could see the effectiveness of the feature selection approach in 10-fold

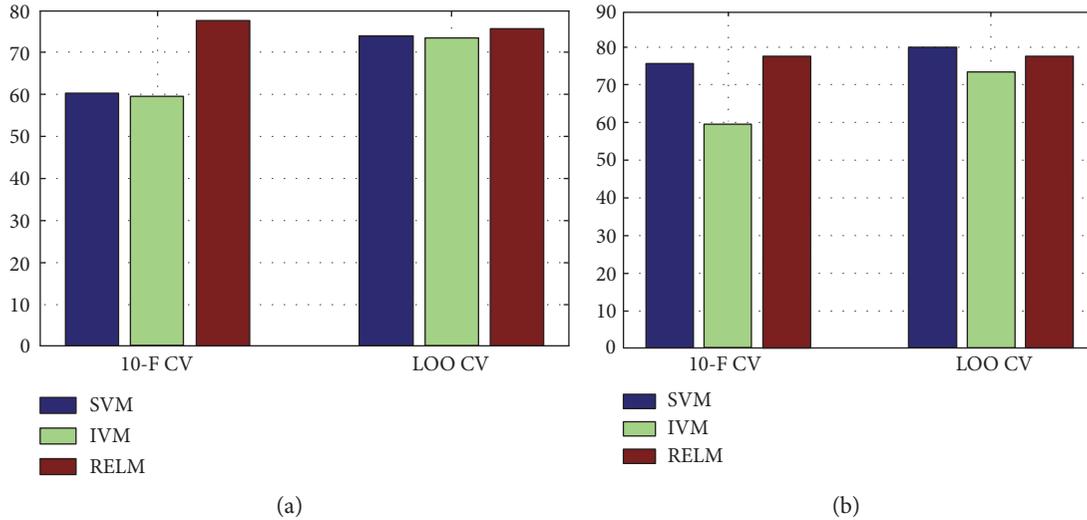


FIGURE 5: Performance comparison of binary classification in terms of accuracy: (a) binary classification and (b) binary classification with feature selection.

TABLE 6: Performance of multiclass classification.

CV method	Classifier	Performance metrics		
		ACC (%)	SEN (%)	SPEC (%)
10-fold CV	SVM	52.63	42.74	56.77
	IVM	54.90	46.18	60.82
	RELM	57.56	56.34	56.73
LOO CV	SVM	57.40	55.25	58.62
	IVM	55.50	60.78	52.22
	RELM	61.20	50.00	66.89

TABLE 7: Performance of multiclass classification with feature selection.

CV method	Classifier	Performance metrics		
		ACC (%)	SEN (%)	SPEC (%)
10-fold CV	SVM	56.60	50.59	56.38
	IVM	56.14	40.16	64.83
	RELM	59.81	58.25	58.82
LOO CV	SVM	58.30	57.12	60.32
	IVM	56.80	64.71	49.56
	RELM	61.58	54.00	62.25

CV and LOO CV cases. Similar to binary classification cases, on the 70/30 CV case, we obtained the experimental results with large variance, and thus, they were also excluded from the analysis. From the results, it is naturally driven that multiclass classification (which is the general form in clinical diagnosis of AD) assisted by RELM is effective compared to the other considered representative classifiers.

4.5. *Discussion on the Results.* It has been known that in many problem tasks, IVM generally performs similar with SVM in

terms of accuracy and provides probabilistic output. From our experiments, we could confirm that SVM generates better accuracy compared to IVM, which is mainly attributed to the robustness of SVM to outliers. The main impetus of this study was to compare representative classifiers, SVM, IVM, and RELM for binary and multiclass classification tasks. Trivially, the accuracy of the binary classification cases was higher than the corresponding multiclass classification cases. Also, the experimental results on large dataset of 214 subjects verified that RELM-based AD diagnosis framework (significantly) outperform the others with higher accuracy. To the best of our knowledge, this is the first study in which the RELM framework was used for multiclass classification on sMRI data obtained from the ADNI dataset. To classify the effectiveness of feature selection in combination with the classifiers, we utilized the PCA-based feature selection method as an efficient approach to validate its efficiency. It selects features that represent higher degrees of significance based on the internal linear SVM-based classification scores, and thus, it has the possibility of making classification significantly more accurate. The experimental results also support that such adoption of feature selection can be beneficial to improve the accuracy of the classifiers, SVM, IVM, and RELM. From the noteworthy results, we could conclude that the approaches for the stage classification can be used as an effective assistive tool for the establishment of a clinical diagnosis.

5. Conclusions and Future Work

The early diagnosis of AD and MCI is essential for patient care and research, and it is widely accepted that preventive measures plays an important role to delay or alleviate the progression of AD. For the classification task of different stages of AD progression, the smaller number of training samples and the larger number of feature representations are the major challenges. In this study, we investigated

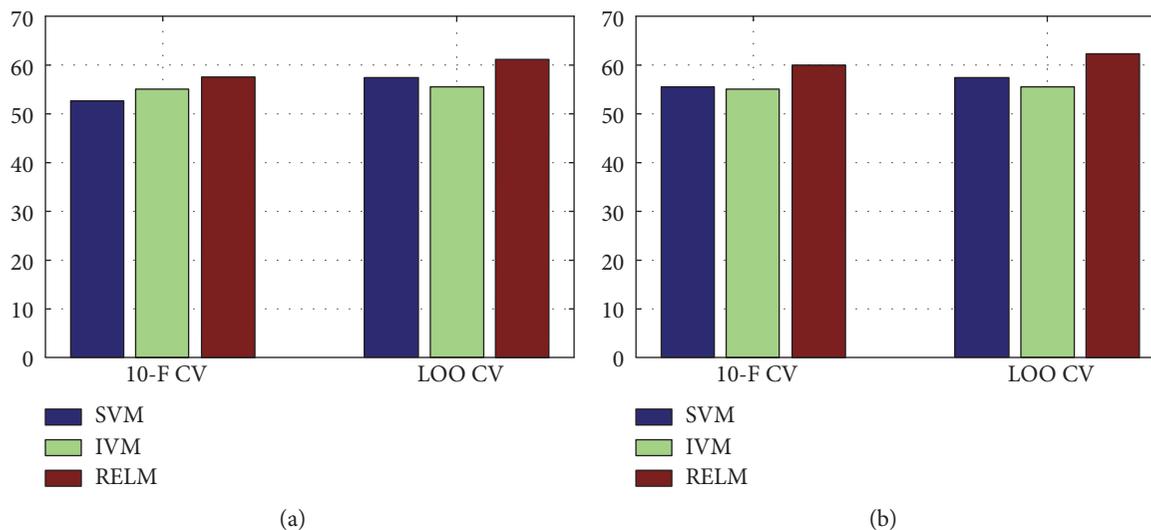


FIGURE 6: Performance comparison of multiclass classification in terms of accuracy: (a) multiclass classification and (b) multiclass classification with feature selection.

SVM, IVM, and RELM for the classification problem. In IVM, only the subsets of the input vectors of KLR are selected by minimizing the regularized cost function to reduce computation time. RELM is an effective solution for SLFNs implemented without iteratively tuning the artificial hidden nodes and adopts reliability-based classification where ELM is adopted if the test data is correctly classified, and sparse representation is selected for the other cases. Experiments on the ADNI dataset showed that RELM-based classifier could significantly improve accuracy in both binary and multiclass classification tasks. In addition, we could observe that adoption of the PCA-based feature selection could improve the accuracy slightly. While this study is focusing on the stage diagnosis of AD progression using sMRI alone, further study is still being carried out to improve the accuracy by elaborating the classifiers, possibly using a model ensemble approach, and feature selection. Also, the studies of adding more modalities such as fMRI and PET in combination with sMRI are also one of our future researches.

Conflicts of Interest

The authors declare that there are no competing interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Research Foundation (NRF) grant funded by the Korean government (NRF-2015R1D1A1A01060422, NRF-2014M3C7A1046050, and NRF-2016M3C7A1905477) and partially supported by the MISP (Ministry of Science, ICT & Future Planning), Korea, under the National Program for Excellence in SW supervised by the IITP (Institute for Information & Communications Technology Promotion). Data collection and sharing were funded by the Alzheimer's Disease Neuroimaging Initiative (ADNI) (National Institutes of Health Grant U01

AG024904) and DOD ADNI (Department of Defense, Award no. W81XWH-12-2-0012). ADNI is funded by the National Institute on Aging and the National Institute of Biomedical Imaging and Bioengineering and through generous contributions from the following: AbbVie, Alzheimer's Association; Alzheimer's Drug Discovery Foundation; Araclon Biotech; BioClinica Inc.; Biogen; Bristol-Myers Squibb Company; CereSpir Inc.; Cogstate; Eisai Inc.; Elan Pharmaceuticals Inc.; Eli Lilly and Company; EuroImmun; F. Hoffmann-La Roche Ltd. and its affiliated company Genentech Inc.; Fujirebio; GE Healthcare; IXICO Ltd.; Janssen Alzheimer Immunotherapy Research & Development LLC; Johnson & Johnson Pharmaceutical Research & Development LLC; Lumosity; Lundbeck; Merck & Co. Inc.; Meso Scale Diagnostics LLC; NeuroRx Research; Neurotrack Technologies; Novartis Pharmaceuticals Corporation; Pfizer Inc.; Piramal Imaging; Servier; Takeda Pharmaceutical Company; and Transition Therapeutics. The Canadian Institutes of Health Research provide funds to support ADNI clinical sites in Canada. Private sector contributions are facilitated by the Foundation for the National Institutes of Health (<http://www.fnih.org/>). The grantee organization is the Northern California Institute for Research and Education, and the study is coordinated by the Alzheimer's Therapeutic Research Institute at the University of Southern California. ADNI data are disseminated by the Laboratory for Neuroimaging at the University of Southern California. The Alzheimer's Disease Neuroimaging Initiative data used in preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (<http://adni.loni.usc.edu/>). As such, the investigators within the ADNI contributed to the design and implementation of ADNI and/or provided data but did not participate in the analysis or writing of this report. A complete listing of ADNI investigators can be found at http://adni.loni.usc.edu/wp-content/uploads/how_to_apply/ADNI_Acknowledgement_List.pdf.

References

- [1] American Psychiatric Association and Task Force on DSM-IV, *Diagnostic and Statistical Manual of Mental Disorders*, vol. xxv of DSM-IV, American Psychiatric Association, Washington, DC, 4th edition, 1994.
- [2] D. Schmitter, A. Roche, B. Maréchal et al., “An evaluation of volume-based morphometry for prediction of mild cognitive impairment and Alzheimer’s disease,” *NeuroImage: Clinical*, vol. 7, pp. 7–17, 2015.
- [3] Alzheimer’s association, “2016 Alzheimer’s disease facts and figures,” *Alzheimer’s and Dementia*, vol. 12, no. 4, pp. 459–509, 2016.
- [4] K. A. Johnson, N. C. Fox, R. A. Sperling, and W. E. Klunk, “Brain imaging in Alzheimer disease,” *Cold Spring Harbor Perspectives in Medicine*, vol. 2, no. 4, article a006213, 2012.
- [5] H. Hanyu, T. Sato, K. Hirao, H. Kanetaka, T. Iwamoto, and K. Koizumi, “The progression of cognitive deterioration and regional cerebral blood flow patterns in Alzheimer’s disease: a longitudinal SPECT study,” *Journal of the Neurological Sciences*, vol. 290, no. 1-2, pp. 96–101, 2010.
- [6] K. R. Gray, R. Wolz, R. A. Heckemann, P. Aljabar, A. Hammers, and D. Rueckert, “Multi-region analysis of longitudinal FDG-PET for the classification of Alzheimer’s disease,” *NeuroImage*, vol. 60, no. 1, pp. 221–229, 2012.
- [7] Y. J. Chen, G. Deutsch, R. Satya, H. G. Liu, and J. M. Mountz, “A semi-quantitative method for correlating brain disease groups with normal controls using SPECT: Alzheimer’s disease versus vascular dementia,” *Computerized Medical Imaging and Graphics*, vol. 37, no. 1, pp. 40–47, 2013.
- [8] F. Liu, L. Zhou, C. Shen, and J. Yin, “Multiple kernel learning in the primal for multi-modal Alzheimer’s disease classification,” *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 3, pp. 984–990, 2014.
- [9] J. M. Górriz, F. Segovia, J. Ramírez, A. Lassl, and D. S. Gonzalez, “GMM based SPECT image classification for the diagnosis of Alzheimer’s disease,” *Applied Soft Computing*, vol. 11, no. 2, pp. 2313–2325, 2011.
- [10] D. Zhang, Y. Wang, L. Zhou, H. Yuan, and D. Shen, “Multi-modal classification of Alzheimer’s disease and mild cognitive impairment,” *NeuroImage*, vol. 55, no. 3, pp. 856–867, 2011.
- [11] G. A. Papakostas, A. Savio, M. Graña, and V. G. Kaburlasos, “A lattice computing approach to Alzheimer’s disease computer assisted diagnosis based on MRI data,” *Neurocomputing*, vol. 150, pp. 37–42, 2015.
- [12] I. Beheshti and H. Demirel, “Probability distribution function-based classification of structural MRI for the detection of Alzheimer’s disease,” *Computers in Biology and Medicine*, vol. 64, pp. 208–216, 2015.
- [13] C. Aguilar, E. Westman, J. S. Muehlboeck, P. Mecocci, B. Vellas, and M. Tsolaki, “Different multivariate techniques for automated classification of MRI data in Alzheimer’s disease and mild cognitive impairment,” *Psychiatry Research: Neuroimaging*, vol. 212, no. 2, pp. 89–98, 2013.
- [14] E. Westman, J. S. Muehlboeck, and A. Simmons, “Combining MRI and CSF measures for classification of Alzheimer’s disease and prediction of mild cognitive impairment conversion,” *NeuroImage*, vol. 62, no. 1, pp. 229–238, 2012.
- [15] M. Li, Y. Qin, F. Gao, W. Zhu, and X. He, “Discriminative analysis of multivariate features from structural MRI and diffusion tensor images,” *Magnetic Resonance Imaging*, vol. 32, no. 8, pp. 1043–1051, 2014.
- [16] E. Moradi, A. Pepe, C. Gaser, H. Huttunen, and J. Tohka, “Machine learning framework for early MRI-based Alzheimer’s conversion prediction in MCI subjects,” *NeuroImage*, vol. 104, pp. 398–412, 2015.
- [17] E. E. Bron, M. Smits, H. Vrenken, F. Barkhof, and P. Scheltens, “Standardized evaluation of algorithms for computer-aided diagnosis of dementia based on structural MRI: the CAD dementia challenge,” *NeuroImage*, vol. 111, pp. 562–579, 2015.
- [18] S. Farhan, M. A. Fahiem, and H. Tauseef, “An ensemble-of-classifiers based approach for early diagnosis of Alzheimer’s disease: classification using structural features of brain images,” *Computational and Mathematical Methods in Medicine*, vol. 2014, Article ID 862307, 11 pages, 2014.
- [19] Y. Zhang, Z. Dong, P. Phillips et al., “Detection of subjects and brain regions related to Alzheimer’s disease using 3D MRI scans based on eigenbrain and machine learning,” *Frontier in Computational Neuroscience*, vol. 9, p. 66, 2015.
- [20] A. H. Andersen, W. S. Rayens, Y. Liu, and C. D. Smith, “Partial least squares for discrimination in fMRI data,” *Magnetic Resonance Imaging*, vol. 30, no. 3, pp. 446–452, 2012.
- [21] L. Mesrob, “DTI and structural MRI classification in Alzheimer’s disease,” *Advances in Molecular Imaging*, vol. 02, pp. 12–20, 2012.
- [22] Y. Fan, S. M. Resnick, X. Wu, and C. Davatzikos, “Structural and functional biomarkers of prodromal Alzheimer’s disease: a high-dimensional pattern classification study,” *NeuroImage*, vol. 41, no. 2, pp. 277–285, 2008.
- [23] M. Graña, M. Termenon, A. Savio, A. P. Gonzalez, J. Echeveste, and J. M. Pérez, “Computer aided diagnosis system for Alzheimer disease using brain diffusion tensor imaging features selected by Pearson’s correlation,” *Neuroscience Letters*, vol. 502, no. 3, pp. 225–229, 2011.
- [24] W. Lee, B. Park, and K. Han, “Classification of diffusion tensor images for the early detection of Alzheimer’s disease,” *Computers in Biology and Medicine*, vol. 43, no. 10, pp. 1313–1320, 2013.
- [25] K. I. Diamantaras and S. Y. Kung, *Principal Component Neural Networks*, Wiley, New York, 1996.
- [26] J. Barnes, J. W. Bartlett, L. A. van de Pol et al., “A meta-analysis of hippocampal atrophy rates in Alzheimer’s disease,” *Neurobiology of Aging*, vol. 30, no. 11, pp. 1711–1723, 2009.
- [27] B. Magnin, L. Mesrob, S. Kinkingnéhun et al., “Support vector machine-based classification of Alzheimer’s disease from whole-brain anatomical MRI,” *Diagnostic Neuroradiology*, vol. 51, no. 2, pp. 73–83, 2009.
- [28] A. M. Fjell, K. B. Walhovd, C. Fennema-Notestine et al., “CSF biomarkers in prediction of cerebral and clinical change in mild cognitive impairment and Alzheimer’s disease,” *The Journal of Neuroscience*, vol. 30, no. 6, pp. 2088–2101, 2010.
- [29] S. Liu, S. Liu, W. Cai et al., “Multimodal neuroimaging feature learning for multiclass diagnosis of Alzheimer’s disease,” *IEEE Transactions of Biomedical Engineering*, vol. 62, no. 4, pp. 1132–1140, 2015.
- [30] S. Liu, S. Liu, W. Cai, S. Pujol, R. Kikinis, and D. Feng, “Early diagnosis of Alzheimer’s disease with deep learning,” in *Proceedings/IEEE International Symposium on Biomedical Imaging: from nano to macro*, pp. 1015–1018, 2014.

- [31] F. Li, L. Tran, K. H. Thung, S. Ji, D. Shen, and J. Li, "A robust deep model for improved classification of AD/MCI patients," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 5, pp. 1610–1616, 2015.
- [32] J. Ye, T. Wu, J. Li, and K. Chen, "Machine learning approaches for the neuroimaging study of Alzheimer's disease," *IEEE Computer*, vol. 4, no. 4, pp. 99–101, 2011.
- [33] R. K. Rama, H. C. Park, and S.-W. Lee, "Sparse feature selection using import vector machines for classification of Alzheimer's disease," in *Proceedings of 2016 KING Fall Conference*, 2016.
- [34] M. Schaer, M. B. Cuadra, N. Schmansky, B. Fischl, J. P. Thiran, and S. Eliez, "How to measure cortical folding from MR images: a step-by-step tutorial to compute local gyrification index," *Journal of Visualized Experiments*, no. 59, article e3417, 2012.
- [35] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [36] J. Zhu and T. Hastie, "Kernel logistic regression and the import vector machine," *Journal of Computational and Graphical Statistics*, vol. 14, no. 1, pp. 185–205, 2005.
- [37] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.
- [38] W. Zhang, H. Shen, Z. Ji, G. Meng, and B. Wang, "Identification of mild cognitive impairment using extreme learning machines model," in *Intelligent Computing Theories and Methodologies: 11th International Conference, ICIC 2015*, August 2015.
- [39] X. Peng, P. Lin, T. Zhang, and J. Wang, "Extreme learning machine-based classification of ADHD using brain structural MRI data," *PloS One*, vol. 8, no. 11, article e79476, 2013.
- [40] M. N. I. Qureshi, B. Min, H. J. Jo, and B. Lee, "Multiclass classification for the differential diagnosis on the ADHD subtypes using recursive feature elimination and hierarchical extreme learning machine: structural MRI study," *PloS One*, vol. 11, no. 8, 2016.
- [41] E. Cambria and G. B. Huang, "Extreme learning machines," *IEEE Intelligent Systems*, vol. 28, no. 6, pp. 2–31, 2013.
- [42] P. Golland and B. Fischl, "Permutation tests for classification: towards statistical significance in image based studies," in *Information Processing in Medical Imaging: 18th International Conference*, C. Taylor and J. A. Noble, Eds., IPMI, 2003.

Research Article

Cell Detection Using Extremal Regions in a Semisupervised Learning Framework

Nisha Ramesh,^{1,2} Ting Liu,³ and Tolga Tasdizen^{1,2}

¹Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT 84112, USA

²Electrical and Computer Engineering Department, University of Utah, Salt Lake City, UT 84112, USA

³Google, Los Angeles, CA 90291, USA

Correspondence should be addressed to Nisha Ramesh; nshramesh@sci.utah.edu

Received 3 November 2016; Revised 29 March 2017; Accepted 19 April 2017; Published 14 June 2017

Academic Editor: Ashish Khare

Copyright © 2017 Nisha Ramesh et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper discusses an algorithm to build a semisupervised learning framework for detecting cells. The cell candidates are represented as extremal regions drawn from a hierarchical image representation. Training a classifier for cell detection using supervised approaches relies on a large amount of training data, which requires a lot of effort and time. We propose a semisupervised approach to reduce this burden. The set of extremal regions is generated using a maximally stable extremal region (MSER) detector. A subset of nonoverlapping regions with high similarity to the cells of interest is selected. Using the tree built from the MSER detector, we develop a novel differentiable unsupervised loss term that enforces the nonoverlapping constraint with the learned function. Our algorithm requires very few examples of cells with simple dot annotations for training. The supervised and unsupervised losses are embedded in a Bayesian framework for probabilistic learning.

1. Introduction

Automatic cell detection is a fundamental problem that is useful for numerous cell-based studies and quantifications. Also, cell detection is a preliminary step for solving high-level problems, such as cell segmentation, tracking, and analyzing cell data. Accurately detecting a large number of cells in dense images is challenging, for example, when there is occlusion, cells of interest touch each other. Therefore, it is important to develop robust generic algorithms that can tolerate noise and be used on a variety of cells (different modalities and shapes). Detection based on local image information can be erroneous, since they can be associated with imaging artifacts or noise. Including prior information about the objects to be segmented helps in resolving these issues. The priors, learned from the training data, can then be used to learn strategies to detect cells. The process of annotating complete masks for the cells of interest in the training examples is tedious. Working with minimalistic annotations as in [1], where a dot is placed inside each cell in the training images, may be a more plausible solution. These observations

motivate us to study semisupervised methods that use minimal training information.

Semisupervised learning benefits from labeled and unlabeled data during the training. Since generating large labeled datasets is time-consuming, it is profitable to build algorithms that can benefit from unlabeled data. This approach is useful in cases when labeled training data is sparse, but there is a large amount of unlabeled data. Incorporating unlabeled data with sparsely labeled training data can help improve the learning accuracy. The goal of our algorithm is to achieve better results in comparison to those of existing algorithms with a significantly smaller amount of training data in a “semisupervised” framework. We also show that in a transductive setting, when we use all the labeled and unlabeled data, we improve the current state-of-the-art results.

Many algorithms have been proposed for semisupervised learning in general, as well as specific applications [2, 3]. Our work is inspired by the semisupervised approach used for electron microscopy image segmentation based on a merge tree structure [4]. The framework in [4] tries to learn the

probability of merging/splitting regions in a hierarchical tree structure, whereas we learn a classifier to predict the probability of a region representing a cell. We extend the work in [4] to solve the detection problem by expressing the nonoverlapping constraint as an unsupervised loss function. The main difference is in our construction of the path loss function, to ensure at most one cell is picked in every path versus monotonically nonincreasing constraints on the merging of regions in [4].

We aim to minimize the need for labeled data by taking advantage of the possible interdependence among the unsupervised data. We transform the intensity images to probability maps, such that the objects can be distinguished from the background. We use ILASTIK [5] to train a pixel classifier to generate probability maps. The MSER detector is used on the probability maps to generate a collection of distinct regions [6]. Every connected component in the image is represented using a hierarchical MSER tree. The MSER tree comprises components over different thresholds, from which a subset of nonoverlapping regions that resemble potential cells is selected as the final segmentation. A dot is placed inside each cell in the training images to contribute to the supervised loss. The major contribution of this paper is formulating the nonoverlapping constraint as a differentiable loss function that can be effectively used to steer the unsupervised search to help find a valid solution. We embed the supervised and unsupervised losses in a Bayesian framework for probabilistic learning. The parameters used in the framework are estimated from the data iteratively. We compare our algorithm extensively with [1], since the authors also use dot annotations as ground truth for training. Most of the current competing methods on cell detection/segmentation use complete cell masks as ground truth for training. Hence, it would not be a fair comparison, because we learn from dot annotations. Similar to [1], we use the MSER region detector to generate candidate cell regions and build a hierarchical representation of the image. One disadvantage in [1] while using reduced labeled data is that it does not directly explore the unsupervised data when searching for the most favorable classification function. Our method exploits the underlying correlation between unsupervised data samples. Also, in [1], the nonoverlapping constraint is embedded in the inference, whereas our supervised loss drives the solution to choose regions similar to those of the cells from the training images, and our novel unsupervised loss enforces that at most only one cell is picked in every path.

We begin with a review of the previous related work on cell detection methods in Section 2. In Section 3, we illustrate our detection framework. Experimental results are shown in Section 4, and we summarize our current work and discuss possible extensions for the future in Section 5.

2. Related Work

Detection of cells is a fundamental problem because cell morphology and its characteristics are useful for further studies and analysis. The different types of imaging techniques, stains, cell types, and densities contribute to the variability

in the cells (shape, size, and texture) [7]. Since the detection and segmentation problems are often addressed together, we will discuss the significant literature for both.

The most fundamental approach for detecting cell regions is intensity thresholding [8, 9]. For cells with homogeneous intensities, which are in contrast to the background and are distinctly separable, simple global thresholding or adaptive thresholding can be used. In reality, the assumption does not hold true. It is very hard to find cell regions with homogeneous intensities. Thresholding may be used as a preprocessing step for the detection problem. The intensity images can also be transformed to represent some derived features using filtering operations (e.g., filter to detect edges, blob detection filters). Edge detection can be used to generate the boundaries of the cells that represent cell contours [10, 11]. These techniques can be included in the preprocessing steps [12, 13]. Even morphological filtering can be used for preprocessing or postprocessing the images to enable detection of individual cells [14–16].

Region segmentation is a popular approach to detect cells, grouping pixels that belong to a cell. Region-growing algorithms start from selected seed points and augment more connected points to form a labeled image [17]. Region segmentation approaches were initially based on clustering algorithms, where similar image pixels were aggregated in an unsupervised framework. The watershed transform, one of the most popular segmentation approaches, uses edge maps to segment an image into multiple regions/catchment basins [18]. The watershed transform usually results in oversegmentation in the case of cell images. Several algorithms use the watershed transform as an initial step in the segmentation pipeline [19–21]. Belongie et al. [22] used Gaussian mixture models to cluster pixels using color and texture features.

Also, considering candidates from a hierarchy of regions has been demonstrated to be an effective method for cell detection and segmentation [1, 4, 19, 20, 23–25]. Hierarchical image segmentation encompasses image segmentations at different detail levels/thresholds. The segmentations are nested. The higher-level regions can be produced from a combination of regions from segmentations at lower levels. Hierarchical methods preserve the spatial and neighboring information among segmented regions. A number of methods can be used to generate the cell candidates, generally running simple algorithms with different parameters. A graph or tree is constructed in which the edges exist between overlapping regions. The watershed transform, an ultrametric contour map (UCM) [26]; the Felzenszwalb's method [27]; the MSER detector [6]; or any other method can be used to build a hierarchical representation. A conservative global intensity, followed by the watershed transform and a persistence-based clustering (PBC) agglomeration, is used in [28]. In [29], the watershed segmentation with different thresholds gives cell candidates, and a learning algorithm based on conditional random field (CRF) is utilized to find the best ensembles to present the final segmentation results. In [30], UCM is used to get a hierarchical representation; a sparse representation-based cell shape model is learned to encode the high-level shape constraints, which are combined with low-level cellular features to produce

probability scores of region segmentation candidates. For cell segmentation from candidates obtained from region merging, the supervised term learns the probability of merging regions using examples of true and false merges [30–33]. Dynamic programming or ILP can be used to find a non-overlapping set of candidates from a hierarchical representation. In [1], the MSER detector is used to build a hierarchical representation and the costs are learned in a structured SVM framework. The nonoverlapping constraint is enclosed in the inference problem. A novel topological loss function to capture the prior information of the cells is used to learn the cost in the structured SVM framework [23].

Graph-cut methods have also been used to solve the cell segmentation problem. It is efficient to represent an image as a graph, where pixels are the nodes and the edge weights are based on the similarity with neighbors [21, 34]. The segmentation problem is then solved as a graph partitioning problem. To reduce the computational load, the graphical representation has also been constructed using superpixels. Superpixels are a group of pixels in spatial proximity that have similar intensity and texture. Superpixels can be obtained by oversegmentation, thereby preserving the image features [35, 36]. In Zhang et al. [21], a region adjacency graph is constructed using superpixels, and segmentation is inferred by partitioning the graph using correlation clustering.

Cells can also be represented using deformable models. Among such methods, variational image segmentation with level sets has been a prominent choice due to their attractive properties such as adaptive topology, which can naturally evolve. An energy functional is defined based on the intensity and shape information. By minimizing the energy functional, the deformable model evolves. The energy terms need to be chosen carefully to minimize the errors in the solution. An extension of level sets to segment cells is used in [37], but it is computationally expensive. Implicit parametric shape models, namely, disjunctive normal shape models, are used to learn representations of cells accurately, where the time complexity increases with the density of the images [38].

Most recently, fully convolutional networks (FCNs) have been used for cell segmentation. FCNs have been extended for cell segmentation in [39]. The algorithm uses cell masks as ground truth for training the network. In [40], the algorithm learns to predict density maps using FCNs in microscopy images. The FCN is trained using dot annotations in simulated data. Since the dot annotations are each represented by a Gaussian, their method cannot be applied to detect cells with arbitrary sizes and shapes.

This paper proposes a new semisupervised framework for detecting cells. Minimalistic dot annotations are used, a dot being placed inside each cell. Our method is evaluated on four datasets and is able to learn a model that achieves better detection accuracy with limited training data, in our evaluation, despite the variation between the datasets. To show the competence of our method, we compare the segmentation accuracy with that of competing methods that train using complete cell masks for only one dataset, since we do not have a one-to-one match for the data used in the other cases.

3. Method

To generate cell candidates, we need to be able to distinguish the cells from the background pixels. We use the cell centers to train a pixel classifier using ILASTIK. The dot annotations are each represented by a Gaussian, where the cell centers represent the peak and its width is determined based on the standard deviation. Using the density map, which is formed by the superposition of all the Gaussians, we pick the positive and negative samples. While selecting the positive samples, we use a small value for the standard deviation to enable us to pick only pixels that are close to the centers of the cells. Similarly, we use a large value for the standard deviation to enable us to pick only the background pixels as negative samples. Due to the limited training size in our experiments, we used only two images to learn the pixel classifier in each dataset. The number of pixels selected depends on the number of cells in the image, which varies with different datasets. A set of candidate regions, $\mathbf{R} = \{R_1, R_2, \dots, R_N\}$, is detected using the MSER detector on the cell probability maps. Let $f : R^D \rightarrow B$, where $B = \{0, 1\}$ and D is the dimensionality of the input vector, be a binary indicator function used to predict regions that represent cells. Based on these predictions, we pick a subset of nonoverlapping regions with high similarity to the class of interest from the training examples. The classifiers are trained in a semisupervised framework. The framework includes supervised quadratic loss that relies on user annotations. Similarly, there is an unsupervised loss that restrains the solution to choose nonoverlapping regions. The details of the algorithm are provided below.

3.1. Extremal Region Generation. We use the efficient maximally stable extremal region (MSER) detector on the output of the pixel classifier to find a representative subset of all extremal regions. Extremal regions are regions in which there is a distinct contrast between intensities inside the region and those of its boundary, that is, the average image intensity inside the region is higher or lower than the intensity at the boundary. The image I is thresholded at all possible levels. At any level of threshold t , the connected components form the set of extremal regions. By having varying thresholds, a pool of extremal regions is generated. In Figure 1(a), we see the image thresholded at increasing values of t . An important property of extremal regions is their nestedness. The extremal regions A and B are nested if $A \subset B$ or $A \supset B$. Similarly, regions A and B are nonoverlapping if $A \cap B = \emptyset$. The hierarchical relationships in the tree shown in Figure 1(b) correspond to the nestedness of the regions in Figure 1(a). The tree structure is utilized by our learning framework to determine which of those candidates correspond to cells. The MSER detector from [6] considers only regions that are maximally stable, meaning that stable regions have a lower variation than the regions that are one level below or above. The stability threshold is fixed to a high value to accommodate all potential cell candidates. Similarly, all the parameters of the MSER algorithm are set to enable us to work with the complete tree that can be generated from the image. Hence, regions that have a slight variation from the background

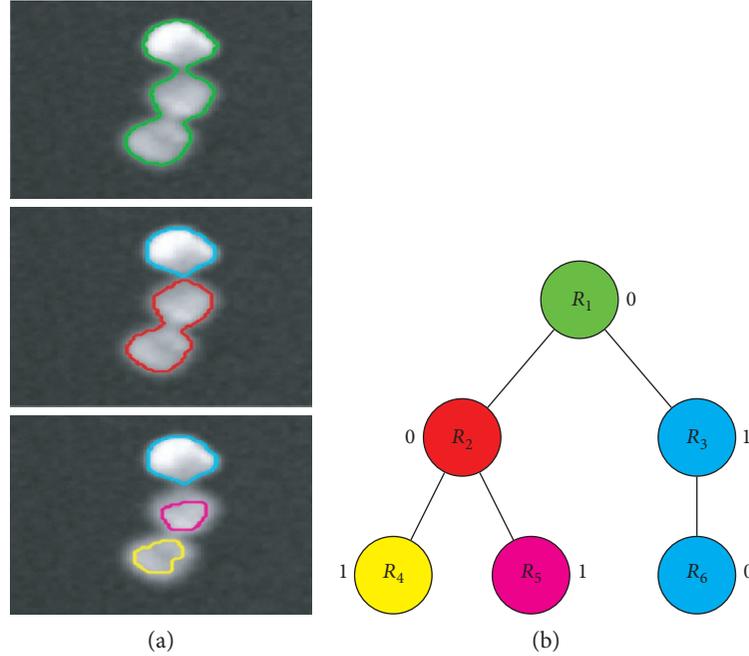


FIGURE 1: (a) Top-bottom: an example of the boundaries of multiple MSERs that represent the potential cell regions by thresholding the image at increasing values of t . (b) The parent-child relationships in the tree correspond to the nestedness of the regions. The tree structure is utilized in our semisupervised framework. The colors indicate individual cell hierarchy.

are also selected. MSER can extract bright (dark) regions on dark (bright) backgrounds, based on what is selected.

3.2. Path Consistency Constraint. The path consistency constraint enforces selection of at most one cell in every path in the tree representation of the connected component. Every image I is represented using a forest of trees. Each tree represents a connected region from the image and can contain multiple cells. Let \mathbf{P} represent all the possible paths in a tree. All paths in a tree do not necessarily have the same depth. For each path $p_i \in \mathbf{P}$ with n_{p_i} nodes, the indices of the regions that comprise the path are given as $d_i \{d_i^0, d_i^1, \dots, d_i^{n_{p_i}-1}\}$. The feature vector for any region d_i^k is represented as $\mathbf{x}_{d_i^k}$. The nonoverlapping loss using a disjunctive normal form to ensure that at most one cell is picked in the i th path can be represented as

$$F_i = \bigvee_{j=0}^{n_{p_i}} \underbrace{\bigwedge_{k=0}^{n_{p_i}-1} f^j(\mathbf{x}_{d_i^k})}_{b_j}, \quad (1)$$

$$f^j(\mathbf{x}_{d_i^k}) = \begin{cases} f(\mathbf{x}_{d_i^k}) & \text{if } j = k \\ \neg f(\mathbf{x}_{d_i^k}) & \text{otherwise} \end{cases},$$

where $f(\mathbf{x}_{d_i^k})$ represents the score for the region whose indices are given as d_i^k . We introduce a binary indicator vector $\mathbf{y} = \{y_1, y_2, \dots, y_N\}$ such that $y_i = 1$ implies the region R_i is picked as a potential cell. An example of a valid combination of regions to represent potential cells for the hierarchical tree in Figure 1(b) is given as $\mathbf{y} = [0 \ 0 \ 1 \ 1 \ 1 \ 0]$ such that regions R_3 , R_4 , and R_5 are picked. The path consistency constraint

can be easily converted to a differentiable model. We convert the disjunctive normal form into a differentiable model by (1) using DeMorgan's laws to replace unions with intersections and complements; (2) representing the intersections as products; (3) representing the detection classifier using a differentiable classification function, such as an artificial neural network; and (4) relaxing the binary indicator function as $\tilde{f}(\mathbf{x}_{d_i^k})$ to estimate the probability of a region to represent a cell. First, the conjunction of binary variables $\bigwedge_{k=0}^{n_{p_i}-1} f^j(\mathbf{x}_{d_i^k})$ is equivalent to the product $\prod_{k=0}^{n_{p_i}-1} f^j(\mathbf{x}_{d_i^k})$. Next, using DeMorgan's laws, we can express the disjunction $\bigvee_{j=0}^{n_{p_i}} b_j$ as negation of conjunctions, $\neg \bigwedge_{j=0}^{n_{p_i}} \neg b_j$, which in turn can be replaced by $1 - \prod_{j=0}^{n_{p_i}} (1 - b_j)$. We need to choose \tilde{f} such that it is differentiable. We implement a multilayer perceptron (MLP) with one hidden layer parameterized using \mathbf{w} as our detection classifier, $\tilde{f}_{\mathbf{w}}(\mathbf{x}_{d_i^k})$. The number of neurons in the input layer is dependent on the size of the data vector (number of features including the bias node described in Section (3.3)). For the hidden layer, we use half the number of nodes used in the input layer. Gradient-based optimization algorithms are used to find the model parameters. The activation function used in the MLP is the tanh function. Finally, the approximate path loss is given as

$$F_i = 1 - \prod_{j=0}^{n_{p_i}} \left(1 - \prod_{k=0}^{n_{p_i}-1} \tilde{f}_{\mathbf{w}}^j(\mathbf{x}_{d_i^k}) \right), \quad (2)$$

$$\tilde{f}_{\mathbf{w}}^j(\mathbf{x}_{d_i^k}) = \begin{cases} \tilde{f}_{\mathbf{w}}(\mathbf{x}_{d_i^k}) & \text{if } j = k \\ 1 - \tilde{f}_{\mathbf{w}}(\mathbf{x}_{d_i^k}) & \text{otherwise} \end{cases}.$$

We would like to find \mathbf{w} so that the predictions, $\tilde{f}_{\mathbf{w}}(\mathbf{x})$, satisfy the path consistency constraint, where \mathbf{x} represents the feature vector of a region.

3.3. Bayesian Semisupervised Formulation for Learning. Our framework for learning uses both supervised and unsupervised data. Supervised data corresponds to the regions that represent true cells, generated from the ground truth centroids. Let the supervised data represented by $(\mathbf{X}_s, \mathbf{y}_s)$ denote the features of the candidate regions and annotations that indicate if the region is a true cell. Unsupervised data refers to the unlabeled images. Since we train with a significantly small amount of labeled data, the unlabeled samples from the training set contribute to the unsupervised data. We can also incorporate more unlabeled data in a “transductive learning” setting by including data from the testing images. Hence, unsupervised data can have contributions from the training images (unlabeled) as well as the testing images (unlabeled). Let \mathbf{X} be the collection of all supervised and unsupervised samples. The feature vector for every region can be grouped into the following categories:

- (1) Intensity—histogram of intensities of the region, absolute mean differences, $L1$ and $L2$ distances, and absolute entropy differences between histograms of the region border and a dilation of it for two different dilation radii.
- (2) Area—areas of the region normalized by the area of the image and perimeters and the boundary length of the regions normalized by the length of the image diagonal.
- (3) Shape—a shape descriptor using a histogram that represents the variation and distribution of the boundary of the region on a size-normalized polar coordinate system. The curvature and roundness ratio are also computed for the region.
- (4) Texture—features such as local entropy, standard deviation, and local range of the region are used to describe the texture of the region.

The supervised loss can be represented as an i.i.d. Gaussian $N(0, \sigma_s)$ that penalizes the prediction errors and tries to minimize errors from the true estimate.

$$P(\mathbf{y}_s | \mathbf{X}_s, \mathbf{w}, \sigma_s) = \frac{1}{(\sqrt{2\pi}\sigma_s)^{N_s}} \exp\left(-\frac{\|\mathbf{y}_s - \tilde{f}_{\mathbf{w}}(\mathbf{X}_s)\|_2^2}{2\sigma_s^2}\right), \quad (3)$$

where N_s is the number of training samples and σ_s is the standard deviation that can be estimated while learning the model. The training data helps in avoiding a trivial solution such as the case in which the predictions for all regions are 0. Next, we formulate the unsupervised loss to constrain the solution such that at most one cell may be picked in every path of the MSER tree. For every tree, all possible paths are deduced, and the path function F is computed as in (2). Let $\tilde{\mathbf{F}}_{\mathbf{w}} = [\tilde{F}_1, \tilde{F}_2, \dots, \tilde{F}_{N_p}]$ be the collection of path functions, where N_p is the number of

paths from all the images. The unsupervised loss likelihood is an i.i.d. Gaussian $N(0, \sigma_u)$ that penalizes the difference between each element of $\tilde{\mathbf{F}}_{\mathbf{w}}$ and 1. Let $\mathbf{1}$ be an N_p dimensional vector of ones that force the predictions of the classifier to conform to the path constraints.

$$P(\mathbf{1} | \mathbf{X}, \mathbf{w}, \sigma_u) = \frac{1}{(\sqrt{2\pi}\sigma_u)^{N_p}} \exp\left(-\frac{\|\mathbf{1} - \tilde{\mathbf{F}}_{\mathbf{w}}(\mathbf{X})\|_2^2}{2\sigma_u^2}\right), \quad (4)$$

where σ_u is the standard deviation that can be estimated while learning the model. The standard deviation parameters σ_s and σ_u control the contributions of the supervised loss and the unsupervised path loss in the learning framework. They can be tuned using a validation set or estimated from the data. Finally, we include a regularization term to prevent overfitting. This term constrains any abrupt change in the model parameters, thereby establishing the smoothness constraint for the solution.

$$P(\mathbf{w}) = \frac{1}{(\sqrt{2\pi})^D} \exp\left(-\frac{\|\mathbf{w}\|_2^2}{2}\right). \quad (5)$$

By applying Bayes’ rule, we have the posterior distribution of \mathbf{w} as

$$\begin{aligned} P(\mathbf{w} | \mathbf{X}, \mathbf{X}_s, \mathbf{y}_s, \sigma_u, \sigma_s) &\propto P(\mathbf{w}) \cdot P(\mathbf{1} | \mathbf{X}, \mathbf{w}, \sigma_u) \\ &\cdot P(\mathbf{y}_s | \mathbf{X}_s, \mathbf{w}, \sigma_s) \\ P(\mathbf{w} | \mathbf{X}, \mathbf{X}_s, \mathbf{y}_s, \sigma_u, \sigma_s) &\propto \frac{1}{(\sqrt{2\pi})^D} \exp\left(-\frac{\|\mathbf{w}\|_2^2}{2}\right) \\ &\cdot \frac{1}{(\sqrt{2\pi}\sigma_u)^{N_p}} \exp\left(-\frac{\|\mathbf{1} - \tilde{\mathbf{F}}_{\mathbf{w}}(\mathbf{X})\|_2^2}{2\sigma_u^2}\right) \\ &\cdot \frac{1}{(\sqrt{2\pi}\sigma_s)^{N_s}} \exp\left(-\frac{\|\mathbf{y}_s - \tilde{f}_{\mathbf{w}}(\mathbf{X}_s)\|_2^2}{2\sigma_s^2}\right). \end{aligned} \quad (6)$$

3.4. Estimating the Parameters. We infer the model parameters \mathbf{w} and σ_s and σ_u using maximum a posteriori estimation.

$$\mathbf{w}^* = \arg \max_{\mathbf{w}, \sigma_u, \sigma_s} P(\mathbf{w} | \mathbf{X}, \mathbf{X}_s, \mathbf{y}_s, \sigma_u, \sigma_s). \quad (7)$$

We effectively minimize the negative logarithm of the posterior.

$$\begin{aligned} L(\mathbf{w}, \sigma_s, \sigma_u) &= N_p \log \sigma_u + \underbrace{\frac{1}{2\sigma_u^2} \|\mathbf{1} - \tilde{\mathbf{F}}_{\mathbf{w}}(\mathbf{X})\|_2^2}_{E_w^u} \\ &+ \underbrace{\frac{1}{2} \|\mathbf{w}\|_2^2}_{E_w^c} + \underbrace{\frac{1}{2\sigma_s^2} \|\mathbf{y}_s - \tilde{f}_{\mathbf{w}}(\mathbf{X}_s)\|_2^2 + N_s \log \sigma_s}_{E_w^s}, \end{aligned} \quad (8)$$

$$[\mathbf{w}, \sigma_s, \sigma_u] = \arg \min_{\mathbf{w}, \sigma_s, \sigma_u} (E_w^U + E_w^S + E_w^C). \quad (9)$$

The parameters \mathbf{w} and σ_u and σ_s are updated alternatively. Gradient descent is used to update \mathbf{w} . We alternatively update σ_s and σ_u along with \mathbf{w} . Setting $\partial L / \partial \sigma_u = 0$ and $\partial L / \partial \sigma_s = 0$, we can derive closed-form expressions for σ_u and σ_s .

$$\begin{aligned} \sigma_u &= \frac{\|1 - \tilde{\mathbf{F}}_w(\mathbf{X})\|_2}{\sqrt{N_p}}, \\ \sigma_s &= \frac{\|\mathbf{y}_s - \tilde{f}_w(\mathbf{X}_s)\|_2}{\sqrt{N_s}}. \end{aligned} \quad (10)$$

The gradient of (8) with respect to the classifier parameter \mathbf{w} is

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{w}} &= \mathbf{w}^T - \frac{1}{\sigma_u^2} (\mathbf{1} - \tilde{\mathbf{F}}_w(\mathbf{X}))^T \frac{\partial \tilde{\mathbf{F}}_w(\mathbf{X})}{\partial \mathbf{w}} \\ &\quad - \frac{1}{\sigma_s^2} (\mathbf{y}_s - \tilde{f}_w(\mathbf{X}_s))^T \frac{\partial \tilde{f}_w(\mathbf{X}_s)}{\partial \mathbf{w}}, \\ \frac{\partial \tilde{\mathbf{F}}_w(\mathbf{X})}{\partial \mathbf{w}} &= \left[\left(\frac{\partial \tilde{F}_1}{\partial \mathbf{w}} \right)^T, \left(\frac{\partial \tilde{F}_2}{\partial \mathbf{w}} \right)^T, \dots, \left(\frac{\partial \tilde{F}_{N_p}}{\partial \mathbf{w}} \right)^T \right], \\ \frac{\partial \tilde{F}_i}{\partial \mathbf{w}} &= \frac{\partial}{\partial \mathbf{w}} \left[1 - \prod_{j=0}^{n_{p_i}} \left(1 - \prod_{k=0}^{n_{p_i}-1} \tilde{f}_w^j(\mathbf{x}_{d_i^k}) \right) \right] \\ &= - \frac{\partial}{\partial \mathbf{w}} \left[\prod_{j=0}^{n_{p_i}} \left(1 - \prod_{k=0}^{n_{p_i}-1} \tilde{f}_w^j(\mathbf{x}_{d_i^k}) \right) \right] \\ &= - \sum_{j=0}^{n_{p_i}} \prod_{r \neq j} \left(1 - \prod_{k=0}^{n_{p_i}-1} \tilde{f}_w^r(\mathbf{x}_{d_i^k}) \right) \\ &\quad \cdot \frac{\partial}{\partial \mathbf{w}} \left(1 - \prod_{k=0}^{n_{p_i}-1} \tilde{f}_w^j(\mathbf{x}_{d_i^k}) \right) \\ &= \sum_{j=0}^{n_{p_i}} \prod_{r \neq j} \left(1 - \prod_{k=0}^{n_{p_i}-1} \tilde{f}_w^r(\mathbf{x}_{d_i^k}) \right) \\ &\quad \cdot \sum_{k=0}^{n_{p_i}-1} \left(\prod_{q \neq k} \tilde{f}_w^j(\mathbf{x}_{d_i^q}) \right) \frac{\partial \tilde{f}_w^j(\mathbf{x}_{d_i^k})}{\partial \mathbf{w}}, \end{aligned} \quad (11)$$

where $\tilde{f}_w^j(\mathbf{x}_{d_i^k})$ represents the detection probability, which is predicted using an ANN as a classifier. After the model parameters are learned, we can use the model, \tilde{f}_w , to predict the detection probabilities. Initially, a random set of weights is chosen, $\sigma_s = 1$. The weights \mathbf{w} are then updated using gradient descent based only on the supervised loss; σ_s is also alternatively updated. The supervised loss gives an approximate solution for the weights, which can be fine-tuned using the unsupervised loss. The weights are updated using the gradients computed using the supervised and the unsupervised loss for path consistency, alternatively updating \mathbf{w} and σ_s and σ_u . We then

select a set of nonoverlapping regions using the inference algorithm described in the next section to identify true cells.

3.5. Inference. The probability of a node/region representing a cell f_i is predicted using our classifier. The goal of the inference procedure is to select with minimal cost a subset of nonoverlapping nodes that represent the cells by imposing constraints on the hierarchical tree. Each node in the tree is a potential cell candidate. A label $y_i = 1$ or $y_i = 0$ is assigned to each node indicating whether the node is selected as a valid cell. The complete set of labels is represented as \mathbf{Y} . The detection labels given to all nodes must satisfy the path loss such that at most one cell is picked in every path.

We have explored two approaches to select the cells. Firstly, we used a greedy approach for inferring the final detections. The greedy inference algorithm selects the node with the highest score in a given path predicted using our classifier and assigns it as a label=1. The other nodes in the path are assigned as a label=0. The nodes with a label=1 are the final detected cells.

Alternatively, we can use a bottom-up/top-down algorithm to find the optimal solution imposing the path constraints in the tree structure. If a node i is selected, then, $y_i = 1$ and all of its descendants should be labeled as 0. If the node i is not selected, then, problem reduces to finding the best possible cells from the subtrees of the node i . All nodes with a label=1 will be selected as the detected cells. We formulate our detection problem as a constrained optimization problem

$$\min_{\mathbf{Y}} \sum_{y_i \in \mathbf{Y}} -y_i \log(f_i) - (1 - y_i) \log(1 - f_i) \quad (12)$$

s.t. if $y_i = 1$, $y_{i'} = 0$, and $i' \in D_i$, where D_i is the set of descendent nodes i . The optimization problem is solved in a similar manner as seen in [20]. Using the tree structure, we use dynamic programming to find the best and most efficient solution with the path consistency loss. In our bottom-up/top-down algorithm, for inference in the bottom-up step, the minimum energies for both possibilities (region selected/not selected) consistent with the path loss are propagated from the leaves to the root. Then, we choose valid detections by parsing through all the nodes from the root to the leaves.

In the bottom-up step, for every node i , a pair of energy sums is computed. The selected energy E_i^s for the i th node represents the cost of selecting the node i by setting its label=1 and the labels for all its descendants to 0. Similarly, the not selected energy E_i^{ns} for the i th node computes the cost of not selecting the node i by setting its label=0 and labeling its children optimally subject to the path loss. Let j and k be the children of the i th node. The energies are computed bottom-up in a recursive manner as

Input: A list of detection probability $\{f_i\}_i$ of each node r_i
Output: Energy pairs for selecting and not selecting a region $G_E = \{(E_i^s, E_i^{ns})\}_i$

- (1) $G_E \leftarrow \{\}$
- (2) **ComputeEnergy**(r_0), where r_0 is a root node.
- (3) /* Sub-function that recursively computes energy terms */
- (4) **Function ComputeEnergy** (r_i)
- (5) **if** r_i is a leaf node **then**
- (6) $E_i^s = -\log f_i$
- (7) $E_i^{ns} = -\log(1-f_i)$
- (8) **else**
- (9) $(E_k^s, E_k^{ns}) = \text{ComputeEnergy}(r_k)$
- (10) $(E_j^s, E_j^{ns}) = \text{ComputeEnergy}(r_j)$
- (11) $E_i^s = -\log f_i + \sum_{k \in D_i} E_k^{ns}$
- (12) $E_i^{ns} = -\log(1-f_i) + \min(E_k^s, E_k^{ns}) + \min(E_j^s, E_j^{ns})$
- (13) **end**
- (14) $G_E \leftarrow G_E \cup \{(E_i^s, E_i^{ns})\}$
- (15) **return**(E_i^s, E_i^{ns})

ALGORITHM 1: Bottom-up energy computation

Input: Energies for every region, $G_E = \{(E_i^s, E_i^{ns})\}_i$
Output: A complete label assignment $\mathbf{Y} = \{y_i\}_i$

- (1) $\mathbf{Y} \leftarrow \{\}$
- (2) **AssignNodeLabels**(r_0)
- (3) /* Sub-function that recursively assigns node labels */
- (4) **Function AssignNodeLabels** (r_i)
- (5) **if** $E_i^s < E_i^{ns}$ **then**
- (6) $\mathbf{Y} \leftarrow \mathbf{Y} \cup \{y_i = 1\} \cup \{y_{i'} = 0 \mid i' \in D_i\}$,
- (7) **else**
- (8) $\mathbf{Y} \leftarrow \mathbf{Y} \cup \{y_i = 0\}$
- (9) **AssignNodeLabels** (r_k)
- (10) **AssignNodeLabels** (r_j)
- (11) **end**

ALGORITHM 2: Top-down label assignment

$$\begin{aligned}
 E_i^s &= -\log f_i + \sum_{k \in D_i} -\log(1-f_k), \\
 E_i^{ns} &= -\log(1-f_i) + \min(E_k^s, E_k^{ns}) + \min(E_j^s, E_j^{ns}),
 \end{aligned} \tag{13}$$

where D_i is a set of indices of descendants of the i th node. For leaf nodes, we assign $E_i^s = -\log(f_i)$ and $E_i^{ns} = -\log(1-f_i)$. Algorithm 1 gives the pseudocode of the bottom-up algorithm.

After computing the energies for all nodes, we label the regions in a top-down manner. We start from the root node and compare the selected energy E_i^s and the not selected energy E_i^{ns} . If the selected energy is lower than the not selected, then, we select the node i by setting its label = 1 and the label for all its descendants to 0. Otherwise, we assign a label = 0 to the i th node and search its subtrees. Algorithm 2 gives the pseudocode of the top-down algorithm. Finally, the nodes with a label = 1 are seen as true detections to give an optimal solution.

We demonstrate the results of using the greedy approach versus the bottom-up/top-down algorithm for inference with our trained classifier for one of the datasets used in our experiments in Table 1. The choice of the methodology of inference does not contribute much to the overall detection scores. Hence, we can choose either of the schemes. We choose the bottom-up/top-down algorithm for our inference.

4. Experiments

We evaluate the performance of our detection framework using minimal training data under various conditions to show the individual contribution of the different parts of the algorithm. We also compare the performance of our method in detail to one of the methods that use dot annotations for training their detection classifier.

4.1. Datasets. We experiment with four datasets for cell detection. The datasets encompass a wide variety of cells with different shape and intensity characteristics:

- (1) Phase-contrast images of cervical cancer hela cells from [1]. The dataset contains 11 testing images and 11 training images of size 400×400 .
- (2) Synthetic cells from [41]. The dataset contains 20 images. We used 10 images for testing and 10 images for training. The images are of size 950×950 .
- (3) *Drosophila melanogaster* Kc167 cells from [41]. The dataset contains 14 images. We used 9 images for testing and 5 images for training. The images are of size 950×950 .
- (4) Bright-field images of fission yeast cells from [23, 42]. We did not have access to their complete data. We could obtain only 9 images of size 1024×1024 .

TABLE 1: Comparison of inference methods for phase-contrast hela cells. SS + ILASTIK refers to our semisupervised method using pixel probability maps from ILASTIK. The unlabeled data has contributions from both the training and testing images.

Number of subimages	SS + ILASTIK (greedy inference)			SS + ILASTIK (bottom-up/top-down inference)		
	Precision	Recall	<i>F</i> -score	Precision	Recall	<i>F</i> -score
1	0.9262	0.9460	0.9359 ± 0.0074	0.9266	0.9498	0.9381 ± 0.0078
3	0.9378	0.9579	0.9477 ± 0.0068	0.9397	0.9580	0.9488 ± 0.0064
5	0.9495	0.9605	0.9549 ± 0.0065	0.9498	0.9599	0.9548 ± 0.0066
7	0.9510	0.9625	0.9567 ± 0.0081	0.9513	0.9620	0.9566 ± 0.0076
9	0.9526	0.9659	0.9592 ± 0.0052	0.9531	0.9661	0.9596 ± 0.0050

TABLE 2: Quantitative comparison of detection for different datasets using our semisupervised approach using ILASTIK as a pixel classifier for varying number of subimages. Left-right: number of subimages, results using [1], SS + ILASTIK (unlabeled data has contributions from only the training images), and SS + ILASTIK (unlabeled data has contributions from both the training and testing images).

(a) Phase-contrast hela cells

Number	Prec	Arteta et al. [1]		SS + ILASTIK (training)			SS + ILASTIK (training + testing)		
		Rec	<i>F</i> -score	Prec	Rec	<i>F</i> -score	Prec	Rec	<i>F</i> -score
1	0.9065	0.9253	0.9158 ± 0.0203	0.9180	0.9351	0.9264 ± 0.0082	0.9266	0.9498	0.9381 ± 0.0078
3	0.9342	0.9497	0.9419 ± 0.0084	0.9360	0.9511	0.9434 ± 0.0057	0.9397	0.9580	0.9488 ± 0.0064
5	0.9478	0.9545	0.9511 ± 0.0092	0.9480	0.9568	0.9523 ± 0.0061	0.9498	0.9599	0.9548 ± 0.0066
7	0.9481	0.9594	0.9537 ± 0.0091	0.9500	0.9596	0.9547 ± 0.0091	0.9513	0.9620	0.9566 ± 0.0076
9	0.9536	0.9603	0.9570 ± 0.0055	0.9528	0.9632	0.9579 ± 0.0080	0.9531	0.9661	0.9596 ± 0.0050

(b) Synthetic fluorescence cell images

Number	Prec	Arteta et al. [1]		SS + ILASTIK (training)			SS + ILASTIK (training + testing)		
		Rec	<i>F</i> -score	Prec	Rec	<i>F</i> -score	Prec	Rec	<i>F</i> -score
1	0.9695	0.9527	0.9620 ± 0.0106	0.9730	0.9611	0.9670 ± 0.0034	0.9791	0.9698	0.9744 ± 0.0026
3	0.9716	0.9626	0.9671 ± 0.0086	0.9755	0.9662	0.9708 ± 0.0015	0.9801	0.9712	0.9756 ± 0.0005
5	0.9787	0.9643	0.9714 ± 0.0051	0.9794	0.9689	0.9741 ± 0.0013	0.9806	0.9723	0.9764 ± 0.0003
7	0.9792	0.9651	0.9721 ± 0.0036	0.9800	0.9699	0.9749 ± 0.0009	0.9811	0.9732	0.9771 ± 0.0006
9	0.9808	0.9652	0.9730 ± 0.0009	0.9806	0.9701	0.9753 ± 0.0010	0.9813	0.9750	0.9789 ± 0.0004

(c) *Drosophila* Kc167 cells

Number	Prec	Arteta et al. [1]		SS + ILASTIK (training)			SS + ILASTIK (training + testing)		
		Rec	<i>F</i> -score	Prec	Rec	<i>F</i> -score	Prec	Rec	<i>F</i> -score
1	0.8195	0.9275	0.8702 ± 0.0156	0.8254	0.9306	0.8748 ± 0.0069	0.8311	0.9496	0.8864 ± 0.0043
3	0.8287	0.9337	0.8781 ± 0.0145	0.8358	0.9416	0.8855 ± 0.0082	0.8402	0.9566	0.8946 ± 0.0056
5	0.8402	0.9380	0.8864 ± 0.0083	0.8433	0.9452	0.8913 ± 0.0093	0.8451	0.9568	0.8975 ± 0.0080
7	0.8427	0.9435	0.8903 ± 0.0055	0.8463	0.9475	0.8940 ± 0.0070	0.8499	0.9580	0.9007 ± 0.0032
9	0.8516	0.9456	0.8962 ± 0.0045	0.8519	0.9497	0.8981 ± 0.0052	0.8526	0.9590	0.9028 ± 0.0038

(d) Fission yeast cells

Number	Prec	Arteta et al. [1]		SS + ILASTIK (training)			SS + ILASTIK (training + testing)		
		Rec	<i>F</i> -score	Prec	Rec	<i>F</i> -score	Prec	Rec	<i>F</i> -score
1	0.7492	0.9238	0.8274 ± 0.0136	0.8064	0.9230	0.8607 ± 0.0074	0.8528	0.9243	0.8871 ± 0.0053
3	0.7602	0.9342	0.8383 ± 0.0112	0.8145	0.9276	0.8673 ± 0.0089	0.8602	0.9245	0.8912 ± 0.0050
5	0.7622	0.9412	0.8423 ± 0.0076	0.8226	0.9414	0.8779 ± 0.0076	0.8707	0.9409	0.9044 ± 0.0051
7	0.7645	0.9494	0.8470 ± 0.0039	0.8258	0.9465	0.8820 ± 0.0062	0.8736	0.9420	0.9065 ± 0.0039
9	0.7718	0.9460	0.8501 ± 0.0046	0.8287	0.9460	0.8834 ± 0.0058	0.8791	0.9470	0.9118 ± 0.0047

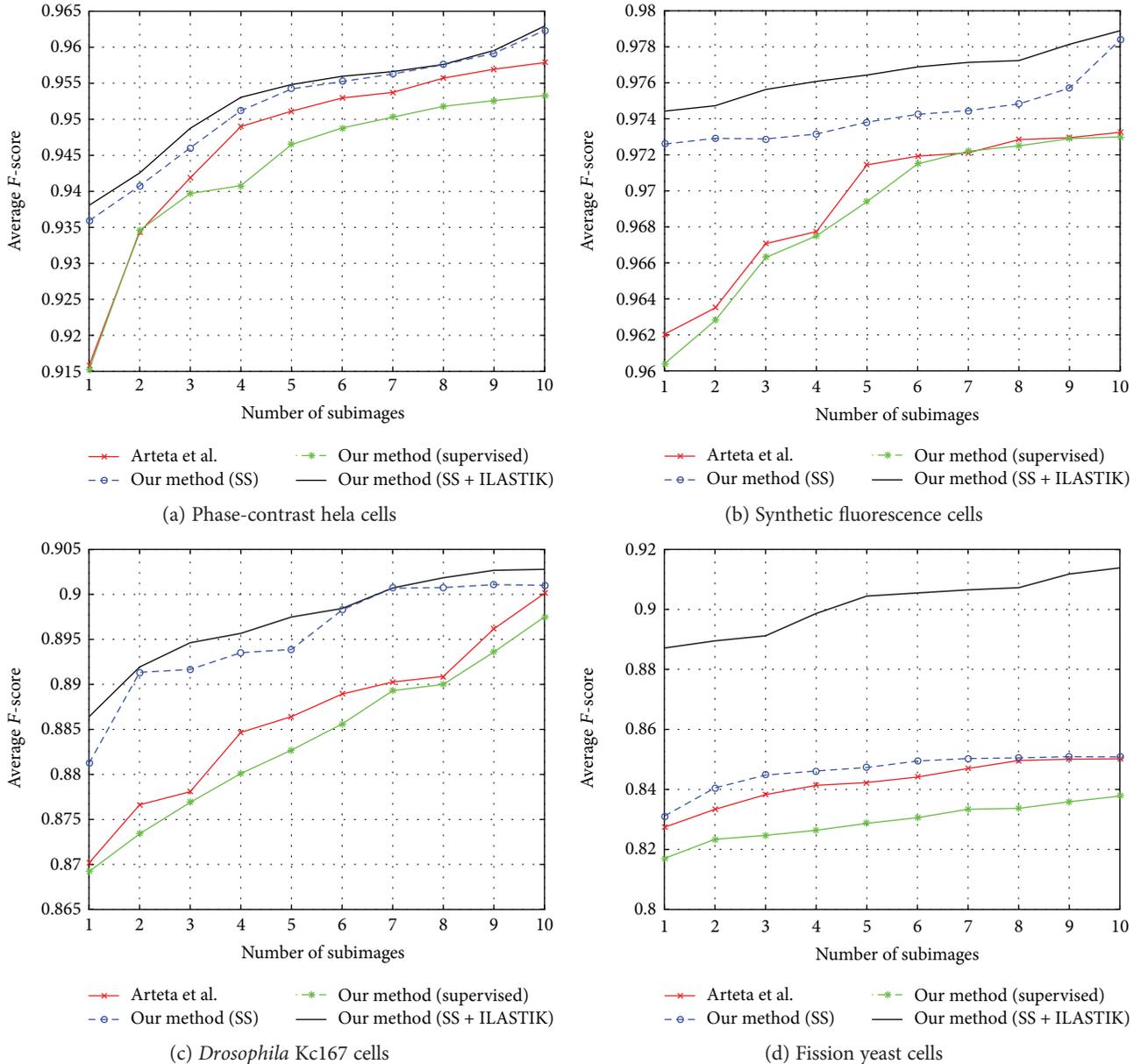


FIGURE 2: Average F -score for varying size of training subimages for our method with only supervised loss (supervised), our semisupervised framework (SS), and our semisupervised framework using ILASTIK as the pixel classifier (SS + ILASTIK) is shown for all the testing datasets. The unlabeled data has contributions from both the training and testing images. We compare the performance of our method with that of Arteta et al. [1].

We used only 2 images for training and 7 images for testing.

4.2. Implementation Details. For our detection experiments, we do not use the entire labeled training data to learn a model. The probability maps for the images are learned using ILASTIK trained with dot annotations from only two images. The number of cells in each image varies with different datasets. We work with subimages from the training dataset to reduce the manual annotations needed. For every dataset, we create a pool of subimages from the training dataset that contains few cells with their centers

annotated. We randomly choose subimages from this pool to train our classifier. The first step is to adapt the training dataset to our algorithm. Since we have only the centroid annotations for the cells, we need to generate ground truth information for every region generated by the MSER detector, which states if the region is a true cell. This can be done by checking all possible regions in a path and selecting the largest regions with a single annotation. The true labels (X_s, y_s) now can be used for the supervised learning in our framework. We have tested the performance of our algorithm when only the training dataset (labeled and unlabeled) is used as well as when we include the testing dataset (unlabeled) along with the training

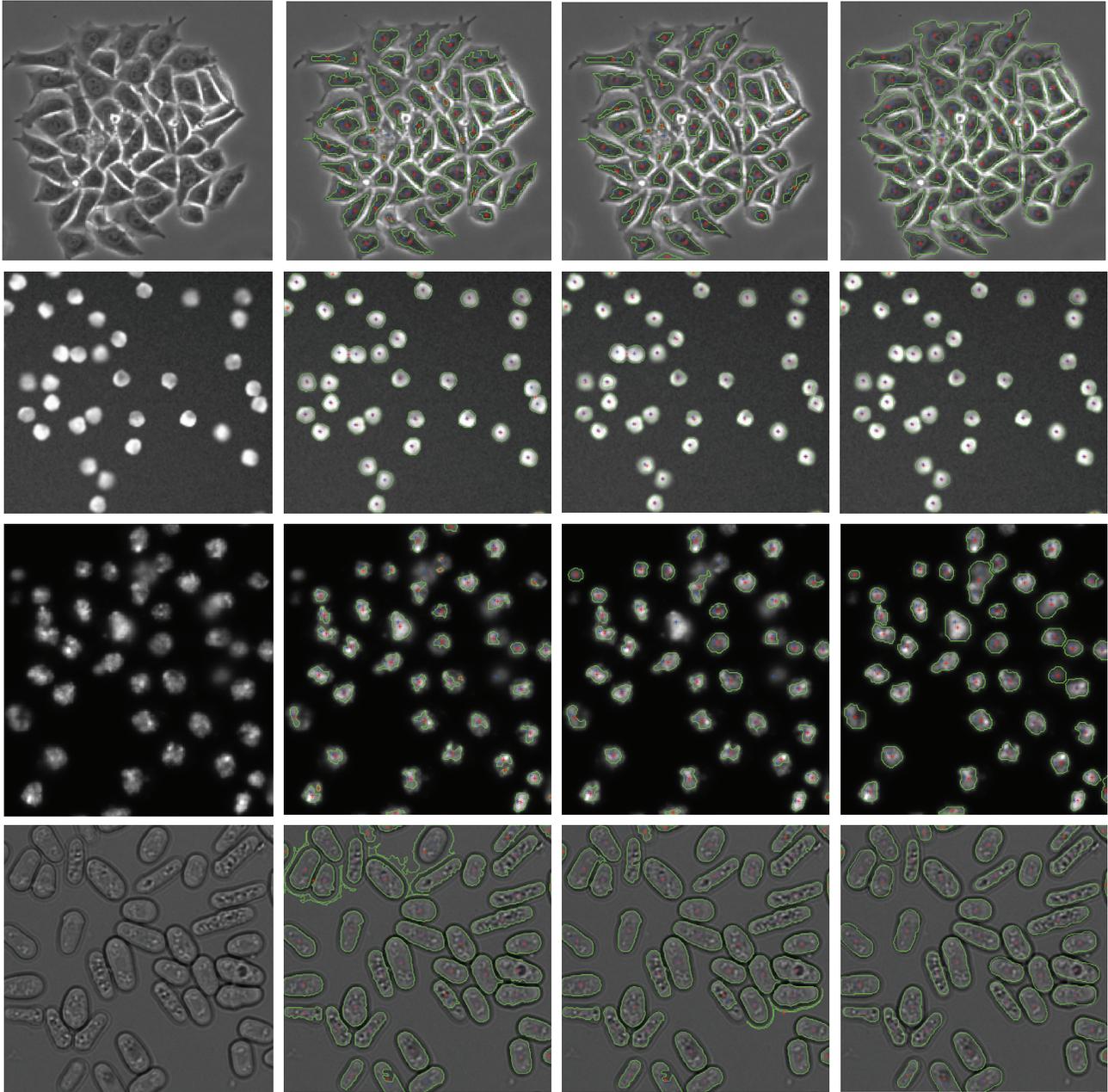


FIGURE 3: Visual representation of cell boundaries for the different datasets (top-bottom: phase-contrast hela cells, synthetic fluorescence cells, *Drosophila* Kc167 cells, and fission yeast cells). The cell boundaries are shown in green, the predicted center is in red, and the true centroids are in blue. Left-right: input image, results—Arteta et al. [1], results—semisupervised framework (SS), and results—semisupervised framework with pixel classifier (SS + ILASTIK). The unlabeled data has contributions from both the training and testing images.

dataset (labeled and unlabeled) to formulate the unsupervised path loss. The weights w we learned using the supervised and unsupervised loss are used to evaluate the testing dataset. The candidate regions and their features extracted from the testing images are scored using the learned weights. The label assignment process for the regions in the testing images is based on the inference procedure described in Section 3.5. It takes approximately 48.76 seconds for supervised and 96.59 seconds for semisupervised training for the phase-contrast hela dataset with 10 training images and 10 testing images in MATLAB

using 3.5 GHz Intel CPU. We also evaluate the DICE and detection scores for the phase-contrast hela cells when the entire training dataset is used to contribute to the supervised loss. We report results for both cases when we include and exclude the testing dataset in the unsupervised path loss term. We compare our detection and DICE scores with those of other methods.

4.3. Evaluation Metrics. We evaluate the performance of our algorithm by computing the detection scores. We find corresponding matching pairs from the pool of ground truth

TABLE 3: Comparison of DICE and detection scores for phase-contrast hela cells. SS refers to our semisupervised approach when the cell candidates are generated using the MSER detector on the intensity images. SS + ILASTIK refers to our semisupervised approach using pixel probability maps from ILASTIK to generate cell candidates using the MSER detector. We evaluate our experiments with and without the contributions of the unlabeled data from the testing dataset.

Method	Detection scores	DICE
Arteta et al. [1]	0.90	0.81
Funke et al. [23]	0.97	0.84
Zhang et al. [21]	0.91	0.80
Our method (SS (only training dataset))	0.94	0.80
Our method (SS (training + testing))	0.95	0.82
Our method (SS + ILASTIK (only training dataset))	0.96	0.86
Our method (SS + ILASTIK (training + testing))	0.97	0.86

regions and predicted regions using overlapping areas. Similar to [1], we use a Hungarian matching algorithm to find the corresponding areas, by minimizing the distance between their centroids. The oversegmented regions or regions with no correspondences with the ground truth segmentations are labeled as false positives (FP). The unmatched ground truth regions/missed detections are accounted for as false negatives. We evaluate the performance of our algorithm using precision, recall, and F -score. Since the ground truth configurations for the regions are generated from the centroids, we do not have true segmentation masks for the cells of all datasets. Hence, we compute the segmentation accuracy using the DICE coefficient for only one of the datasets, for which we had masks.

4.4. Results. We compare our algorithm to [1] since the authors also use centroids of cells as user annotations. From all the training images, we generate subimages that are approximately 1/8th the size of the original image. We randomly pick a subset of subimages generated from the training pool and use it to train our supervised loss. Firstly, we evaluate our algorithm using only our supervised loss (our method (supervised)). Next, we train the classifier in our semisupervised framework (our method (SS)). In our semisupervised framework (our method (SS)), we generate the cell candidates using the MSER detector on intensity images. The MSER works well as a region detector only when there is high contrast between the pixels that are inside the cell and those of the background. In cases where the contrast is low, we cannot use the MSER detector on intensity images directly. We can use it on a transformed map of the intensity image in which the contrast is better. Hence, we use the MSER detector on the output of the pixel classifier (ILASTIK). This benefits low-contrast images such as in dataset 4. Finally, we evaluate our algorithm when we generate the cell candidates using the MSER detector on the pixel probability maps trained using ILASTIK (our method (SS + ILASTIK)). We repeat each experiment 20 times to account for the random selection of subimages for the supervised

data. Our experiment using ILASTIK (SS + ILASTIK) has been performed in two scenarios: firstly, when the contributions to the unlabeled data are only from the training images and secondly, when both the training and testing datasets contribute to the unlabeled data. We report the mean and standard deviation of the F -score, mean precision, and recall on the four datasets for both cases in Table 2. We observe that with significantly fewer supervised samples, the classifier benefits from more unlabeled samples. Hence, if we have limited training images, we would benefit by using the unlabeled samples from the testing images. We see that in the case of datasets 3 and 4, we have significantly fewer training images, and hence, using the unlabeled data from the testing dataset improves the performance considerably. Alternatively, in datasets 1 and 2, where we have more unlabeled data from the training images, we see a marginal improvement when we use the testing dataset. We plot the variation in the F -score for varying numbers of training subimages in Figure 2. We can clearly see that with the addition of the unsupervised loss, our classifier is able to give better detection scores. We have included the results of using only the supervised loss in our framework to highlight the advantage of using the unlabeled samples.

Our proposed method achieves better accuracy than that of [1] for all four datasets (1, 2, 3, and 4). A visual inspection of our detection with cell boundaries can be seen in Figure 3. We computed the DICE and detection scores for the phase-contrast hela cells using all the labeled images from the training data along with our unsupervised path loss term (Table 3). We observe that our detection scores are comparable to those of the other methods, including [23], which is the state-of-the-art, but we get a better DICE score.

5. Conclusion

We proposed a generic semisupervised framework for cell detection that relies on minimal training data. Since our algorithm works with fewer annotations, as the subimages used for training have fewer number of pixels, it reduces the effort required for manually annotating the entire image. The tree framework generated using the MSER detection proves to be very efficient in enforcing our unsupervised loss. The MSER tree framework also limits the search space by considering only regions of desired sizes. By using an ANN as a classifier, we are able to learn complex functions using the supervised and unsupervised loss. We learn the weights for our model by minimizing our cost criterion without the need for parameter tuning. In this paper, we have evaluated our framework only on biological data. Furthermore, we would like to extend this work for time-lapse microscopy images, where we can learn classifiers to predict cell lineages in a semisupervised framework.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman, "Learning to detect cells using non-overlapping extremal regions," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI): 15th International Conference, Proceedings, Part I*, pp. 348–356, Nice, France, October 2012.
- [2] O. Chapelle, B. Scholkopf, and A. Zien, *Semi-Supervised Learning*, The MIT Press, 1st edition, 2010.
- [3] X. Zhu, "Semi-supervised learning literature survey," in *Computer Sciences Technical Report 1530*, University of Wisconsin, Madison, 2006.
- [4] T. Liu, M. Zhang, M. Javanmardi, N. Ramesh, and T. Tasdizen, "SSHMT: semi-supervised hierarchical merge tree for electron microscopy image segmentation," in *14th European Conference on Computer Vision, Proceedings, Part I*, pp. 144–159, Amsterdam, Netherlands, October 2016.
- [5] C. Sommer, C. Straehle, U. Koethe, and F. A. Hamprecht, "Ilastik: interactive learning and segmentation toolkit," in *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 230–233, Chicago, IL, March 2011.
- [6] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *British Machine Vision Conference*, pp. 36.1–36.10, Cardiff, UK, September 2002.
- [7] T. Kanade, Z. Yin, R. Bise et al., "Cell image analysis: algorithms, system and applications," in *IEEE Workshop on Applications of Computer Vision*, pp. 374–381, Kona, HI, January 2011.
- [8] Q. Wu, F. A. Merchant, and K. R. Castleman, *Microscope Image Processing*, Academic Press, Burlington MA, 2008.
- [9] E. Meijering, "Cell segmentation: 50 years down the road [life sciences]," *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 140–145, 2012.
- [10] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.
- [11] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 207, no. 1167, pp. 187–217, 1980.
- [12] Y. Al-Kofahi, W. Lassoued, W. Lee, and B. Roysam, "Improved automatic detection and segmentation of cell nuclei in histopathology images," *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 4, pp. 841–852, 2010.
- [13] J. Byun, M. R. Verardo, B. Sumengen, G. P. Lewis, B. S. Manjunath, and S. K. Fisher, "Automated tool for the detection of cell nuclei in digital microscopic images: application to retinal images," *Molecular Vision*, vol. 12, no. 105–07, pp. 949–960, 2006.
- [14] D. Anoraganingrum, "Cell segmentation with median filter and mathematical morphology operation," in *Proceedings 10th International Conference on Image Analysis and Processing*, pp. 1043–1046, Venice, September 1999.
- [15] C. Di Ruberto, A. Dempster, S. Khan, and B. Jarra, "Segmentation of blood images using morphological operators," in *International Conference on Pattern Recognition*, vol. 3, pp. 397–400, Barcelona, September 2000.
- [16] N. Ramesh, M. E. Salama, and T. Tasdizen, "Segmentation of haematopoietic cells in bone marrow using circle detection and splitting techniques," in *9th IEEE International Symposium on Biomedical Imaging*, pp. 206–209, Barcelona, May 2012.
- [17] R. Adams and L. Bischof, "Seeded region growing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 641–647, 1994.
- [18] S. Beucher and C. Lantuejoul, "Use of watersheds in contour detection," in *International Workshop on Image Processing: Real-Time Edge and Motion Detection/Estimation*, Rennes, France, 1979.
- [19] J. Yarkony, C. Zhang, and C. Fowlkes, "Hierarchical planar correlation clustering for cell segmentation," in *Energy Minimization Methods in Computer Vision and Pattern Recognition: 10th International Conference*, pp. 492–504, Hong Kong, China, January 2015.
- [20] T. Liu, M. Seyedhosseini, and T. Tasdizen, "Image segmentation using hierarchical merge tree," *IEEE Transactions on Image Processing*, vol. 25, no. 10, pp. 4596–4607, 2016.
- [21] C. Zhang, J. Yarkony, and F. Hamprecht, "Cell detection and segmentation using correlation clustering," in *Medical Image Computing and Computer-Assisted Intervention: 17th International Conference, Proceedings, Part I*, pp. 9–16, Boston, MA, USA, September 2014.
- [22] S. Belongie, C. Carson, H. Greenspan, and J. Malik, "Color and texture-based image segmentation using EM and its application to content-based image retrieval," in *Sixth International Conference on Computer Vision*, pp. 675–682, Bombay, January 1998.
- [23] J. Funke, F. A. Hamprecht, and C. Zhang, "Learning to segment: training hierarchical segmentation under a topological loss," in *Medical Image Computing and Computer-Assisted Intervention: 18th International Conference, Proceedings, Part III*, pp. 268–275, Munich, Germany, October 2015.
- [24] L. Fujun, X. Fuyong, Z. Zizhao, M. Mason, and Y. Lin, "Robust muscle cell quantification using structured edge detection and hierarchical segmentation," in *Medical Image Computing and Computer-Assisted Intervention: 18th International Conference, Proceedings, Part III*, pp. 324–331, Munich, Germany, October 2015.
- [25] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2011.
- [26] P. Arbelaez, "Boundary extraction in natural images using ultrametric contour maps," in *Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, New York, NY, p. 182, June 2006.
- [27] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.
- [28] F. Amat, W. Lemon, D. P. Mossing et al., "Fast accurate reconstruction of cell lineages from large-scale fluorescence microscopy data," *Nature Methods*, vol. 11, no. 9, pp. 951–958, 2014.
- [29] M. G. Uzunbaş, C. Chen, and D. Metaxas, "Optree: a learning-based adaptive watershed algorithm for neuron segmentation," in *Medical Image Computing and Computer-Assisted Intervention: 17th International Conference, Proceedings, Part I*, pp. 97–105, Boston, MA, USA, September 2014.
- [30] F. Liu, F. Xing, and L. Yang, "Robust muscle cell segmentation using region selection with dynamic programming," in

- International Symposium on Biomedical Imaging*, pp. 521–524, Beijing, April 2014.
- [31] T. Liu, E. Jurrus, M. Seyedhosseini, M. Ellisman, and T. Tasdizen, “Watershed merge tree classification for electron microscopy image segmentation,” in *International Conference on Pattern Recognition*, pp. 133–137, Tsukuba, November 2012.
- [32] T. Liu, C. Jones, M. Seyedhosseini, and T. Tasdizen, “A modular hierarchical approach to 3D electron microscopy image segmentation,” *Journal of Neuroscience Methods*, vol. 226, pp. 88–102, 2014.
- [33] J. Funke, B. Andres, F. Hamprecht, A. Cardona, and M. Cook, “Efficient automatic 3D-reconstruction of branching neurons from EM data,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1004–1011, Providence, RI, June 2012.
- [34] J. Shi and J. Malik, “Normalized cuts and image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.
- [35] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Siisstrunk, “SLIC superpixels compared to state-of-the-art superpixel methods,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.
- [36] O. Veksler, Y. Boykov, and P. Mehrani, “Superpixels and supervoxels in an energy optimization framework,” in *11th European Conference on Computer Vision, Heraklion, Proceedings, Part V*, pp. 211–224, Crete, Greece, September 2010.
- [37] S. Nath, K. Palaniappan, and F. Bunyak, “Cell segmentation using coupled level sets and graph-vertex coloring,” in *Medical Image Computing and Computer-Assisted Intervention: 9th International Conference, Proceedings, Part I*, pp. 101–108, Copenhagen, Denmark, October 2006.
- [38] N. Ramesh, F. Mesadi, M. Cetin, and T. Tasdizen, “Disjunctive normal shape models,” in *IEEE 12th International Symposium on Biomedical Imaging*, pp. 1535–1539, New York, NY, April 2015.
- [39] O. Ronneberger, P. Fischer, and T. Brox, “U-net: convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention: 18th International Conference, Proceedings, Part III*, pp. 234–241, Munich, Germany, October 2015.
- [40] W. Xie, J. A. Noble, and A. Zisserman, “Microscopy cell counting with fully convolutional regression networks,” *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, pp. 1–10, 2016.
- [41] “Source for datasets,” <https://www.broadinstitute.org/bbbc/>.
- [42] J. Peng, Y. Chen, M. Green, S. Sabatinos, S. Forsburg, and C. Hsu, “Pombex: robust cell segmentation for fission yeast transillumination images,” *PloS One*, vol. 8, no. 12, article e81434, 2013.

Research Article

Low-Grade Glioma Segmentation Based on CNN with Fully Connected CRF

Zeju Li,¹ Yuanyuan Wang,^{1,2} Jinhua Yu,^{1,2} Zhifeng Shi,³ Yi Guo,^{1,2} Liang Chen,³ and Ying Mao³

¹Department of Electronic Engineering, Fudan University, Shanghai, China

²Key Laboratory of Medical Imaging Computing and Computer Assisted Intervention of Shanghai, Shanghai, China

³Department of Neurosurgery, Huashan Hospital, Fudan University, Shanghai, China

Correspondence should be addressed to Yuanyuan Wang; yywang@fudan.edu.cn and Jinhua Yu; jhyu@fudan.edu.cn

Received 4 November 2016; Revised 21 February 2017; Accepted 20 March 2017; Published 13 June 2017

Academic Editor: Ashish Khare

Copyright © 2017 Zeju Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This work proposed a novel automatic three-dimensional (3D) magnetic resonance imaging (MRI) segmentation method which would be widely used in the clinical diagnosis of the most common and aggressive brain tumor, namely, glioma. The method combined a multipathway convolutional neural network (CNN) and fully connected conditional random field (CRF). Firstly, 3D information was introduced into the CNN which makes more accurate recognition of glioma with low contrast. Then, fully connected CRF was added as a postprocessing step which purposed more delicate delineation of glioma boundary. The method was applied to T2flair MRI images of 160 low-grade glioma patients. With 59 cases of data training and manual segmentation as the ground truth, the Dice similarity coefficient (DSC) of our method was 0.85 for the test set of 101 MRI images. The results of our method were better than those of another state-of-the-art CNN method, which gained the DSC of 0.76 for the same dataset. It proved that our method could produce better results for the segmentation of low-grade gliomas.

1. Introduction

Among all brain tumors, glioma is the most severe [1]. According to the World Health Organization (WHO) criteria [2], gliomas were categorized into four grades from level I to level IV according to tumor malignancy. Normally, grade III and grade IV gliomas are called as high-grade gliomas (HGG) and grade I and grade II as low-grade gliomas (LGG). LGG could be further classified into astrocytomas, oligodendroglioma, and oligodendrocytes astrocytoma based on pathological type.

Magnetic resonance imaging (MRI) is the most common imaging diagnostic technique in the clinical diagnosis of gliomas. With MRI images, the accurate segmentation of gliomas is one of the most crucial procedures in treatment planning and follow-up evaluations. However, manual labeling is very time-consuming, and it is difficult to adopt a unified standard for segmentation. Meanwhile, automatic segmentation is still hard to be achieved because of the diversity of gliomas in size,

shape, and location [3]. Several limits of medical images such as the intensity inhomogeneity and unexpected intensity ranges of tissues would also cause difficulty for automatic segmentation of glioma [4].

A large number of algorithms have been developed to complete the task of tumor segmentation. Many traditional segmentation methods were based on gray scale values, such as fuzzy clustering and region growing [5]. These methods would be likely to fail when processing nonenhanced tumor images. Another kind of popular methods was multiatlas segmentation, which was based on the correlation of the priori brain atlas and the medical images to be processed [6]. However, these methods are often problematic when the atlases and target images are obtained via different imaging protocols and the deformable registration is also considered as a difficult process.

Recently, several methods related with machine learning have been applied in brain tumor segmentation. Parisot et al. used the prior knowledge to classify the tumor first

and then used another graph to determine the class of each voxel [7]. Huang et al. utilized the sparseness of samples to build up a particular dictionary and used a softmax model to optimize the error reconstruction coefficients for different classes [8]. Random forests have been considered to be good at dealing with a great number of features to accomplish brain tumor segmentation. Meier et al. applied a set of dedicated features to get decision forests to discriminate pathological regions from brain MRI volumes [9]. In addition, Markov random field (MRF) and conditional random field (CRF) are also often mentioned to obtain smooth edges. Zhao et al. proposed a semisegmentation method based on the MRF [10], in which one slice was labeled and other slices were sequentially labeled based on a MRF label. Meier et al. estimated the CRF to improve the voxel-wise classification performance on the top of the decision forest classifier [11]. These conventional machine learning methods are often based on a large number of features extracted from the image, reflecting the shape, gray value, and texture of the tumor area. But an important problem with these approaches is that the computation of too many features is too time-consuming and particular feature can cause difficulties in tuning.

Another kind of approach to segment gliomas is based on the well-known convolutional neural network (CNN). Primarily due to its abilities to obtain image global and local information directly from the convolution kernels, CNNs have made breakthrough progress in image processing and object recognition and been widely used thereafter [12]. CNNs have shown good performances in the field of medical image processing in recent years, not only in terms of accuracy, but also in terms of efficiency [13]. Pereira et al. developed two CNN structures with different depths to deal with the HGG and the LGG [14]. Dvorak et al. evaluated the effectiveness of different patch selection strategies based on the segmentation results of CNNs [15]. Havaei et al. proposed a multiscale CNN structures in order to make better use of local and global information [16]. Rao et al. combined random forests with the final output of CNNs to achieve better classification results [17]. Several CNN methods mentioned previously are based on a two-dimensional convolution kernels and do not make good use of the natural three-dimensional (3D) information of medical images. Typically, 3D filters can take fully advantage of 3D connection characteristics of images. Kamnitsas et al. [18] evaluated the use of 3D filters. However, the 3D convolution algorithm limits the size of convolution kernels and causes a great increase of the computation load. Furthermore, 3D filters require high resolution on the vertical plane, while actual MRI images usually need interpolation and do not have such high resolution. The process of interpolation and down sampling in 3D filters often brings additional errors in segmentation. Therefore, how to make good use of 3D information with CNN in gliomas segmentation still remains an important problem.

On the other hand, segmentation methods mentioned above mainly focused on the segmentation of the HGG. Although the internal structure of the LGG is simpler than that of the HGG, the segmentation of the LGG is considered more difficult because of its lower contrast and smaller size [14]. Thus, these segmentation methods

TABLE 1: Patients characteristics.

Characteristics	Quantity	Percentage
Tumor grade		
Grade II	160	100.0%
Radiological diagnosis		
Astrocytoma	48	30.0%
Oligodendroglioma	25	15.6%
Oligodendrocytes astrocytoma	73	45.6%
Gender		
Male	81	50.6%
Female	79	49.4%
Category		
Training set	59	36.9%
Test set	101	63.1%

mentioned above often do not produce good results when dealing with the LGG.

As it is known, the lesion area of the LGG is more distinguishable from T2flair MRI than from other MRI modalities [19]. So in this study, we chose T2flair modal MRI images as the original data for image processing. LGG has high signal in the T2flair images. Compared with HGG, the signal intensity distribution of LGG is more uniform and the boundaries of tumors and surrounding brain tissues tend to be clearer. In addition, LGG usually shows no necrosis, perifocal edema, or hemorrhagic foci. Oligodendroglioma and astrocytoma were two major types of LGG. The two subtypes could be distinguished radiologically by the presence of calcification. Generally, calcification inside oligo tumor turns out hypointensity on T2-weighted and isointensity on T1-weighted pre-contrast MRI.

In this paper, a new method is presented aiming at automatic segmentation of LGG MRI images. Main contributions of the paper are as follows. Firstly, the effect of different CNN depths and the number of neurons in the fully connected layers on the segmentation result were thoroughly evaluated. Secondly, in order to use the 3D information, nearby slices were set into the network and connected with a fully connected CRF. Lastly, the results on the LGG T2flair dataset showed that the method is better than the state-of-the-art CNN method.

The rest of the paper is organized as follows: in Section 2, we present our materials and method flows. Experimental design, results, and discussion can be found in Section 3. Finally, the main conclusions are presented in Section 4.

2. Materials and Methods

2.1. Patients. All data from patients used in our study were obtained from Shanghai Huashan Hospital. These patients were diagnosed between July 2013 and March 2016, and MRI images of these patients were collected at the time of diagnosis without any treatment. All 160 cases of data are described in detail in Table 1. We randomly selected 59 of them as the training data and 101 as the test data. Manual

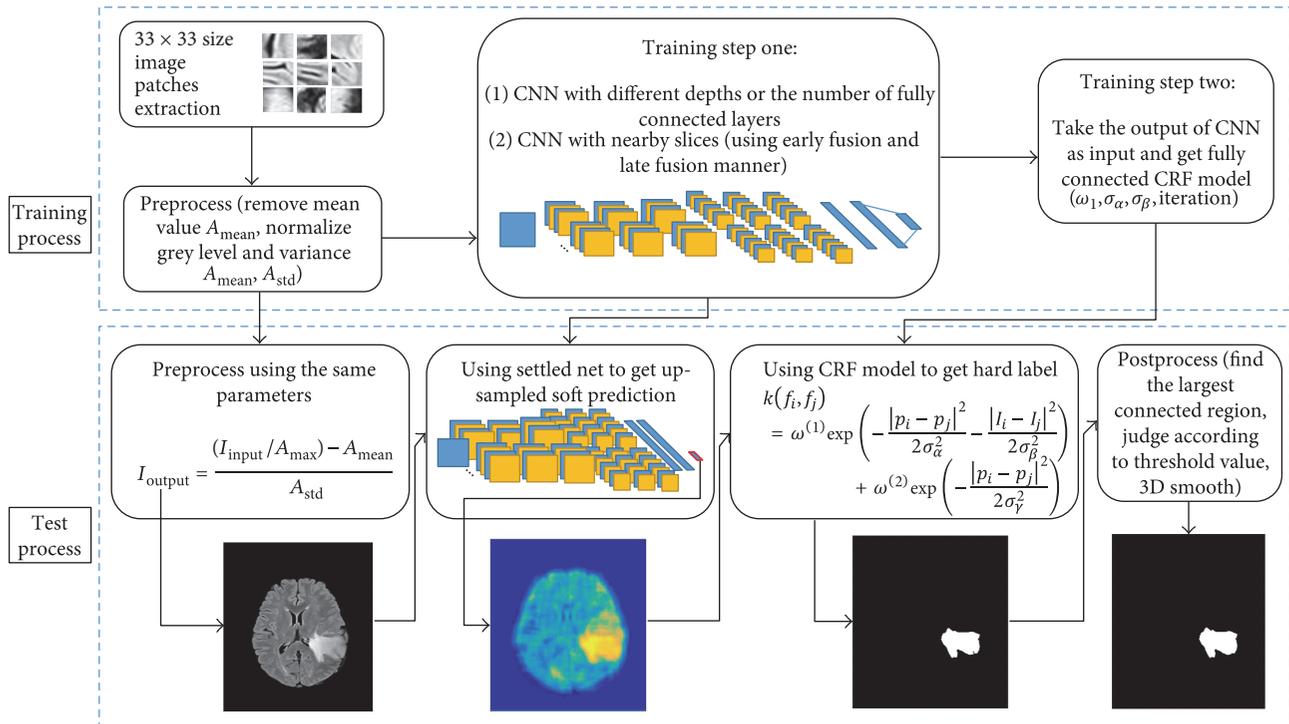


FIGURE 1: The flow chart of our method.

labeling of the tumor area was performed by two experienced neurosurgeons.

The size of the MRI images was 512×400 . MRI images were stored as 16-bit unsigned integer. All images were acquired according to the following parameters: pixel width = 0.47 mm, slice spacing = 2 mm, repetition time = 9000 ms, echo time = 99 ms, inversion time = 2501 ms, and flip angle = 150° .

2.2. Data Analysis. The original MRI data contains a lot of information which is not related to the segmentation problem, and these noises will greatly affect the tumor segmentation [3]. Some pretreatment was firstly operated to get rid of these noises.

BrainSuite is an open-source software and is able to automatically process the human brain medical images [20]. BrainSuite was utilized to remove the skull and scalp from MRI images and corrected nonuniformity problem of images.

2.3. CNNs for Tumor Segmentation. Figure 1 presents the overview of our approach, which is divided into several parts. The proposed method was also demonstrated in Algorithm 1. The further explanation will be given in the following sections.

2.3.1. The Preparation of the Input. The major idea of using CNN to segment gliomas is to take tumor segmentation as the problem of tumor recognition. Because the proportion of tumors in the brain is very small, some normal CNN network structures of image recognition could not get good segmentation results of gliomas [21]. Instead of utilizing the full-sized images in the training phase, CNN was trained

using patches randomly extracted from the images in this study. The conventional processing is to divide images into several patches during the training and set categories of center points as targets. The method was widely used in the medical image processing with unbalanced data [22]. More storage and more time were required by the training strategy using patches compared with training strategy using full-sized images. However, the former one is more suitable for tumor segmentation because the portion of the tumor regions is very small and the strategy using full-sized images would cause false positive results with unbalanced samples. The tumor region could be sampled more with the strategy using patches by picking more samples in the tumor regions on purpose.

In our study, we divided MRI images into 33 pixels \times 33 pixels size patches randomly at the training stage. Normally, about 50 patches were extracted from a single slice. The training set contained about 230,000 patches from 59 patients' brain images.

Taking into account the uneven distribution of image data, we chose an unbalanced selection method to obtain sufficient tumor samples. There were about 40% of samples in the channel containing tumors.

After the completion of sample extraction, we adopted several preprocessing methods. We removed the mean grey level of patches and normalized gray value and variance. It is worth being noted that these parameters are preserved for the test image to do the same processing.

2.3.2. CNN Base Line Structure. We selected one of the most advanced CNN structures as the base line. The selected CNN took part in the Brain Tumor Segmentation Challenge 2013 (BRATS 2013) and ranked first place in BRATS 2013 and

Input: image $\mathbf{I}_{\text{input}}$, mean value A_{mean} , max value A_{max} , standard deviation A_{std} , parameters of settled multi-way CNN network and parameters of fully connected CRF model ($\omega_1, \sigma_\alpha, \sigma_\beta$, **Iteration**)

Output: tumor segmentation result

1: Normalize $\mathbf{I}_{\text{input}}$ using $A_{\text{mean}}, A_{\text{max}}$ and A_{std} and get $\mathbf{I}_{\text{output}}$.

2: Enter $\mathbf{I}_{\text{output}}$ into the settled multi-way CNN, and get the output $\mathbf{P}(x_i)$ from the softmax layers

3: Using $\mathbf{P}(x_i)$ as original unary potential

4: **for Iteration do**

5: Compute fully connected CRF with $\mathbf{I}_{\text{output}}$ and unary potential using parameters $\omega_1, \sigma_\alpha, \sigma_\beta$

6: Correct the segmentation results using several morphology methods

ALGORITHM 1: Proposed segmentation method.

TABLE 2: Base line and improved CNN structure configurations.

Base line	CNN configuration		
	Base line + deeper	Base line + more fc	Base line + deeper + more fc
	Input		
Conv ¹ 3-64	Conv3-64	Conv3-64	Conv3-64
Conv3-64	Conv3-64	Conv3-64	Conv3-64
	Conv3-64		Conv3-64
	Max-pool		
Conv3-128	Conv3-128	Conv3-128	Conv3-128
Conv3-128	Conv3-128	Conv3-128	Conv3-128
	Conv3-128		Conv3-128
	Max-pool		
	FC ² -256		FC-4096
	FC-256		FC-4096
		FC-2	
		Softmax	
		Output	

¹Conv: followed by the size of convolutional kernels and the number of filter banks. ²FC: followed by the number of fully connected layers.

second place in BRATS 2015 [14]. The network consisted of seven convolution layers, and structures are briefly described as the base line in Table 2. The network utilized small and continuous convolution kernels to enhance tumor recognition ability of network without increasing computation.

Every convolution layer was followed by active layers, and the dropout layer was set after each fully connected layer. According to regular CNN strategy, we chose rectifier linear units (ReLU) as the function of active layers.

It should be mentioned that we put the entire image into the network in the phase of testing to reduce the processing time. For the purpose of getting the correct and accurate labeling map, we removed the dropout and softmax loss layers of the network in testing.

2.3.3. Building Deeper Networks. Some research results on the CNN showed that objects would be recognized better with increasing depth of the network structure and the number of neurons in the fully connected layers. Therefore, we conducted a number of contrast experiments with different depths and different number of neurons in the fully connected layers based on the base line. We referred to a well-behaved network structure at the time of designing the network structure [23]. Detailed information is also shown in Table 2.

2.3.4. Adding Near Slices into Networks. As previously mentioned, it is a tricky problem to input the 3D information into the network without increasing the computational load and bringing the complex registration process. This problem cannot be well solved by existing methods. In order to drop out a solution, two ideas in motion recognition from video processing were introduced into our CNN structure. Two new network architectures were shown in Figure 2, called early fusion and late fusion processes.

In the field of video processing, Karpathy et al. reported that early fusion structure obtained more accurate local motion recognition and late fusion compared the contents of each channel to obtain the better recognition of global motion [24]. Video processing and 3D brain MRI images have similar characteristics; although several frames nearby are not exactly the same, relative information could provide assistance to make more specific judgments. In fact, neighbor slices were always referred to get better differentiation of the tumor area at the time of manual labeling. Based on this fact, introducing these structures of combining 3D information could be helpful in segmenting tumor regions.

It is worth being mentioned that we utilized the network with a deeper structure and more neurons in the fully connected layers as a basis. The basic structure is referred to as “Base line + deeper + more fc” CNN structure in Table 2. As seen in the figure, early fusion mixes the information of three slices together at the beginning of the network and three relative slices shared the same network structure parameters. Each slice was utilized to train a one-way network in the structure of late fusion and finally connected together by the fully connected layers.

The connection of different slices in the network has many advantages. Firstly, introducing 3D information can make the network identify the tumor region more accurately. This may be of benefit to the CRF decision. Secondly, it is possible to effectively avoid missegmentation for similar tumor regions in other brain regions. Combining nearby slices can correct erroneous identification by the use of 3D information. Moreover, the structures combined with 3D information could help CNN recognize the upper and lower surfaces of the tumor better. The accurate identification of tumor regions would also bring benefits to the segmentation of small tumors.

2.3.5. Fully Connected CRF for Further Identification. Current CNN structures for the tumor segmentation [14–17] have several limitations due to the structure. First of all, the receptive field that corresponds to a single neuron of the last fully

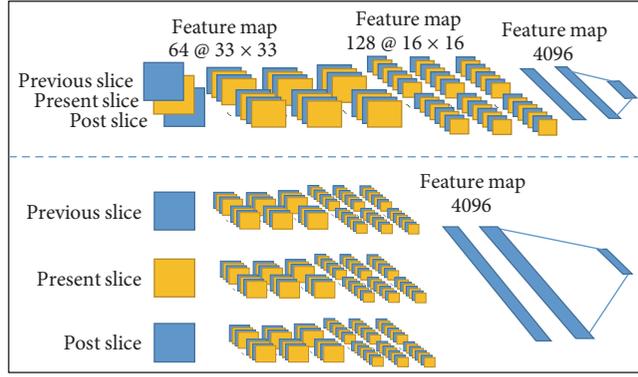


FIGURE 2: Two network structures which can make better use of 3D information. The implementation structure of early fusion is shown above, and the figure below is the diagram of late fusion.

connected layer is too large. Take our network structure for example, due to the presence of two pooling layers, one neuron corresponds to four pixels in images. What is more, the CNN was limited by the lack of space and edge constraints compared with other machine learning methods. So the result of the segmentation is rough on edges. Secondly, for MRI images, the contrast and the gray scale of images from different patients are different. In addition, the contrast of the LGG is often lower than that of the HGG, increasing the difficulty for accurate identification of whole tumors. Because of these reasons, sometimes the CNN can only recognize a part of the tumor area and sometimes the region near the tumor is misidentified.

A possible improvement of the CNN is to integrate the CRF into the network structure to refine the segmentation results [25]. CRF is a framework for building probabilistic models. Iterative parameter estimation algorithms were always applied for CRF [26]. Krähenbühl reduced the complexity of fully connected CRF computation from quadratic complexity to linear complexity through an optimization algorithm based on the mean field approximation [27]. On his basis, we used the network features as the contribution of each category and optimized the probability model in a wide domain. The energy function that we used is

$$E(x) = \sum_i \theta_u(x_i) + \sum_{i,j,i \neq j} \theta_p(x_i, x_j), \quad (1)$$

where $\theta_u(x_i)$ is the unary potential which is computed independently for each pixel. In our study, we utilized features extracted from the CNN to calculate this parameter.

$$\theta_u(x_i) = -\log P(x_i), \quad (2)$$

where $P(x_i)$ is the label assignment probability. We set the output of the last fully connected layer of the CNN as $P(x_i)$. It should be noted that, the bi-cubic method was used to interpolate the score map since the CNN output size is four times smaller than the image. The other parts of the energy function are the pairwise potentials, which reflect the relationship between any two pixels. Pairwise potentials were defined as

$$\theta_p(x_i, x_j) = \mu_p(x_i, x_j) \sum_{m=1}^K \omega^{(m)} k^{(m)}(f_i, f_j), \quad (3)$$

where $\mu_p(x_i, x_j)$ is a function that determines whether a point is the same or not. The Gaussian kernel $k^{(m)}$ was calculated as follows:

$$k(f_i, f_j) = \omega^{(1)} \exp\left(-\frac{|p_i - p_j|^2}{2\sigma_\alpha^2} - \frac{|I_i - I_j|^2}{2\sigma_\beta^2}\right) + \omega^{(2)} \exp\left(-\frac{|p_i - p_j|^2}{2\sigma_\gamma^2}\right), \quad (4)$$

where p_i and I_i represent the location and intensity of the corresponding pixel i , respectively. As seen from the formula, the first Gaussian kernel is related to the location and intensity of a particular pixel, which is the main convergence factor of the graph model. However, the second kernel is only related to the location of the pixel, which makes the graph model smoother. In our experiments, the iterations of the CRF algorithm also had a great influence on the segmentation result. These parameters of the CRF should be obtained through the second training phase of the training set.

2.3.6. Postprocessing. After obtaining the segmentation results of the CRF output, the segmentation results were corrected using several morphology methods. We chose the largest connectivity region of each slice as the candidate region. Then regions containing the area greater than 300 pixels were selected as identified tumors. After yielding the segmentation results for each slice, 3D smoothing of the tumor was performed by box filter with 3-dimensional convolution kernel to make segmentation results more natural.

3. Results and Discussion

3.1. Experimental Setup and Evaluation. All of our experiments were built on top of MatConvNet [28], a MATLAB toolbox implementing CNNs for computer vision applications.

In order to make experimental results more contrastive, we used the same parameters when models were trained. Network parameters were initialized by using the improved Xavier method [29], and the echo setting for training was 25. The initial learning rate was 3×10^{-3} and declined to the final 3×10^{-5} using the logarithmic descent method.

In the process of training the fully connected CRF parameters, we used a relatively simple genetic algorithm to specify the optimal parameters. As shown in Table 3, all the parameters were discretized. ω_1 and σ_y were considered unimportant and were set to 5 and 5, separately. At first, we generated a set of parameters randomly and then changed four parameters one by one in order to optimize the selection, retaining the best one. Thus, 27 cycles were required to estimate the parameters of the fully connected CRF model. The parameters corresponding to the best overall segmentation results of the 59 training data were chosen as the optimized parameters during the second training phase.

In order to evaluate the accuracy of the segmentation results, we used three indices as usual [3] Dice similarity coefficient (DSC) [30], positive predictive value (PPV), and sensitivity. The DSC evaluated the segmentation results globally by calculating the overlapping parts, which often serve as an indicator of the overall outcome of segmentation results.

$$\text{DSC} = \frac{2\text{TP}}{\text{FP} + 2\text{TP} + \text{FN}}, \quad (5)$$

where TP, FP, and FN represent the regions of the true positive, the false positive, and the false negative, respectively. The PPV calculated the accuracy of the segmentation result. The higher the coefficient is, the less the nontumor area can be covered by the segmentation results.

$$\text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (6)$$

The sensitivity reflects the sensitivity of the algorithm to the tumor area. The higher the sensitivity, the more regions of the tumor that are included in the segmentation results.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (7)$$

3.2. Results for Different Network Depths. Four experiments were designed with different CNN depths and architectures. The design of the network was mentioned in Table 2. The convergence of the objective function and the error rate during the training is shown in Figure 3.

We can see from the figure that increasing the depth of the network and the number of neurons in the fully connected layers can make the network stabilize more quickly and gain better results. Moreover, deepening the network structure and increasing the number of neurons in the fully connected layers at the same time could further enhance the training results.

The segmentation results of training set and test set were evaluated, as shown in Tables 4 and 5. Quantitative results showed that making network deeper can improve the segmentation results without significantly increasing network parameters. When the network is deep enough, the increase

TABLE 3: The optional parameters selected for the fully connected CRF (the first and the last digits represent the selected minimum and maximum values, and the middle number represents the chosen step size, expressed in MATLAB representation).

	ω_1	σ_α	σ_β	Iterations
Selection range	5:5:10	20:5:70	3:1:10	5:1:10
The number of optional parameters	2	11	8	6
Optimized parameters	5	25	10	7

of the number of neurons in the fully connected layers would also improve the tumor recognition ability of the network.

All 160 cases of tumor segmentation results were presented in Figure 4. It can be seen that the tumor recognition ability of complex network was better than the ones with simple network structure. Nevertheless, the improved network can also better handle some special cases.

The effect of changing network structure on the segmentation results can be better reflected in Figure 5. We showed three examples of segmentation in Figure 5, and it can be seen that the segmentation results became more detailed and accurate when the network was deepened or the number of neurons in the fully connected layers was increased.

The results were also consistent with the conclusion in the previous quantitative analysis. The promotion of the PPV was the most remarkable, and the nontumor region covered by segmentation results was decreased.

We also looked into the difference in the network architecture by analyzing network features values. Representative features in the network from different depths were presented in Figure 6. As some of the research in computer vision reported [31], the lower layers represent the information such as edges, contours, and intensities. With the deepening of the network, the characteristics become more different with the input images. Besides, the regions from different classes become more and more obvious, especially after the fully connected layers. It can be seen that networks with more neurons in the fully connected layers can better identify tumor regions at the same depth. Because the increase of neurons in the fully connected layers gives more choices of fitting, the lower layers have the opportunity to establish better parameters.

The effect of increasing the depth of the network can be seen from the diagram in the last column. With the deepening of the network, image recognition is significantly more detailed and the resolution of pixel recognition is higher. Fine identification of images can benefit the reduction of recognition errors. This may be because the increase in the number of convolution layers leads to multiple processing of the same pixel point.

By doing so, the CNN could effectively reduce the adverse impact of surrounding pixels so that recognition is more precise.

3.3. Results for Adding the Near Slices and Fully Connected CRF. Similarly, we presented the training parameters of the network structure after adding the neighboring slice information in Figure 3. By comparison, it can be seen from

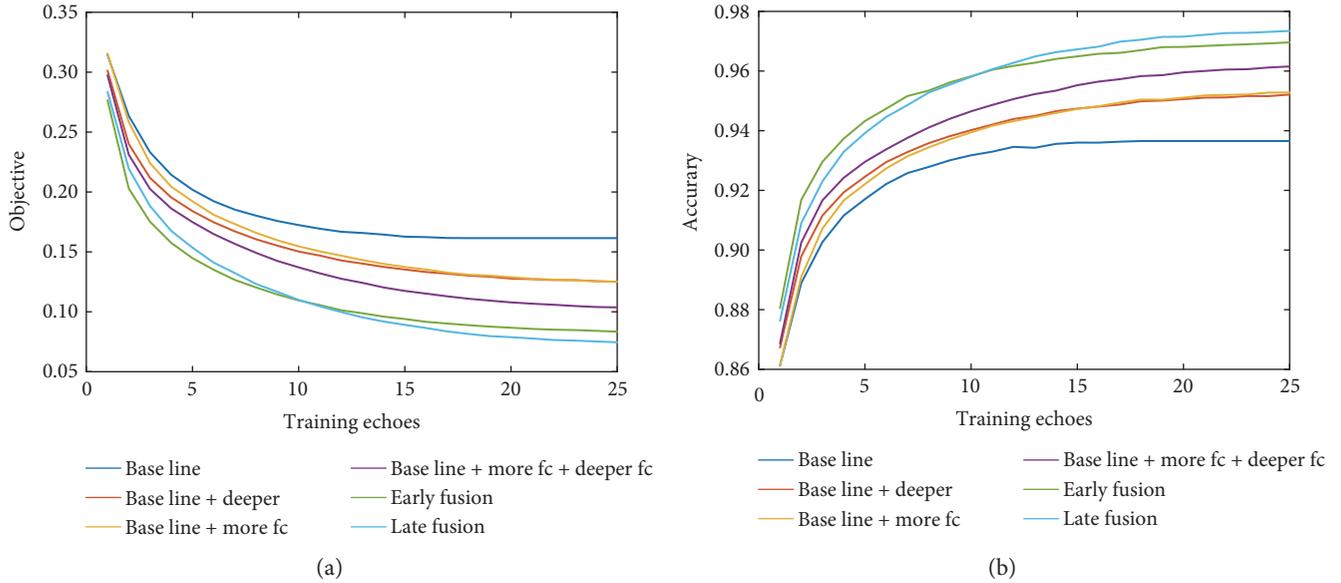


FIGURE 3: The training parameters of different network structures. (a) shows the convergence of the objective function during training, and (b) shows the error change during training.

TABLE 4: Performance of different CNN structures on the training data.

CNN structure	DSC	PPV	Sensitivity	Total parameters
Base line	0.7609	0.6626	0.8937	1.9×10^6
Base line + deeper	0.7872	0.6849	0.9253	2.1×10^6
Base line + more fc	0.7678	0.6568	0.9239	4.3×10^7
Base line + deeper +more fc	0.8073	0.7296	0.9036	4.3×10^7

TABLE 5: Performance of different CNN structures on the test data.

CNN structure	DSC	PPV	Sensitivity
Base line	0.7834	0.6915	0.9034
Base line + deeper	0.7910	0.6987	0.9114
Base line + more fc	0.7841	0.6773	0.9306
Base line + deeper +more fc	0.8021	0.7310	0.8886

Figure 3 that the parameters of the training after the addition of 3D information yield a much better convergence result. This was because the increased 3D information improves the recognition ability of the network. However, maybe due to the noises brought by 3D information, the actual segmentation results were not very satisfactory.

We also showed the result of the segmentation of the training set and the test set in Tables 6 and 7, after combined with the fully connected CRF. It can be seen that the combination of the CNN and the CRF can improve the recognition effect of tumors to a great extent. More importantly, the combination of these two 3D structures with the CRF seems to achieve better results. The results of early fusion had a higher DSC, and the results of late fusion were able to achieve higher sensitivity with similar global segmentation results.

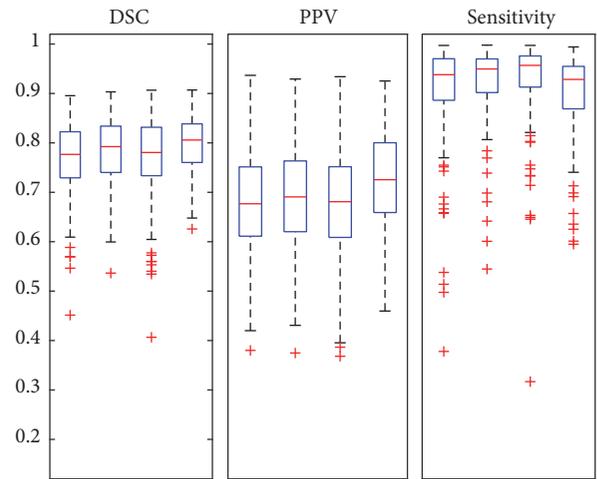


FIGURE 4: Boxplots of all data segmentation results of different CNN structures. The first of each group represents the base line, the second represents the deeper structure of the network, the third represents the network with more neurons in the fully connected layers, and the last one is the network with both deeper and more neurons in the fully connected layers.

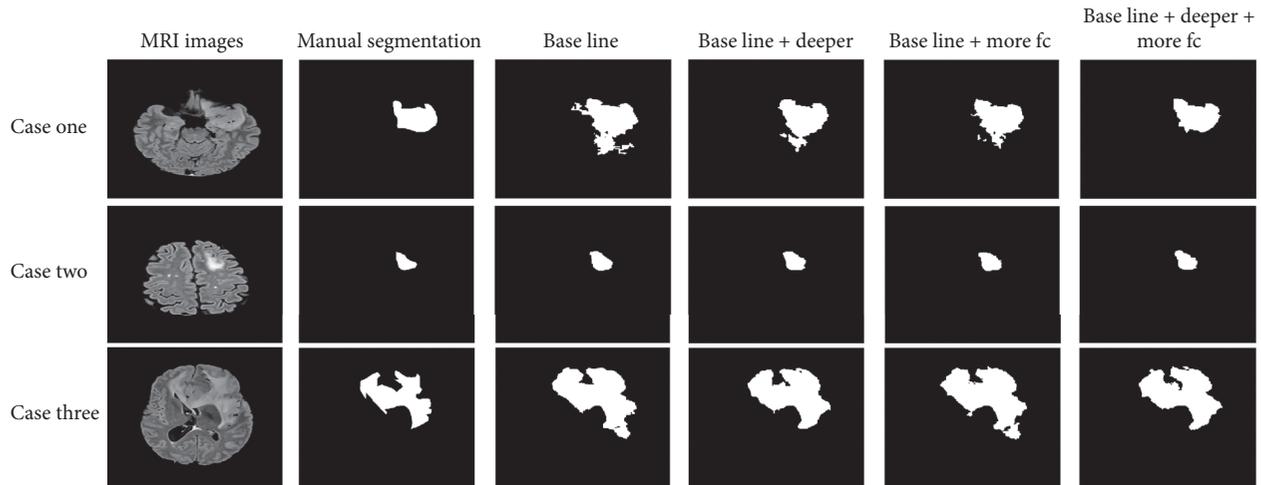


FIGURE 5: Segmentation results for networks with different depths. Each row corresponds to a case, and each column corresponds to a network structure of the segmentation results.

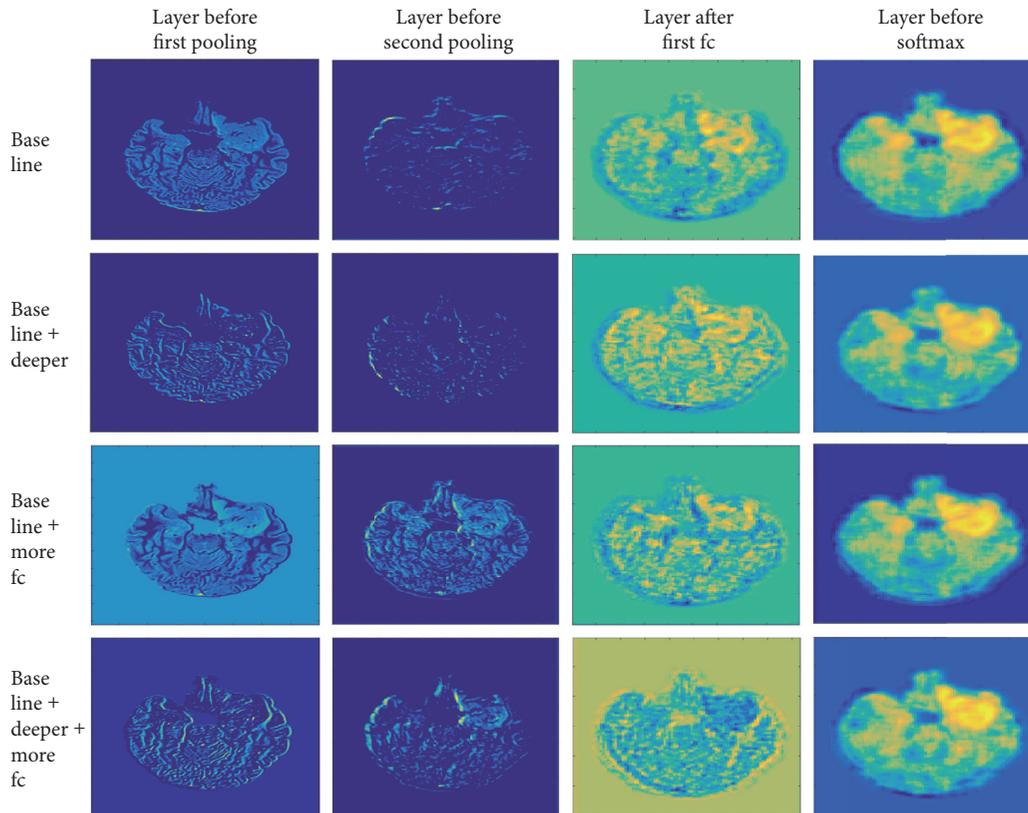


FIGURE 6: Features in networks of case one in Figure 4. Each column corresponds to a network structure, and each row corresponds to one kind of depth; the output of the last filter in the filter bank was selected.

The results are more evident in the globally validated Figure 7. Compared with the single way CNN, the combination of the CNN and the CRF achieved a significant improvement. Moreover, early fusion and late fusion accessed to more promotion in different aspects.

In Figure 8, we showed the superiority of our method in three examples. In the first case, we can see that the contours

of the tumor obtained with one way CNN are very rough and the boundaries acquired are not very good. The segmentation results are more precise and more accurate after combining the CNN with the fully connected CRF. It is worth being mentioned that we did not use the fully connected CRF to smoothen the edges completely. We found that appropriate iteration time of CRF would lead to better segmentation than

TABLE 6: Performance of different CNN structures associated with the CRF on the training data.

CNN structure	DSC	PPV	Sensitivity
One way CNN with CRF	0.8493	0.8355	0.8637
Early fusion with CRF	0.8506	0.8297	0.8745
Late fusion with CRF	0.8059	0.7415	0.8825

TABLE 7: Performance of different CNN structures associated with the CRF on the test data.

CNN structure	DSC	PPV	Sensitivity
One way CNN with CRF	0.8459	0.8577	0.8344
Early fusion with CRF	0.8504	0.8561	0.8447
Late fusion with CRF	0.8372	0.8113	0.8649

the one with fully smoothing. Therefore, the segmentation of our method might be rugged and preserved good edges of the tumor regions.

In the second case and the third case, we showed the benefits of combining nearby slices. The CNN combined with 3D information can identify the tumor area globally. In the second case, two network structures combined with 3D information can better identify the whole tumor area, even if a part of the tumor had very poor contrast.

In the third case, the upper surface of the tumor cannot be recognized by the single way CNN. But two kinds of network structures combined with 3D information can complete segmentation successfully and make the tumor identification more detailed.

As for two structures of early fusion and late fusion, we can see that the results of the former are more detailed and the latter are more likely to recognize more regions. Three convolution channels combined in late fusion showed responses to larger regions. This may be due to the fact that separate convolution channels obtained more 3D information but were more susceptible to noises.

We can further explore the advantages of our method of combining 3D information with Figure 9. In Figure 9, we showed the unary potential preparing for the input of the fully connected CRF. These features were set as the output of the last fully connected layers in CNNs. The color bars next to the score map showed the intensity distribution in the graph.

It can be seen that the intensity distribution of the unary potential obtained from two CNN structures with 3D information was more extensive and the color of the tumor region shown in the figure was brighter. The results corresponded to the point we said previously that the CNN can recognize tumor regions more certainly after adding 3D information. This capability would be beneficial for better recognition of the fully connected CRF, as can be inferred from (1).

3.4. Operation Times. Thanks to the CNN processing speed advantage and the simplified calculation method of the fully connected CRF, our proposed pipeline can be completed in a very short time. Specifically, by using GPU NVIDIA Quadro 600 on an Intel Xeon E5620 2.40 GHz machine, we need one

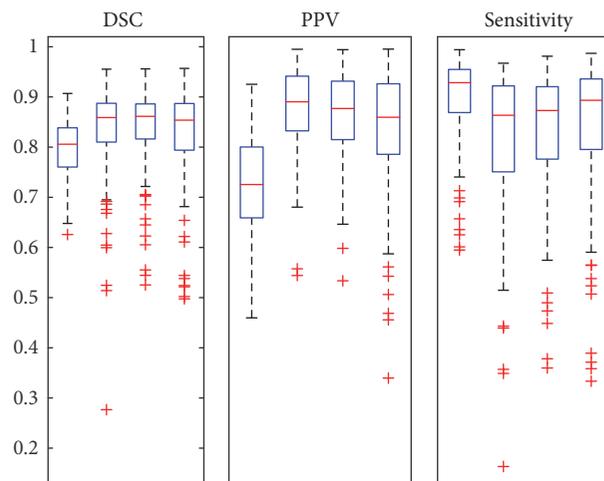


FIGURE 7: Boxplots of all data segmentation results of different CNN structures combined with fully connected CRF. The first plots of each group represent results of one way CNN, the second plots represent results of one way CNN connected to CRF, the third plots represent results of the early fusion structure with CRF, and the last ones are the results of the late fusion structure with CRF.

to seven days to train networks, and the specific time depends on the network complexity. After the network is obtained, the segmentation time was much less since the segmentation process required only the forward operation of the network. A slice takes only 2 to 10 seconds to get the segmentation results, and all images of a patient need 2 to 10 minutes. Similarly, the specific time varies according to the network structure. Compared with the traditional machine learning methods, our method is very advantageous in computing time [3].

4. Conclusions

In this study, we explore better ways to segment the LGG based on the CNN in the binary framework. We firstly studied the effect of different depths and the number of neurons in the fully connected layers on the tumor segmentation. It was found that the deepening of the network can optimize the segmentation results without increasing the amount of computation. Increasing the number of neurons in the fully connected layers under the same conditions can also improve the segmentation results. Next, we found that the use of the fully connected CRF as the CNN postprocessing can improve the segmentation results of the contour and edge and greatly enhances the segmentation results. At last, we proposed two kinds of network structures combined with 3D information. Experiments showed that the combination of these two structures and CRF can get better results. Early fusion can improve the segmentation results globally, and late fusion can make the segmentation results more sensitive. Using our proposed workflow, we can achieve better results on the LGG segmentation than one of the best CNN tumor segmentation methods at present. In order to better verify the applicability of our method, we will have a more detailed identification of brain tumors and validate our method on BRATS database in the future.

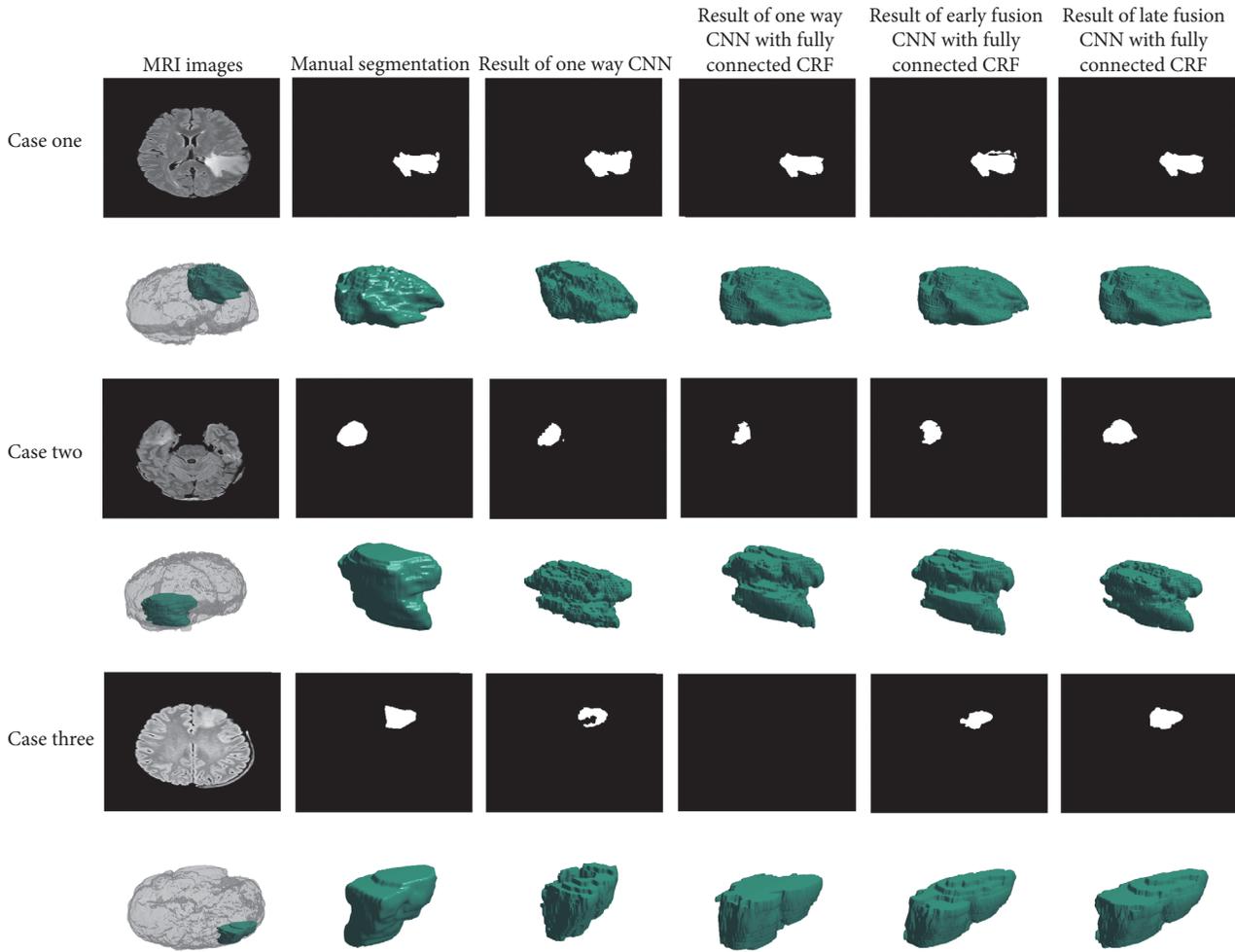


FIGURE 8: Segmentation results of different network structures combined with the fully connected CRF. Each row corresponds to a case, and each column corresponds to a network structure of the segmentation results. The 3D reconstructions of tumors are shown in the second row of each group.

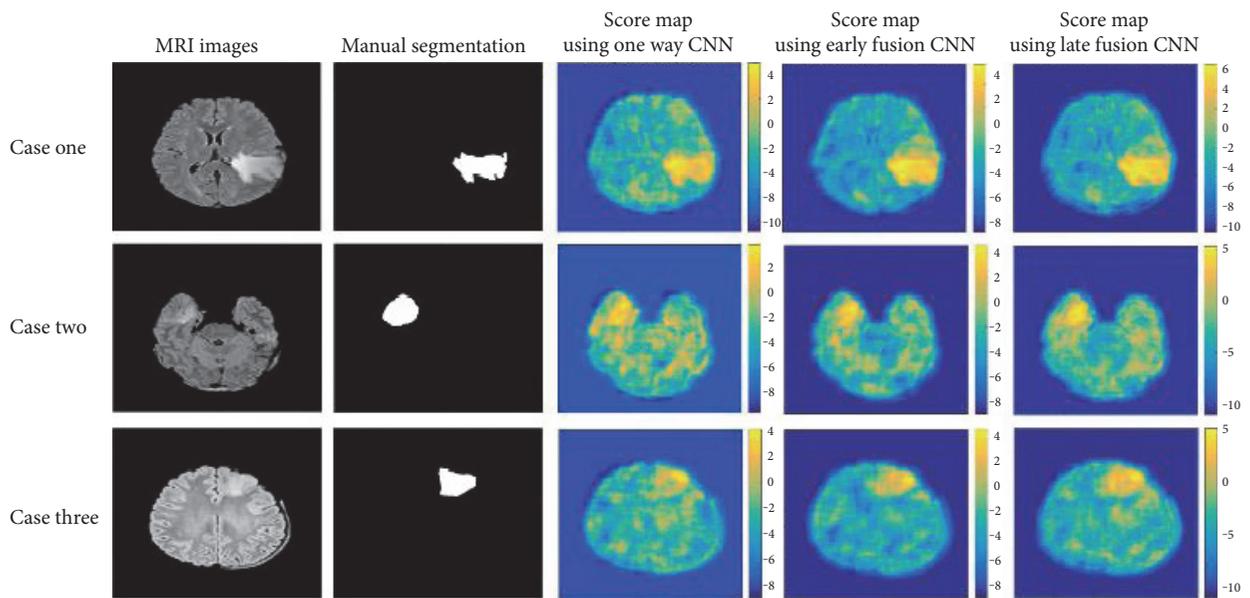


FIGURE 9: The score maps that are entered into the CRF of different network structures. Each row corresponds to a case and each column corresponds to a network structure.

Conflicts of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Basic Research Program of China (2015CB755500) and the National Natural Science Foundation of China (11474071).

References

- [1] F. B. Furnari, F. Tim, R. M. Bachoo et al., "Malignant astrocytic glioma: genetics, biology, and paths to treatment," *Genes and Development*, vol. 21, no. 21, pp. 2683–2710, 2007.
- [2] D. N. Louis, H. Ohgaki, O. D. Wiestler et al., "WHO classification of tumors of the central nervous system," *Acta Neuropathologica*, vol. 114, no. 2, pp. 97–109, 2007.
- [3] B. Menze, A. Jakab, S. Bauer et al., "The multimodal brain tumor image segmentation benchmark (BRATS)," *IEEE Transaction on Medical Imaging*, vol. 34, no. 2, pp. 1993–2024, 2015.
- [4] N. J. Tustison, B. B. Avants, P. A. Cook et al., "N4ITK: improved N3 bias correction," *IEEE Transaction on Medical Imaging*, vol. 29, no. 6, pp. 1310–1320, 2010.
- [5] T. M. Hsieh, Y. M. Liu, C. C. Liao, F. Xiao, I. J. Chiang, and J. M. Wong, "Automatic segmentation of meningioma from non-contrasted brain MRI integrating fuzzy clustering and region growing," *BMC Medical Informatics and Decision Making*, vol. 11, no. 54, p. 54, 2011.
- [6] J. E. Iglesias, M. R. Sabuncu, and K. Van Leemput, "A unified framework for cross-modality multi-atlas segmentation of brain MRI," *Medical Image Analysis*, vol. 17, no. 8, pp. 1181–1191, 2013.
- [7] S. Parisot, H. Duffau, S. Chemouny, and N. Paragios, "Graph-based detection, segmentation & characterization of brain tumors," in *Proceedings of the 30th IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'12)*, pp. 988–995, Providence, Rhode Island, USA, June 2012.
- [8] M. Huang, W. Yang, Y. Wu, J. Jiang, W. Chen, and Q. Feng, "Brain tumor segmentation based on local independent projection-based classification," *IEEE Transaction on Medical Imaging*, vol. 61, no. 10, pp. 2633–2645, 2014.
- [9] R. Meier, V. Karamitsou, S. Habeger, R. Wiest, and M. Reyes, "Parameter learning for CRF-based tissue segmentation of brain tumors," in *Proceedings of the Multimodal Brain Tumor Image Segmentation Challenge (MICCAI-BRATS)*, pp. 48–51, Munich, Germany, October 2015.
- [10] L. Zhao, W. Wu, and J. J. Corso, "Semi-automatic brain tumor segmentation by constrained MRFs using structural trajectories," in *Proceedings of the 16th International Conference on Medical Image and Computer-Assisted Intervention (MICCAI'13)*, pp. 567–575, Nagoya, Japan, September 2013.
- [11] O. Meier, M. Wilms, and H. Handels, "Highly discriminative features for glioma segmentation in MR volumes with random forests," in *Proceedings of the Multimodal Brain Tumor Image Segmentation Challenge (MICCAI-BRATS)*, pp. 38–41, Munich, Germany, October 2015.
- [12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Advances in Neural Information Processing Systems (NIPS'12)*, Lake Tahoe, USA, December 2012.
- [13] H. C. Shin, H. R. Roth, M. C. Gao et al., "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning," *IEEE Transaction on Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.
- [14] S. Pereira, A. Pinto, V. Alves, and C. A. Silva, "Brain tumor segmentation using convolutional neural networks in MRI images," *IEEE Transaction on Medical Imaging*, vol. 35, no. 5, pp. 1240–1251, 2016.
- [15] P. Dvorak and B. Menze, "Structured prediction with convolutional neural networks for multimodal brain tumor segmentation," in *Proceedings of the Multimodal Brain Tumor Image Segmentation Challenge (MICCAI-BRATS)*, pp. 13–24, Munich, Germany, October 2015.
- [16] M. Havaei, F. Dutil, C. Pal, H. Larochelle, and P. M. Jodoin, "A convolutional neural network approach to brain tumor segmentation," in *Proceedings of the Multimodal Brain Tumor Image Segmentation Challenge (MICCAI-BRATS)*, pp. 29–33, 2015.
- [17] V. Rao, M. S. Sarabi, and A. Jaiswal, "Brain tumor segmentation with deep learning," in *Proceedings of the Multimodal Brain Tumor Image Segmentation Challenge (MICCAI-BRATS)*, pp. 56–59, Munich, Germany, October 2015.
- [18] K. Kamnitsas, C. Ledig, V. F. J. Newcombe et al., "Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation," 2017, <http://arxiv.org/abs/1603.05959>.
- [19] R. Alexander, L. Kira, and W. Benedikt, "Relevance of T2 signal changes in the assessment of progression of glioblastoma according to the response assessment in neuro-oncology criteria," *Neuro-Oncology*, vol. 14, no. 2, pp. 222–229, 2012.
- [20] D. W. Shattuck and R. M. Leahy, "BrainSuite: an automated cortical surface identification tool," *Medical Image Analysis*, vol. 6, no. 2, pp. 129–142, 2002.
- [21] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the 33rd IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'15)*, pp. 1337–1342, Boston, MA, USA, June 2015.
- [22] A. A. A. Setio, F. Ciompi, G. Litjens et al., "Pulmonary nodule detection in CT images: false positive reduction using multi-view convolutional networks," *IEEE Transaction on Medical Imaging*, vol. 35, no. 5, pp. 1160–1169, 2016.
- [23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, <http://arxiv.org/abs/1409.1556>.
- [24] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the 32nd IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'14)*, pp. 1725–1732, Columbus, OH, USA, June 2014.
- [25] L. C. Chen, G. Papandreou, L. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," 2014, <http://arxiv.org/abs/1412.7062>.
- [26] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: probabilistic models for segmenting and labeling sequence data," in *Proceedings of the 18th International*

- Conference on Machine Learning (ICML'01)*, pp. 282–289, San Francisco, CA, USA, June 2001.
- [27] P. Krähenbühl and V. Koltun, “Efficient inference in fully connected CRFs with Gaussian edge potentials,” in *Proceedings of the 24th International Conference on Advances in Neural Information Processing Systems (NIPS'11)*, Granada, Spain, December 2011.
- [28] A. Vedaldi and K. Lenc, “MatConvNet: convolutional neural networks for MATLAB,” 2014, <http://arxiv.org/abs/1412.4564>.
- [29] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: surpassing human-level performance on ImageNet classification,” in *Proceedings of the 33rd IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'15)*, pp. 1026–1034, Boston, MA, USA, June 2015.
- [30] L. R. Dice, “Measures of the amount of ecologic association between species,” *Ecology*, vol. 26, no. 3, pp. 297–302, 1945.
- [31] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” 2013, <http://arxiv.org/abs/1311.2901>.

Research Article

A Real-Time Analysis Method for Pulse Rate Variability Based on Improved Basic Scale Entropy

Yongxin Chou,¹ Ruilei Zhang,¹ Yufeng Feng,² Mingli Lu,¹ Zhenli Lu,^{1,3} and Benlian Xu¹

¹School of Electrical and Automatic Engineering, Changshu Institute of Technology, Changshu 215500, China

²Changshu No. 1 People's Hospital, Changshu, China

³State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110014, China

Correspondence should be addressed to Yongxin Chou; lutchouyx@163.com

Received 17 November 2016; Revised 18 February 2017; Accepted 7 March 2017; Published 9 May 2017

Academic Editor: Valentina Camomilla

Copyright © 2017 Yongxin Chou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Base scale entropy analysis (BSEA) is a nonlinear method to analyze heart rate variability (HRV) signal. However, the time consumption of BSEA is too long, and it is unknown whether the BSEA is suitable for analyzing pulse rate variability (PRV) signal. Therefore, we proposed a method named sliding window iterative base scale entropy analysis (SWIBSEA) by combining BSEA and sliding window iterative theory. The blood pressure signals of healthy young and old subjects are chosen from the authoritative international database MIT/PhysioNet/Fantasia to generate PRV signals as the experimental data. Then, the BSEA and the SWIBSEA are used to analyze the experimental data; the results show that the SWIBSEA reduces the time consumption and the buffer cache space while it gets the same entropy as BSEA. Meanwhile, the changes of base scale entropy (BSE) for healthy young and old subjects are the same as that of HRV signal. Therefore, the SWIBSEA can be used for deriving some information from long-term and short-term PRV signals in real time, which has the potential for dynamic PRV signal analysis in some portable and wearable medical devices.

1. Introduction

Electrocardiogram (ECG) signal has been used for many diseases to assist in diagnosis in a clinic. The subtle changes of heart beat periods are called heart rate variability (HRV). The continuous heart rate or continuous RR wave intervals extracted from ECG signal are denoted as heart rate variability (HRV) signal [1]. An increasing number of studies have shown that HRV is a useful quantitative indicator for assessing the balance between the cardiac sympathetic nervous system and the parasympathetic nervous system and can be engaged in the diagnosis and prevention of some cardiovascular diseases such as sudden cardiac death and arrhythmia [2–4]. Pulse signal or continuous blood pressure signal generated by the systolic and diastolic of heart contains abundant physiological and pathological information of the cardiovascular system [5, 6]. The subtle change of vessel pulse periods is denoted as pulse rate variability (PRV). The continuous pulse rate or continuous PP wave intervals extracted from pulse signal or continuous blood pressure

signal are defined as PRV signal [7]. Because a heartbeat produces a vessel pulse, many studies show that PRV is a substitute for HRV to present the physiological and pathological changes of the cardiovascular system when the subjects are sleeping or testing, as well as in some nonstationary states [8–10]. In addition, due to the wide distribution of human vessels, the acquisition of a pulse signal is easier than that of an ECG signal. Therefore, the pulse signal is employed in many wearable and portable medical devices such as smart watches, wristbands, and smart glasses but not ECG signal [11, 12], and PRV signal has more practical values than HRV signal.

Because PRV signal has similar characteristics with HRV signal, the analysis methods of HRV signal are often employed to analyze PRV signal. These methods are divided into time domain methods, frequency domain methods, time-frequency domain methods, and nonlinear methods [13]. HRV signal and PRV signal generated by heartbeat are neither stochastic nor periodic; they are the results of many independent factors and have nonlinear properties.

Thus, the nonlinear methods are more useful for analyzing HRV signal and PRV signal, and there are many nonlinear methods such as recurrence quantification analysis, detrended fluctuation analysis, the Lyapunov exponent, and information entropy analysis [14, 15]. Among them, the information entropy analysis is an effective tool to present the complexity of the nonlinear signal. The sample entropy (SampEn), the approximate entropy (ApEn), the sign series entropy analysis (SSEA), the base scale entropy analysis (BSEA), and so on are been used for analyzing HRV signal [13, 16–18]. However, because of the long time consumption of these methods, they are not suitable for the PRV signal in real time. The BSEA, proposed by Li and Ning, can effectively detect the complexity dissimilarity of short-term HRV signal (about 5 minutes) in different physiological or pathological states [17], while it is unknown whether the BSEA is suitable for analyzing pulse rate variability (PRV) signal, so far. In addition, the 5 minutes of HRV signal analysis is too long for some acute cardiovascular disease (ACVD), and its time consumption still needs to be improved.

Therefore, this study proposed an improved basic scale entropy on the basis of BSEA with the theory of sliding window iterative; we denote it as sliding window iterative basic scale entropy analysis (SWIBSEA). The BSEA and SWIBSEA are engaged in analyzing the measured PRV signals, and by the results of the experiments, the accuracy and time consumption are compared between BSEA and SWIBSEA. In addition, the structure of this paper is as follows: in Section 2, the theories of BSEA and SWIBSEA are presented and then the experimental data are introduced. The results are shown in Section 3. Then, the results are discussed in Section 4. The conclusion is given in the last section.

2. Methods and Materials

2.1. Basic Scale Entropy Analysis. The process of BSEA is as follows [16]: (1) a series of vectors are constructed from PRV signal, and for each vector, we compute their basic scale (BS). (2) The vectors are symbolized and classified according to BS, each of these categories is a heart or pulse beat mode. (3) Computing the probability of each beat mode, and getting entropy of their probabilities, the entropy is denoted as BSE.

For a PRV signal with the length of N , $PP: \{PP(i): 1 \leq i \leq N, i \in N^*\}$, the m consecutive data points are used to construct a vector:

$$X(i) = [PP(i), PP(i+1), \dots, PP(i+m-1)]. \quad (1)$$

Thus, we will get $N-m+1$ vectors which are denoted as temporal sequence vectors (TSVs). m is the length of TSV; the larger the value of m , the more complex of the beat mode that TSV expresses. For each TSV, the BS is defined by the root mean square (RMS) of the difference for two adjacent data points:

$$BS(i) = \sqrt{\frac{\sum_{j=1}^{m-1} (PP(i+j) - PP(i+j-1))^2}{m-1}}, \quad (2)$$

where $BS(i)$ is the BS of the i th TSV.

Then, the BS is multiplied by a constant α ; the result is as the standard for the vector symbolization. The $(N-m+1) \times m$ TSVs $\{X(i)\}$ are symbolized, and the results are named symbol sequence vectors (SSVs) and denoted as $\{S_i(j)\}$, $\{S_i(j): 1 \leq i \leq N-m+1, 0 \leq j \leq m-1, i \in N^*, j \in N\}$. The symbolization process is as follows:

$$S_i(j) = \begin{cases} 0, & \overline{\mu}_i < PP_i(j) \leq \overline{\mu}_i + \alpha \times BS(i) \\ 1, & PP_i(j) > \overline{\mu}_i + \alpha \times BS(i) \\ 2, & \overline{\mu}_i - \alpha \times BS(i) < PP_i(j) \leq \overline{\mu}_i \\ 3, & PP_i(j) \leq \overline{\mu}_i - \alpha \times BS(i) \end{cases}, \quad (3)$$

where $\overline{\mu}_i$ is the mean of the i th TSV. $PP_i(j)$ is the $(i+j)$ th data points of $\{PP(i)\}$ or is the $(j+1)$ th datum in the i th TSV. The symbols 0, 1, 2, and 3 are the labels of different scopes for PRV amplitude and are employed for probability calculation; their values are of no practical significance. $S_i(j)$ is the $(j+1)$ th datum in the i th SSV. α is used to control the value of BS and to adjust the division range of PRV amplitude, the way to choose the value of α is as [19].

After getting the SSVs, we compute the probability of each vector. There are 4 symbols, 0, 1, 2, and 3, to express the vector, so we can get 4^m kinds of different SSVs, denoted by π . Each SSV is a heart or pulse beat mode. Then, we compute the probability of each beat mode in $N-m+1$ SSVs:

$$p(\pi) = \frac{\#\{t | PPG(t), \dots, PPG(t+m-1) \text{ has type } \pi\}}{N-m+1}, \quad (4)$$

where $1 \leq t \leq 4, t \in N^*$, and $\#$ is the number of π . The beat state with probability 0 is denoted as “disabled mode.”

Therefore, we define BSE as

$$BSE(m) = - \sum p(\pi) \log_2 p(\pi). \quad (5)$$

The BSE can be used to describe the change of heartbeat mode. Obviously, $0 \leq BSE(m) \leq \log_2 4^m$. When there is only one pulse mode, $BSE(m) = 0$. When there are 4^m pulse modes, and each mode has equal probability, $BSE(m) = \log_2 4^m$ is the maximum. The larger the entropy value, the more complicated the heartbeat mode, whereas the smaller the entropy value, the simpler the heartbeat mode.

2.2. Sliding Window Iterative Basic Scale Entropy Analysis. We improve the BSEA with the theory of sliding window iterative, and define the improved method as sliding window iterative base scale entropy analysis. The process is shown in Figure 1.

In Figure 1, a data buffer with the length of N_w is set to store PRV data points which are extracted from dynamic pulse signal. The process of SWIBSEA contains data updating and iterative. The PRV datum will be updated by the sliding window theory, and the BSE will be calculated with the iterative theory.

During data updating, we set 1 byte in buffer to store the latest PRV datum and denote it as $PP(N_w+1)$. Then, we delete the oldest PRV datum $PP(1)$, and the data in higher addresses move to lower addresses, $PP(i) = PP(i+1)$. If we

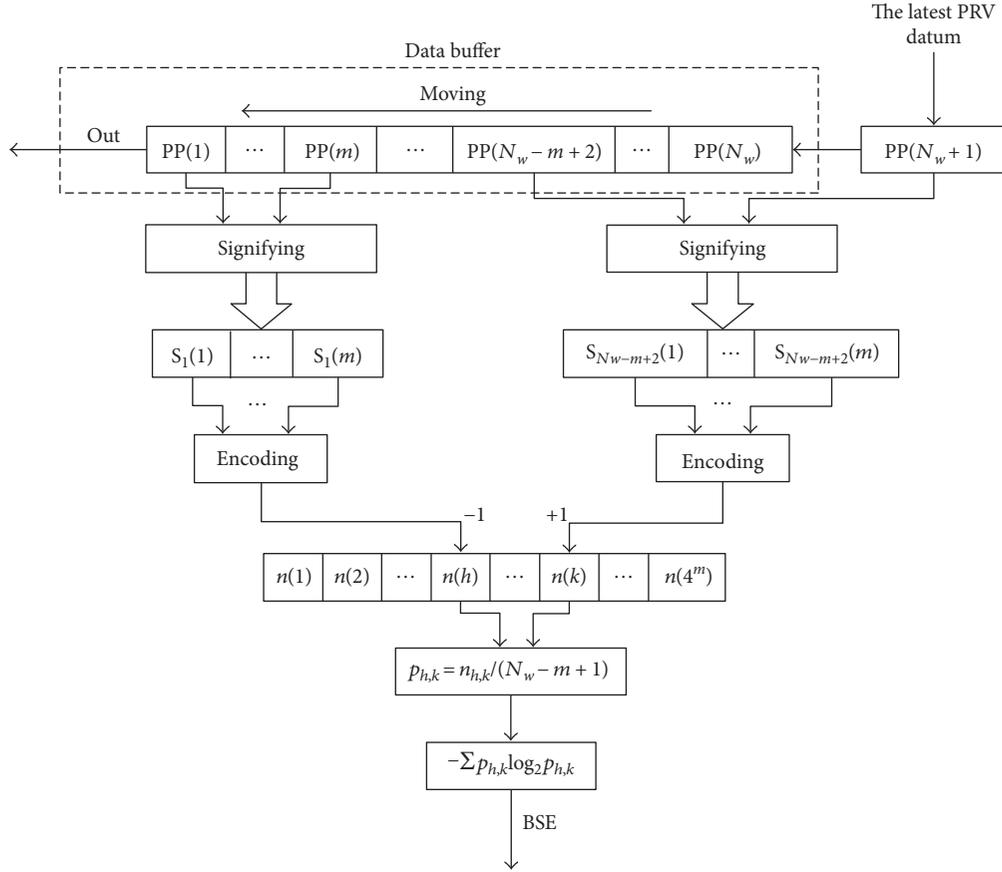


FIGURE 1: The process of SWIBSEA.

image the PRV signal as fixed, the buffer is like sliding forward in the data. This is the process of sliding window.

During iterative, the data in $PP(1)$ and $PP(N_w)$ is only changed in data updating, so it is only needed to compute the entropy of SSVs corresponding to $PP(1)$ and $PP(N_w)$, and then we will get the new BSE of all PRV data in buffer.

2.2.1. Construct the Temporal Sequence Vectors. According to the theory of BSEA, we need to reconstruct the PRV data with the length of N_w to $(N_w - m + 1) \times m$ TSVs. In the process of updating data, only the data in $PP(N_w + 1)$ and $PP(1)$ are changed; thus, for speeding up the calculation, we only construct the TSVs corresponding to them.

$$X(1) = [PP(1), PP(2), \dots, PP(m)], \quad (6)$$

$$X(N_w - m + 2) = [PP(N_w - m + 2), PP(N_w - m + 3), \dots, PP(N_w + 1)], \quad (7)$$

where $X(1)$ is the TSV of $PP(1)$, $X(N_w - m + 2)$ is the TSV of $PP(N_w + 1)$.

2.2.2. The Symbolization of Temporal Sequence Vectors $X(N_w - m + 2)$ and $X(1)$. For $X(1)$ computation, by (3),

$$S_1(j) = \begin{cases} 0, & \mu_1 < PP(j) \leq \mu_1 + \alpha \times BS_1 \\ 1, & PP(j) > \mu_1 + \alpha \times BS_1 \\ 2, & \mu_1 - \alpha \times BS_1 < PP(j) \leq \mu_1 \\ 3, & PP(j) \leq \mu_1 - \alpha \times BS_1 \end{cases}, \quad (8)$$

where μ_1 is the mean of $X(1)$. BS_1 is the base scale of $X(1)$, by (2):

$$BS_1 = \sqrt{\frac{\sum_{j=1}^{m-1} (PP(j+1) - PP(j))^2}{m-1}}. \quad (9)$$

Similarly, $X(N_w - m + 2)$ is symbolized and denoted as $\{S_{N_w-m+2}(j)\}$, $j = 1, \dots, m-1$.

2.2.3. Encode the Symbol Sequence Vector. During the data updating, only the numbers of $\{S_1(j)\}$ and $\{S_{N_w-m+2}(j)\}$ are changed; thus we only need to update their numbers. Moreover, for speeding up the calculation, we encode $\{S_1(j)\}$ and $\{S_{N_w-m+2}(j)\}$ to generate their storage addresses. If we denote the addresses as h and k (as is shown in Figure 1), then

$$h = \sum_{j=1}^m S_1(j) \times 4^{m-j} \quad (10)$$

and

$$k = \sum_{j=1}^m S_{N_w-m+2}(j) \times 4^{m-j}. \quad (11)$$

2.2.4. Entropy Calculation with Iterative Theory. We update the PRV data in buffer by sliding window theory; when getting a new PRV datum, the oldest PRV datum should be deleted. Thus, when computing entropy, we should just subtract the entropy of $\{S_1(j)\}$ and add the entropy of $\{S_{N_w-m+2}(j)\}$ from the entropy before data updating. Meanwhile, the intermediate variables are also need update by iterative after each data updating.

As shown in Figure 1, we denote the numbers of $\{S_1(j)\}$ and $\{S_{N_w-m+2}(j)\}$ as $n'(h)$, $n'(k)$ and $n(h)$, $n(k)$, the probabilities of $\{S_1(j)\}$ and $\{S_{N_w-m+2}(j)\}$ as $p'(h)$, $p'(k)$ and $p(h)$, $p(k)$, and the entropies of $\{S_1(j)\}$ and $\{S_{N_w-m+2}(j)\}$ as $BSE'(m)$ and $BSE(m)$ before and after updating PRV data, respectively. After data updating, $n(h)$ should subtract 1 and $n(k)$ should add 1. Then, $n(h) = n'(h) - 1$, $n(k) = n'(k) + 1$. The initial value of $BSE'(m) = 0$. In the iterative process of computing entropy, the antilogarithm of logarithm must be over 0; according to the changes of $n'(h)$, $n'(k)$, $n(h)$, and $n(k)$, there are four kinds of iterative methods to computing BSE as follows:

(1) $\mathbf{n(h)} > \mathbf{0}$, $\mathbf{n(k)} > \mathbf{1}$, which means the beat modes that $\{S_1(j)\}$ and $\{S_{N_w-m+2}(j)\}$ expressed all exist. Thus, during computing BSE, the antilogarithm of logarithm will be over 0. By (5),

$$\begin{aligned} BSE'(m) &= - \sum_{i=1}^M p'(i) \log_2 p'(i) \\ &= - p'(1) \log_2 p'(1) \cdots - p'(h) \log_2 p'(h) \cdots \\ &\quad - p'(k) \log_2 p'(k) \cdots - p'(M) \log_2 p'(M) \\ &= - p'(h) \log_2 p'(h) \cdots - p'(1) \log_2 p'(1) \cdots \\ &\quad - p'(M) \log_2 p'(M) \cdots - p'(k) \log_2 p'(k), \end{aligned} \quad (12)$$

$$\begin{aligned} BSE(m) &= - \sum_{i=1}^M p(i) \log_2 p(i) \\ &= - p(1) \log_2 p(1) \cdots - p(h) \log_2 p(h) \cdots \\ &\quad - p(k) \log_2 p(k) \cdots - p(M) \log_2 p(M) \\ &= - p(h) \log_2 p(h) \cdots - p(1) \log_2 p(1) \cdots \\ &\quad - p(M) \log_2 p(M) \cdots - p(k) \log_2 p(k), \end{aligned} \quad (13)$$

where $M = 4^m$.

Because only the modes of $\{S_1(j)\}$ and $\{S_{N_w-m+2}(j)\}$ are changed, thus, $-p(1) \log_2 p(1) = -p'(1) \log_2 p'(1)$, \dots , $-p(M) \log_2 p(M) = -p'(M) \log_2 p'(M)$. With (12) and (13),

$$\begin{aligned} BSE(m) &= BSE'(m) - p(h) \log_2 p(h) - p(k) \log_2 p(k) \\ &\quad + p'(h) \log_2 p'(h) + p'(k) \log_2 p'(k). \end{aligned} \quad (14)$$

Simplify (14):

$$\begin{aligned} BSE(m) &= BSE'(m) - \frac{n(h)}{N_w - m + 1} \log_2 \left(\frac{n(h)}{N_w - m + 1} \right) \\ &\quad - \frac{n(k)}{N_w - m + 1} \log_2 \left(\frac{n(k)}{N_w - m + 1} \right) \\ &\quad + \frac{n'(h)}{N_w - m + 1} \log_2 \left(\frac{n'(h)}{N_w - m + 1} \right) \\ &\quad + \frac{n'(k)}{N_w - m + 1} \log_2 \left(\frac{n'(k)}{N_w - m + 1} \right) \\ &= BS'(m) - \frac{n(h)}{N_w - m + 1} \log_2 \left(\frac{n(h)}{N_w - m + 1} \right) \\ &\quad - \frac{n(k)}{N_w - m + 1} \log_2 \left(\frac{n(k)}{N_w - m + 1} \right) \\ &\quad + \frac{n(h) + 1}{N_w - m + 1} \log_2 \left(\frac{n(h) + 1}{N_w - m + 1} \right) \\ &\quad + \frac{n(k) - 1}{N_w - m + 1} \log_2 \left(\frac{n(k) - 1}{N_w - m + 1} \right) \\ &= BS'(m) + \frac{n(h)}{N_w - m + 1} \log_2 \left(1 + \frac{1}{n(h)} \right) \\ &\quad + \frac{1}{N_w - m + 1} \log_2 \left(\frac{n(h) + 1}{n(k) - 1} \right) \\ &\quad + \frac{n(k)}{N_w - m + 1} \log_2 \left(1 - \frac{1}{n(k)} \right). \end{aligned} \quad (15)$$

Note that when the beat mode of $\{S_1(j)\}$ is the same as that of $\{S_{N_w-m+2}(j)\}$ before and after data updating, the number of the beat mode that $\{S_1(j)\}$ expressed should subtract 1: $n(h) = n'(h) - 1$, and the number of the beat mode that $\{S_{N_w-m+2}(j)\}$ expressed should add 1: $n(k) = n'(k) + 1 = n(h) + 1 = n'(h)$. Then, (14) is

$$\begin{aligned} BSE(m) &= BSE'(m) + \frac{n(h) + 1}{N_w - m + 1} \log_2 \left(\frac{n(h) + 1}{N_w - m + 1} \right) \\ &\quad + \frac{n(h)}{N_w - m + 1} \log_2 \left(\frac{n(h)}{N_w - m + 1} \right) \\ &\quad - \frac{n(h)}{N_w - m + 1} \log_2 \left(\frac{n(h)}{N_w - m + 1} \right) \\ &\quad - \frac{n(h) + 1}{N_w - m + 1} \log_2 \left(\frac{n(h) + 1}{N_w - m + 1} \right) \\ &= BSE'(m). \end{aligned} \quad (16)$$

That means the entropy is not changed when data updating. Because the beat modes of $\{S_1(j)\}$ and $\{S_{N_w-m+2}(j)\}$ are the same, the total number of beat modes remains unchanged.

(2) $\mathbf{n(h)} = \mathbf{0}$, $\mathbf{n(k)} > \mathbf{1}$, which means the beat mode $\{S_1(j)\}$ disappeared after data updating, and $n'(h) = 1$, $p(h) = 0$, $p'(h) = 1/(N_w - m + 1)$. By (14),

$$\begin{aligned}
\text{BSE}(m) &= \text{BSE}'(m) + \frac{1}{N_w - m + 1} \log_2 \left(\frac{1}{N_w - m + 1} \right) \\
&\quad + \frac{n(k) - 1}{N_w - m + 1} \log_2 \left(\frac{n(k) - 1}{N_w - m + 1} \right) \\
&\quad - \frac{n(k)}{N_w - m + 1} \log_2 \left(\frac{n(k)}{N_w - m + 1} \right) \\
&= \text{BSE}'(m) + \frac{n(k)}{N_w - m + 1} \log_2 \left(1 - \frac{1}{n(k)} \right) \\
&\quad - \frac{1}{N_w - m + 1} \log_2 (n(k) - 1).
\end{aligned} \tag{17}$$

(3) $\mathbf{n}(\mathbf{h}) > \mathbf{0}$, $\mathbf{n}(\mathbf{k}) = \mathbf{1}$, which means the beat mode of $\{S_{N_w - m + 2}(j)\}$ appeared the first time after data updating, and $n'(h) = n(h) + 1$, $n(k) = 1$, $n'(k) = 0$, $p'(k) = 0$, $p(k) = 1/(N_w - m + 1)$. By (14),

$$\begin{aligned}
\text{BSE}(m) &= \text{BSE}'(m) + \frac{n(h) + 1}{N_w - m + 1} \log_2 \left(\frac{n(h) + 1}{N_w - m + 1} \right) \\
&\quad - \frac{n(h)}{N_w - m + 1} \log_2 \left(\frac{n(h)}{N_w - m + 1} \right) \\
&\quad - \frac{1}{N_w - m + 1} \log_2 \left(\frac{1}{N_w - m + 1} \right) \\
&= \text{BSE}'(m) + \frac{n(h)}{N_w - m + 1} \log_2 \left(1 + \frac{1}{n(h)} \right) \\
&\quad + \frac{1}{N_w - m + 1} \log_2 (n(h) + 1).
\end{aligned} \tag{18}$$

(4) $\mathbf{n}(\mathbf{h}) = \mathbf{0}$, $\mathbf{n}(\mathbf{k}) = \mathbf{1}$, which means the beat mode of $\{S_1(j)\}$ disappeared, the beat mode of $\{S_{N_w - m + 2}(j)\}$ appeared the first time after updating data, and the total number of beat modes is unchanged. Then, $n'(h) = 1$, $n'(k) = 0$, $p'(k) = p(h) = 0$, $p'(h) = p(k) = 1/(N_w - m + 1)$. By (14),

$$\begin{aligned}
\text{BSE}(m) &= \text{BSE}'(m) + \frac{1}{N_w - m + 1} \log_2 \left(\frac{1}{N_w - m + 1} \right) \\
&\quad - \frac{1}{N_w - m + 1} \log_2 \left(\frac{1}{N_w - m + 1} \right) \\
&= \text{BSE}'(m).
\end{aligned} \tag{19}$$

By (15), (16), (17), (18), and (19), we will obtain the BSE of PRV signal based on sliding window and iterative theory.

2.3. Experimental Data. The experimental data we used are from the international authority of the database: PhysioNet/Fantasia [20]. In this database, there are 40 health subjects which have the same proportion of men and women, 20 of them are the elderly (65–85 years old, data name: f2o01m~f2o20m), the remaining subjects are the young (21–34 years old, data name: f2y01m~f2y20m). The ECG signal, continuous blood pressure signal, and respiration signal are recorded when the subjects under rest and watching the Fantasia movies to keep awake. However, only half of the subjects' blood pressure signals are recorded (data

name: f2y01m~f2y10m, f2o01m~f2o10m). The data sampling frequency is 250 Hz, and the duration is 66 minutes.

The experimental data we used are the continuous blood pressure signals. Compared with the ECG signal, the blood pressure signal is uncalibrated. Therefore, the dynamic difference threshold method is used to calibrate the P waves [21], and the accuracy of the calibration is determined manually. Then, by making a first-order difference for the locations of the calibrated P waves, we will obtain a set of continuous PP intervals, which is PRV signal.

In reality, there are some singularities in PRV signal that have a bad effect on the signal processing results. A pretreatment method is employed to delete the singularities from these short-term PRV signals. The steps are as follows [22]:

Step 1. For the first datum PP(1) of PRV signal, if

$$|\text{PP}(1) - \text{mean}(\text{PP})| > 1.5 \times \text{std}(\text{PP}), \tag{20}$$

then PP(1) is a singularity and thus deleted. In (20), mean(PP), std(PP) are the mean and the standard deviation of a short-term PRV signal, respectively.

Step 2. For the i th datum PP(i) of PRV signal, if

$$\begin{aligned}
\text{PP}(i) &> 1.3 \times \text{PP}(i-1) \quad \text{or} \\
\text{PP}(i) &< 0.7 \times \text{PP}(i-1),
\end{aligned} \tag{21}$$

then PP(i) is singular and deleted, where PP($i-1$) is the datum before PP(i).

3. Experimental Results

For simulating the process of PRV analysis in microcontroller system, the length of buffer is set to N_w , which is the length of sliding window. In this study, the sliding window theory is used to data updating. When obtaining a new PRV datum, the data in buffer are analyzed with BSEA and SWIBSEA, and the performance of the two methods are compared.

The experimental data are continuous blood pressure signals that are used to generate PRV signals; the results of a young subject and an old subject are selected randomly and shown in Figure 2. With the individual differences in heart rate, the length of two PRV signals are different. For 66 minutes of data, the young one is 2793 points corresponding to the mean pulse rate is 56.1 beat per minute (bpm) and the old one is 4849 points corresponding to the mean pulse rate is 73.8 bpm.

In this study, the performance of SSEA and SWISSEA are quantitatively evaluated by mean square error (MSE) and program running time.

The MSE is defined as

$$\text{MSE} = \sqrt{\frac{1}{L} \left[\sum_{i=1}^L \text{BSE}(i) - \text{SWIBSE}(i) \right]^2}, \tag{22}$$

where L is the length of entropy, BSE(i) is the BSE extracted by BSEA, and SWIBSE(i) is the BSE extracted by SWIBSEA.

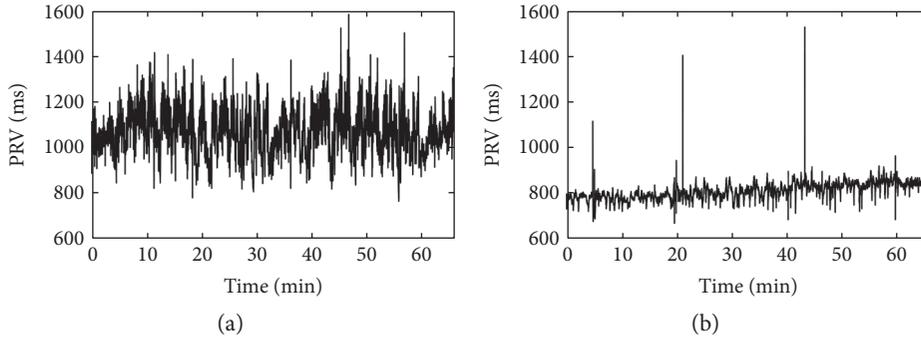


FIGURE 2: The PRV signal of a young subject (a) and an old subject (b).

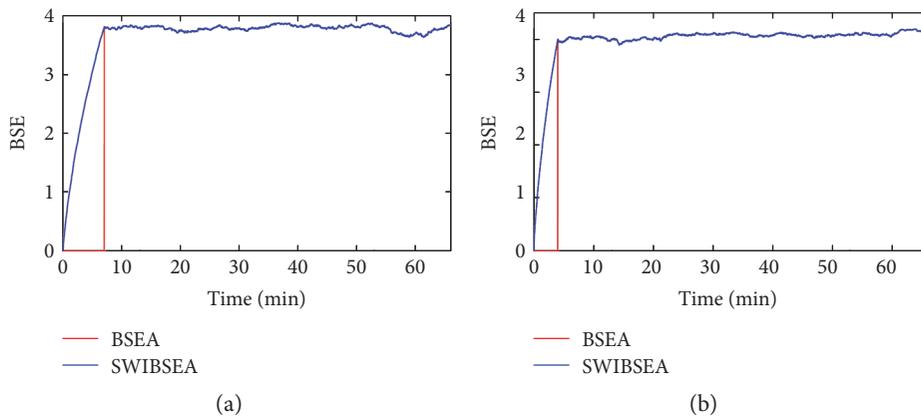


FIGURE 3: The comparison of BSEA and SWIBSEA for a young subject (a) and an old subject (b), when $\alpha = 0.5$, $N_w = 300$, and $m = 3$.

For the program running time, we test the programs of SSEA and SWISSEA in MATLAB 2016a on a PC with i7-6700HQ CPU (2.60 GHz, 16 GB buffered RAM).

3.1. Comparison of BSEA and SWIBSEA. The BSEA and the SWIBSEA are used to compute the entropy of the PRV signals in Figure 2. The results are shown in Figure 3.

When the total number of PRV data points in buffer is less than window width, $N < N_w$, for BSEA, the BSE = 0 based on its theory; for SWIBSEA, this is the initial process, and we compute the entropy by iterative, and the value increases with N . When $N_w \geq N$, the BSEA updates PRV data by sliding window and calculates BSE, while the SWIBSEA computes BSE by sliding window iteration; their entropies are shown in Figure 2. From the figure, it can be seen that their values are the same and their MSE = 0. The BSE of the young subject is 3.784 ± 0.050 (mean \pm std), and the old subject is 4.056 ± 0.053 , except for the entropy of the initial stage.

The time consumption of BSEA and SWIBSEA are 0.132 s and 4.769 s for the young subject and 0.192 s and 8.438 s for the old subject, respectively. For the two 66 minutes PRV signals, the time BSEA cost are 36 times and 44 times for SWIBSEA, respectively. Although the blood pressure signals of the old and the young subjects have the same length, but because of the individual differences, their pulse rates are different, and the PRV signal lengths are

different. Thus, the time consumptions of the young and the old are difference. While compared with BSEA, SWIBSEA saves a lot of running time in computing BSE and keeps its values unchanged.

3.2. Comparison of BSEA and SWIBSEA under the Different Lengths of SSV. According to the process of BSEA and SWIBSEA, the length m of the SSV has great influence on the running time of their program. The longer the length is, the more the heartbeat modes are represented by the SSVs. Therefore, keeping the width of sliding window and α unchanged (here, they are assigned to 300 and 0.5, respectively), we increase the value of m from 2 to 10, and the time consumption of these two methods for a young subject and an old subject are as shown in Figures 4 and 5.

Figure 4 shows the time consumption of the two method for the 66 minutes PRV signal of a young subject; Figure 4(a) is comparison of two method, and Figure 4(b) is the time consumption of SWIBSEA. When m increases from 2 to 10, the time they cost are all increased. The time consumption of SWIBSEA are from 0.182 s to 0.218 s, and that of BSEA are 42, 45, 45, 46, 50, 51, 62, 134, and 426 times for SWIBSEA. The growth rate of BSEA is 85.317 s, while that of SWIBSEA is only 0.036 s.

Figure 5 is the time consumption of an old subject. When m increases from 2 to 10, the time consumption of

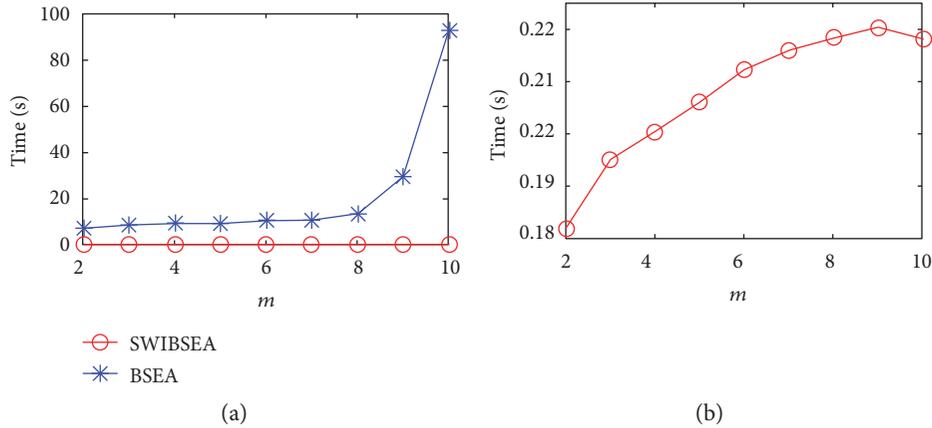


FIGURE 4: The time consumption of SWIBSEA (b) and BSEA (a) under the different length of SSVs for a young subject, when $N_w = 300$ and $\alpha = 0.5$.

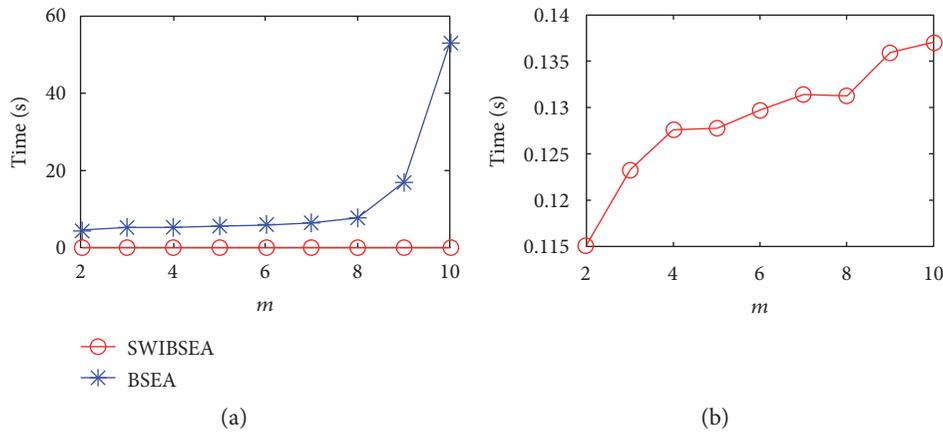


FIGURE 5: The time consumption of SWIBSEA (b) and BSEA (a) under the different length of SSV for an old subject, when $N_w = 300$ and $\alpha = 0.5$.

SWIBSEA are from 0.115 s to 0.137 s, and that of BSEA are 40, 42, 42, 44, 46, 49, 60, 126, and 388 times for SWIBSEA. The growth rate of BSEA is 48.524 s, while that of SWIBSEA is only 0.022 s. Although the time consumption of the two methods are all increased, but the increase of SWIBSEA is far less than that of BSEA, SWIBSEA will save much more running time of a program.

The variation of m will cause some changes of the SSVs in the window, which inevitably causes the change of the BSE value. As is shown in Figure 6, when m increases from 2 to 10, the entropies of young subjects and old subjects are increased. There are significant differences between BSEA and SWIBSEA ($P < 0.001$, two-sample t -test), and the increase of m does not affect the difference between the young and the elderly.

3.3. Comparison of BSEA and SWIBSEA under Different N_w . The width N_w of sliding window is corresponding to the length of buffer in a microcontroller system, and the range of N_w is varied from several minutes PRV data points (short-term

PRV signal) to several hours data points (long-term PRV signal), or even to 24 hours PRV data points. The short-term PRV signal is used to derive some changes in the autonomic nervous system within a short time, and the long-term PRV signal is used to reflect the long time and slow changes of the autonomic nervous system. They have potential to apply in a clinic. However, with the increase of N_w , the time consumption will increase inevitably. The results are shown in Figures 7 and 8. The values of m and α are chosen randomly; here, $m = 3$, $\alpha = 0.5$. Then, the N_w increases from 100 to 1000 data points with the interval of 100. For the PRV signal of the young subject (in Figure 7), the time SWIBSEA used are from 0.1301 s to 0.1093 s, respectively. The time BSEA used are 15, 30, 42, 56, 67, 80, 93, 104, 112, and 121 times for SWIBSEA. For the PRV signal of the old subject (in Figure 8), the time SWIBSEA used are from 0.199 s to 0.185 s, respectively. The time BSEA used are 17, 33, 49, 63, 79, 93, 110, 124, 138, and 153 times for SWIBSEA. The increase of BSEA from $N_w = 100$ to $N_w = 1000$ are 25.188 s, but that of SWIBSEA decreases and the decrease is only 0.014 s. Because

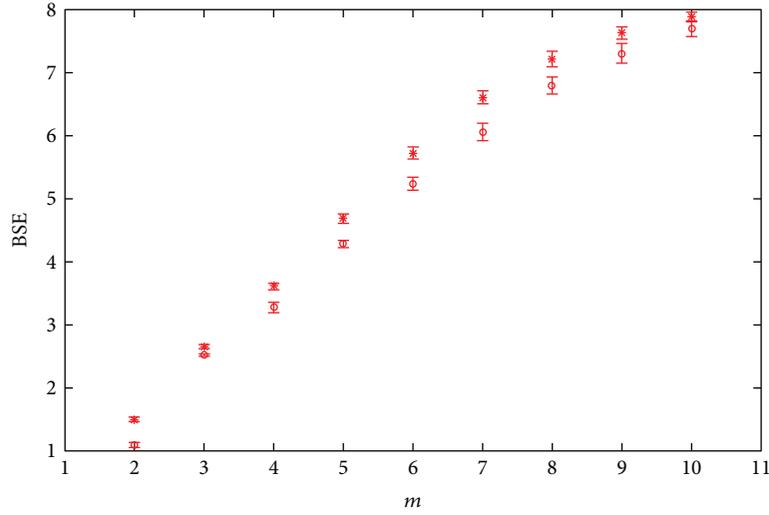


FIGURE 6: The BSE under different m 's, the results are shown with mean \pm std, $N_w = 300$ and $\alpha = 0.5$. “*” is the BSE of old subjects; “o” is the BSE of young subjects.

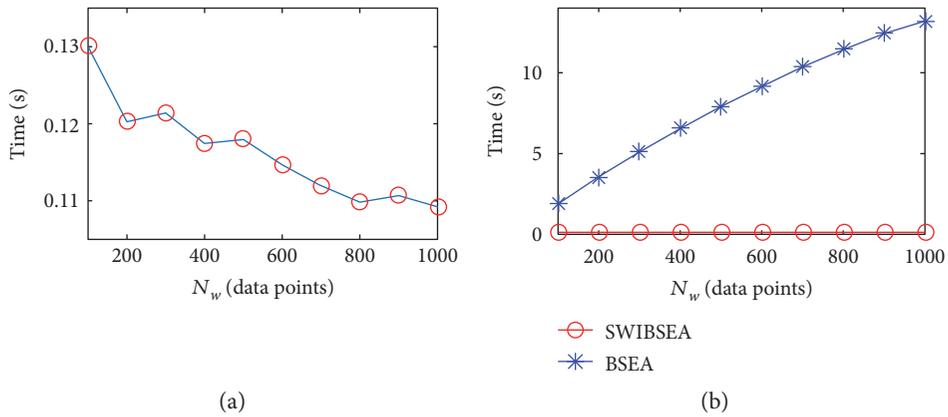


FIGURE 7: The time consumption of SWIBSEA (a) and BSEA (b) under the different N_w for a young subject, when $m = 3$, $\alpha = 0.5$.

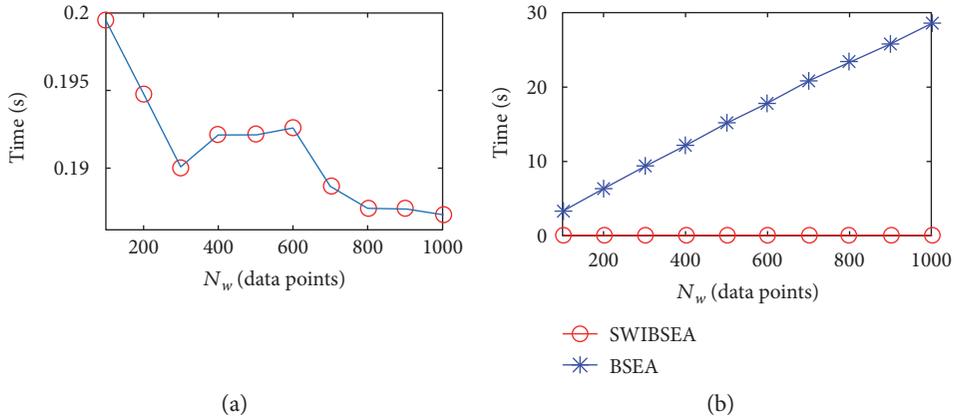


FIGURE 8: The time consumption of SWIBSEA (a) and BSEA (b) under the different N_w for an old subject, when $m = 3$, $\alpha = 0.5$.

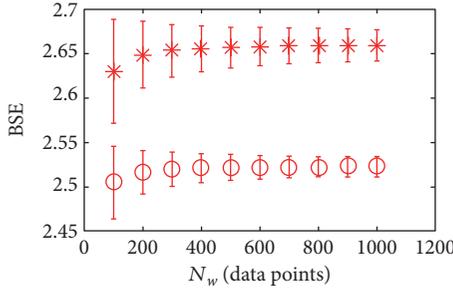


FIGURE 9: The BSE under different N_w . The results are shown with mean \pm std. $m = 3$ and $\alpha = 0.5$. “*” is the BSE of an old subject; “o” is the BSE of a young subject.

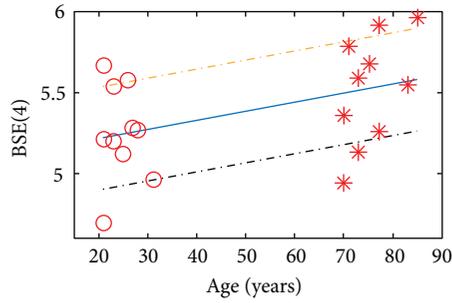


FIGURE 10: The SSE results of young subjects and old subjects. “o” is the young subjects, and “*” is the old subjects. Solid line is the linear fitting result of BSE changed with age. Chain dotted line is the 95% confidential region.

the SWIBSEA computes entropy with iterative and only updates the related variables of $PP(1)$ and $PP(N_w)$, the BSEA needs to update all the variables in sliding window.

The change of N_w will have influence on BSE, as is shown in Figure 9. When N_w increases from 100 to 1000 data points, the difference of BSE between the young subject and the old subject are more and more larger.

4. Discussion

BSEA, as a nonlinear method, has been employed to HRV signal analysis. Li and Ning [17] used BSEA to analyze the short-term HRV (500 data points) extracted from ECG signals of the PhysioNet/fantasia. When $m = 4$, $N_w = 500$, and $\alpha = 0.2$, the results of BSEA show that the entropy increases with aging. The entropies are tested by two-sample t -test and $P < 0.05$. If the BSE of healthy young subjects represent the best physiological state of the human body, the BSE of healthy old subjects deviated from that of the old subjects, which indicates that normal aging can lead to some function degradation of the body’s control system. We use SWIBSEA to analyze the HRV signals in [17] and get the same results as [17]. Similar to the HRV signal, the PRV signal is also generated from heartbeat. When $m = 4$, $N_w = 500$, and $\alpha = 0.2$, the BSE of PRV signals are shown in Figure 10, and the entropies in the figure are the mean of BSE which are derived by SWIBSEA. The two-sample t -test of $P = 0.008$ and < 0.05 . We do the linear fitting of the entropies by aging,

as shown on the solid line, the entropies increase with aging, the same as the result of short-term HRV signals analysis.

BSE has been effectively used for HRV signals analysis. In this study, on the basis of BSEA and with the theory of sliding window iterative, we proposed SWIBSEA for improving the computing efficiency of BSEA. Different from BSEA, it is not necessary to process all the data in buffer after date updating with sliding window for SWIBSEA. For example, when the buffer cache is 1024 bytes, $m = 5$. For BSEA, it needs more than 8169 bytes of memory space to store intermediate variables, that is, $N_w = 1024$ bytes for storing PRV data, $N_w - m + 1 = 1024 - 5 + 1 = 1020$ bytes for storing $\{BS(i)\}$, $(N_w - m + 1) \times m = (1024 - 5 + 1) \times 5 = 5100$ bytes for storing SSVs, $4^m = 1024$ bytes for storing the number of different SSVs, and 1 byte for storing BSE. Meanwhile, it also needs $N_w - 1 = 1023$ times shifting operations to update data; $N_w - m + 1 = 1024 - 5 + 1 = 1020$ times RMS operations to compute BS; 1020 times loops and $(N_w - m + 1) \times m \times 4 = (1024 - 5 + 1) \times 5 \times 4 = 20400$ times comparison operations to construct SSVs; $N_w \times m = 1024 \times 5 = 5120$ times comparison operations to updating $\{n(j)\}$; 1024 times multiplications for computing $\{p(j)\}$; 1024 times multiplications, 1024 times logarithms, and 1023 times additions for getting BSE. However, For SWIBSEA, it only needs 2061 bytes, that is, 1024 bytes for buffering PRV data, 2 bytes for storing BS_1 and $BS_{N_w - m + 2}$, $2^*m = 10$ bytes for storing $\{S_1(j)\}$ and $\{S_{N_w - m + 2}(j)\}$, $4^m = 1024$ bytes for storing $\{n(j)\}$, and 1 byte for storing BSE. Meanwhile, it needs 1023 times shifting operations to update PRV signal, and 2 times RMS operations to compute BS_1 and $BS_{N_w - m + 2}$, $m^*4*2 = 40$ times comparison operations to compute $\{S_1(j)\}$ and $\{S_{N_w - m + 2}(j)\}$, at most 9 times multiplications, 3 times logarithms, and 6 times additions to compute BSE (sometimes, it does not need to update the BSE). Since the storing addresses of $\{n(j)\}$ are generated by encoding, no loops and comparison operations are required when update $\{n(j)\}$. Compared with BSEA, the SWIBSEA saves 6108 bytes buffer space. In addition, the sliding window iterative theory is used by the SWIBSEA, thus, its time consumption are reduced, and it can be engaged in PRV signal analysis in real time.

Moreover, the proposed method significantly reduces the time consumption and buffer cache, and can be used for both short-term and long-term PRV signals (in Section 3.3) by adjusting the width of the sliding window. The length of PRV signal only determines the initialization time; after the initialization, the BSE can be calculated by iterative, and the time consumption between the short-term PRV signal and the long-term PRV signal has not significantly increased.

5. Conclusion

In this study, the sliding window iterative theory is used to improve the BSEA, and the SWIBSEA is proposed and employed to analyze the data of healthy young and old subjects from MIT/PhysioNet/Fantasia database. The results show that compared with BSEA, the SWIBSEA reduces the computing time and saves the buffer cache while keeping the BSE unchanged. Meanwhile, by adjusting the width of

sliding window, the SWIBSEA can analyze the long-term and short-term PRV signals in real time. The experimental results show that the BSE increases with aging, and normal aging leads to some functions degradation of the control system. Therefore, the SWIBSEA could be employed in some wearable and portable devices for analyzing dynamic PRV signal in real time.

Conflicts of Interest

The authors indicated no potential conflicts of interest.

Acknowledgments

This work was supported by the State Key Laboratory of Robotics of China (Grant 2014-O08), the Natural Science Fundamental Research Program of Higher Education Colleges in Jiangsu Province of China (Grant 14KJB510001), the Special Research Foundation of Young Teachers (Grant XZ1613), the Suzhou Municipal Science and Technology Plan Project (Grants SYG201548, SYG201504), and the National Natural Science Foundation of China (Grant 61273312, Grant 61673075, and Grant 61473283).

References

- [1] R. E. Kleiger, P. K. Stein, and J. T. Bigger, "Heart rate variability: measurement and clinical utility," *Annals of Noninvasive Electrocardiology*, vol. 10, no. 1, pp. 88–101, 2005.
- [2] J. L. Hamilton and L. B. Alloy, "Atypical reactivity of heart rate variability to stress and depression across development: systematic review of the literature and directions for future research," *Clinical Psychology Review*, vol. 50, pp. 67–79, 2016.
- [3] A. K. Schuster, J. E. Fisher, J. F. Thayer, D. Mauss, and M. N. Jarczok, "Decreased heart rate variability correlates to increased cardiovascular risk," *International Journal of Cardiology*, vol. 203, pp. 728–730, 2015.
- [4] E. Ebrahimzadeh, M. Pooyan, and A. Bijar, "A novel approach to predict sudden cardiac death (SCD) using nonlinear and time-frequency analyses from HRV signals," *PLoS One*, vol. 9, no. 2, article e81896, 2014.
- [5] E. Zahedi, V. Sohani, M. A. Ali, K. Chellappan, and G. K. Beng, "Experimental feasibility study of estimation of the normalized central blood pressure waveform from radial photoplethysmogram," *Journal of Healthcare Engineering*, vol. 6, no. 1, pp. 121–144, 2015.
- [6] S. Hu, V. Azorinperis, and J. Zheng, "Opto-physiological modeling applied to photoplethysmographic cardiovascular assessment," *Journal of Healthcare Engineering*, vol. 4, no. 4, pp. 505–528, 2013.
- [7] Y. X. Chou, A. Zhang, P. Wang, and J. Gu, "Pulse rate variability estimation method based on sliding window iterative DFT and Hilbert transform," *Journal of Medical and Biological Engineering*, vol. 34, no. 4, pp. 347–355, 2014.
- [8] E. Yu, D. He, Y. Su, L. Zheng, Z. Yin, and L. Xu, "Feasibility analysis for pulse rate variability to replace heart rate variability of the healthy subjects," in *2013 IEEE International Conference on Proceedings of the Robotics and Biomimetics (ROBIO)*, pp. 1065–1070, Shenzhen, December 2013.
- [9] A. Schäfer and J. Vagedes, "How accurate is pulse rate variability as an estimate of heart rate variability?: a review on studies comparing photoplethysmographic technology with an electrocardiogram," *International Journal of Cardiology*, vol. 166, no. 1, pp. 15–29, 2013.
- [10] C. C. Chuang, J. J. Ye, W. C. Lin, K. T. Lee, and Y. T. Tai, "Photoplethysmography variability as an alternative approach to obtain heart rate variability information in chronic pain patient," *Journal of Clinical Monitoring and Computing*, vol. 29, no. 6, pp. 801–806, 2015.
- [11] M. B. Mashhadi, E. Asadi, M. Eskandari, S. Kiani, and F. Marvasti, "Heart rate tracking using wrist-type photoplethysmographic (PPG) signals during physical exercise with simultaneous accelerometry," *IEEE Signal Processing Letters*, vol. 23, no. 2, pp. 227–231, 2015.
- [12] J. Guyomard and R. Stortelder, *Heart Rate Measurement through PPG: Heartbeat Measurement in a Wireless Headset*, The Delft University of Technology, Delft, Netherlands, 2015.
- [13] U. R. Acharya, O. Faust, V. Sree et al., "Linear and nonlinear analysis of normal and CAD-affected heart rate signals," *Computer Methods and Programs in Biomedicine*, vol. 113, no. 1, pp. 55–68, 2014.
- [14] M. P. Tarvainen, J. P. Niskanen, J. A. Lipponen, P. O. Ranta-Aho, and P. A. Karjalainen, "Kubios HRV-heart rate variability analysis software," *Computer Methods and Programs in Biomedicine*, vol. 113, no. 1, pp. 210–220, 2014.
- [15] D. Cornforth, H. Jelinek, and M. Tarvainen, "A comparison of nonlinear measures for the detection of cardiac autonomic neuropathy from heart rate variability," *Entropy*, vol. 17, no. 3, pp. 1425–1440, 2015.
- [16] J. Li and X. Ning, "Dynamical complexity detection in short-term physiological series using base-scale entropy," *Physical Review E Statistical Nonlinear & Soft Matter Physics*, vol. 73, no. 5, p. 052902, 2006.
- [17] J. Li and X. B. Ning, "The base-scale entropy analysis of short-term heart rate variability signal," *Chinese Science Bulletin*, vol. 50, no. 12, pp. 1269–1273, 2005.
- [18] C. H. Bian, Q. L. Ma, J. Si et al., "Sign series entropy analysis of short-term heart rate variability," *Chinese Science Bulletin*, vol. 54, no. 24, pp. 4610–4615, 2009.
- [19] A. L. Song, "Optimum parameters setting in symbolic dynamics of heart rate variability analysis," *Acta Physica Sinica*, vol. 60, no. 2, pp. 307–351, 2011.
- [20] "PhysioNet/fantasia database," December 2016, <http://www.physionet.org/physiobank/database/fantasia/>.
- [21] A. H. Zhang, P. Wang, and Y. X. Chou, "Peak detection of pulse signal based on dynamic difference threshold," *Journal of Jilin University*, vol. 44, no. 3, pp. 847–853, 2014.
- [22] Y. X. Chou and A. H. Zhang, "Age-related alterations in the sign series entropy of short-term pulse rate variability," in *11th ICIC Conference on Advanced Intelligent Computing Theories and Applications*, pp. 723–729, Fuzhou, China, 2015.

Research Article

Patient-Specific Deep Architectural Model for ECG Classification

Kan Luo,^{1,2,3} Jianqing Li,^{2,4} Zhigang Wang,³ and Alfred Cuschieri³

¹*School of Information Science and Engineering, Fujian University of Technology, Xueyuan Road 3, Fuzhou 350118, China*

²*School of Instrument Science and Engineering, Southeast University, Sipailou 2, Nanjing 210096, China*

³*Institute for Medical Science and Technology, University of Dundee, Dundee DD2 1FD, UK*

⁴*School of Basic Medical Sciences, Nanjing Medical University, Longmian Avenue 101, Nanjing 211166, China*

Correspondence should be addressed to Jianqing Li; jqli@njmu.edu.cn

Received 23 October 2016; Revised 2 February 2017; Accepted 16 February 2017; Published 7 May 2017

Academic Editor: Ishwar K. Sethi

Copyright © 2017 Kan Luo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Heartbeat classification is a crucial step for arrhythmia diagnosis during electrocardiographic (ECG) analysis. The new scenario of wireless body sensor network- (WBSN-) enabled ECG monitoring puts forward a higher-level demand for this traditional ECG analysis task. Previously reported methods mainly addressed this requirement with the applications of a shallow structured classifier and expert-designed features. In this study, modified frequency slice wavelet transform (MFSWT) was firstly employed to produce the time-frequency image for heartbeat signal. Then the deep learning (DL) method was performed for the heartbeat classification. Here, we proposed a novel model incorporating automatic feature abstraction and a deep neural network (DNN) classifier. Features were automatically abstracted by the stacked denoising auto-encoder (SDA) from the transferred time-frequency image. DNN classifier was constructed by an encoder layer of SDA and a softmax layer. In addition, a deterministic patient-specific heartbeat classifier was achieved by fine-tuning on heartbeat samples, which included a small subset of individual samples. The performance of the proposed model was evaluated on the MIT-BIH arrhythmia database. Results showed that an overall accuracy of 97.5% was achieved using the proposed model, confirming that the proposed DNN model is a powerful tool for heartbeat pattern recognition.

1. Introduction

Cardiovascular diseases (CVDs) remain the leading cause of noncommunicable deaths worldwide. According to the latest World Health Organization (WHO) report, about 17.5 million people died from CVDs in 2012, accounting for 30% of all global deaths. The incidence of CVD deaths is predicted to rise to 23 million by 2030 [1]. Furthermore, the costs for CVD-related treatment including medication are substantial. The CVD-related cost in the low- and middle-income countries over the period 2011–2025 is estimated approximately 3.8 trillion U.S. dollars [2]. Many of these deaths and associated economic losses can be avoided by early detection and monitoring of patients' cardiac function. Electrocardiogram (ECG) is the standard and most efficient tool for CVD diagnosis [3], which captures the electrical activity of the heart from a human body surface, providing important information on cardiac functional abnormalities. The recent introduction of technology for

the wireless body sensor network- (WBSN-) enabled ECG has attracted the attention of both industry and academic researchers. WBSN-enabled ECG biosensors are seamlessly integrated into wearable fabric vest and can provide real-time continuous 7/24 monitoring and cardiac arrhythmia detection [4, 5]. This wearable WBSN-enabled ECG has the essential need for more efficient and robust data analysis methods for long monitoring of individual patients to ensure timely medical treatment or intervention. However, there are still some challenges in WBSN-enabled ECG signal analysis, particularly for the automatic detection of life-threatening arrhythmias [6].

Traditional methods exited risk of improper manual feature selection and limiting complex classification ability. There have been several reports on heartbeat classification [7–14]. Ince et al. [7] proposed an artificial neural network- (ANN-) based automated heartbeat classification model with morphological wavelet transform features, which achieved highly accurate heartbeat classification. Jiang and

Kong [8] used a block-based neural network model with Hermite transform coefficients and the selected temporal features for personalized ECG signal classification. Ye et al. [9] used morphological features and RR interval information in a support vector machine classifier for heartbeat classification. Alvarado et al. [10] proposed a novel compression sampler for feature extraction of ECG beats and then utilized linear discriminant analysis (LDA) for their classification. Chazal and Reilly [11] also used LDA as a classifier of differential temporal features including heartbeat morphology, heartbeat intervals, and R-R intervals. There are more studies on feature and classifier model [12–14]. Typically, the traditional classification models contain two layers at most, restricting their ability for complex classification tasks. Meanwhile, most of these models need manually designed features. Even the best classifier model will yield poor performance if important features are not selected. Additionally, most reported works focus on establishing common interpatient models for heartbeat classification. These methods were not using samples from the same patient for model training and testing [14]. However, WBSN-enabled ECG monitoring emphasized personalized heart status care; relatively common and patient-specific samples will help training a good performance model in individual heartbeat classification [15, 16]. Patient-specific method is more suitable in a WBSN-enabled ECG monitoring scenario.

Deep learning (DL) is an ideal and potential approach for the heartbeat classification of WBSN-enabled ECG, which can further improve classification performance. As a new research area of machine learning, DL has progressed rapidly since 2006 [17–19]. DL is based on algorithms for learning multiple levels of representation for the modeling complex relationship between data sets. Specifically, it is recognized as an effective method of abstracting hierarchical representation from unlabeled data; since higher-level features are defined by lower-level ones, the hierarchical feature representation of DL is referred to as “deep architecture” [20]. DL models by virtue of their multiple levels and nonlinear information processing provide much more efficient representations of complex functions, resulting in improved performance compared to shallow models [21]. Several studies have confirmed that deep architectural models exhibit excellent performance beating the existing traditional methods in challenge classification tasks [15, 22, 23]. However, there are still some aspects that need to be further studied when DL methods are used in traditional ECG analysis, such as the parameter of layers, size of neurons, and use of tanning samples.

Motivated by these challenges, we proposed a patient-specific heartbeat classification framework using time-frequency representation and a DL architectural model. Considering time-frequency technology is a powerful tool for characterizing the biosignals [24], and some of DL frameworks, such as stacked auto-encoder, convolutional neural networks (CNNs), and deep belief nets (DBNs) [22, 23, 25, 26], can be used to analyze ECG signal, while heartbeat time-frequency spectrograms are seen as images. A modified frequency slice wavelet transform (MFSWT) was used to

generate time-frequency representation of the heartbeat signal. Stacked denoising auto-encoder (SDA) model was chosen as the DL architectural model in our works. A SDA was pretrained by unlabeled MFSWT time-frequency spectrograms. Subsequently, a deep neural network (DNN) model was initialized by weights and bases of the trained SDA and was followed by two levels of fine-tuning. Particularly, after the second fine-tuning stage by using individual annotated heartbeat samples, the patient-specific DNN classifier was obtained. Validation of the proposed heartbeat classification method was performed on MIT-BIH arrhythmia database.

2. Data Description

MIT-BIH arrhythmia database [27] was selected as the data source, which is the most commonly used database for research in ECG signal processing. It consists of 48 annotated, 30 min ambulatory ECG records from 2 leads (II and modified V1, V2, V3, V4, or V5 leads) obtained from 47 subjects and sampled at 360 Hz per channel. Since lead II ECG is commonly used in ambulatory or WBSN-based ECG applications, these channel data were used in the current study.

The five heartbeat classes defined in the American National Standards Institute (ANSI) for the Advancement of Medical Instrumentation (AAMI) standard (IEC 60601-2-47:2012) [28] are (i) normal beat (N), (ii) supraventricular ectopic beat (SVEB or S), (iii) ventricular ectopic beat (VEB or V), (iv) fusion beat (F), and (v) paced beats or unknown beat Q. However, according to the annotation file from PhysioNet (<http://www.physionet.org/>), there are 15 beat types in MIT-BIH database. Table 1 shows the group method for mapping the MIT-BIH heartbeat classes into AAMI classes.

ANSI/AAMI EC57:2012 recommends exclusion of records containing paced beat records (numbers 102, 104, 107, and 217) for classifiers' evaluation. The number of 33 unclassified beats is less than 0.03% of the whole data samples, which can lead to model overfitting. Thus, in this study, Q type includes paced beats and unclassified beats were excluded. Following these exclusions, the heartbeat data samples were regrouped into four types (N, S, V, and F) according to the AAMI standard. Thus, the remaining 44 nonpacemaker records without the unclassified beats were divided into equal training and testing sets [10]. The training set consisted of record numbers 101, 106, 108, 109, 112, 114, 115, 116, 118, 119, 122, 124, 201, 203, 205, 207, 208, 209, 215, 220, 223, and 230; and the testing set consisted of record numbers 100, 103, 105, 111, 113, 117, 121, 123, 200, 202, 210, 212, 213, 214, 219, 221, 222, 228, 231, 232, 233, and 234.

The objective of the proposed framework is to classify heartbeats into N, S, V, and F classes. Clinically, supraventricular ectopic beats (SVEB) and ventricular ectopic beats (VEB) are two critically abnormal and serious heartbeats, and the performance of the classifiers also was elevated by testing S and V heartbeat classification [9, 10, 15].

TABLE 1: Heartbeat classes given by the MIT-BIH database along with the regrouping defined by the AAMI standard [10, 28].

MIT-BIH class	MIT-BIH number	AAMI groups	Number of samples
Normal beat	1		
Left bundle branch block beat	3		
Right bundle branch block beat	2	N: beats not found in the classes S, V, F, and Q	90631
Atrial escape beats	34		
Nodal (junctional) escape beat	11		
Atrial premature beats	8		
Aberrated atrial premature beats	4	S: supraventricular ectopic beats	2781
Nodal (junctional) premature beats	7		
Supraventricular premature beats	9		
Premature ventricular contraction	5	V: ventricular ectopic beats	7236
Ventricular escape beat	10		
Fusion of ventricular and normal beat	6	F: fusion beats	803
Paced beat	12		
Fusion of paced and normal beat	38	Q: paced beats or unclassified beats	8010
Unclassified beat	13		33

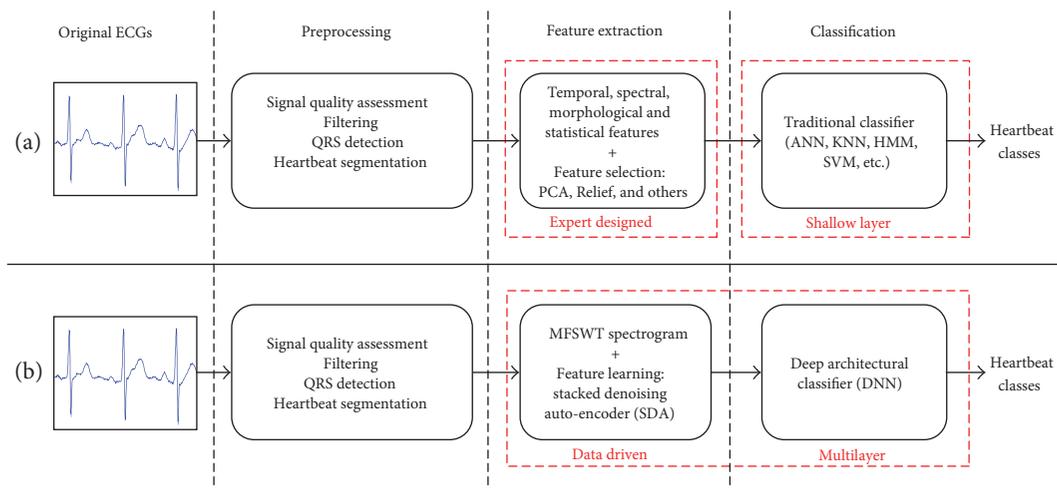


FIGURE 1: Schematic illustration of two heartbeat classification frameworks: (a) traditional framework and (b) the framework of proposed DL architectural model with time-frequency representation.

3. DL Architectural Model-Based Heartbeat Classification

The comparison between traditional and proposed heartbeat classification frameworks is shown in Figure 1. Both include three steps: preprocessing, feature extraction, and classification.

The key differences, which distinguish the new framework, are as follows:

- (i) The use of MFSWT to generate time-frequency spectrogram for using deep learning methods
- (ii) Adoption of stacked denoising auto-encoder (SDA) for automatic abstraction of features from MFSWT spectrogram (instead of human experts),

to avoid the associated risk of improper manual feature selection

- (iii) Integration of data-driven feature extraction and DL architectural classification into a single learning framework for improved heartbeat classification.

3.1. Preprocessing of ECG Signal. Preprocessing of ECG signal includes signal quality assessment (SQA) [29], denoising, QRS detection, heartbeat segmentation, and calculation of time-frequency spectrogram. Since the present study concerns the illustration of DL-based model for heartbeat classification, the SQA step was omitted, except for the removal of low signal quality heartbeats and their pre- and after beats from the data set. Power and high-frequency noise

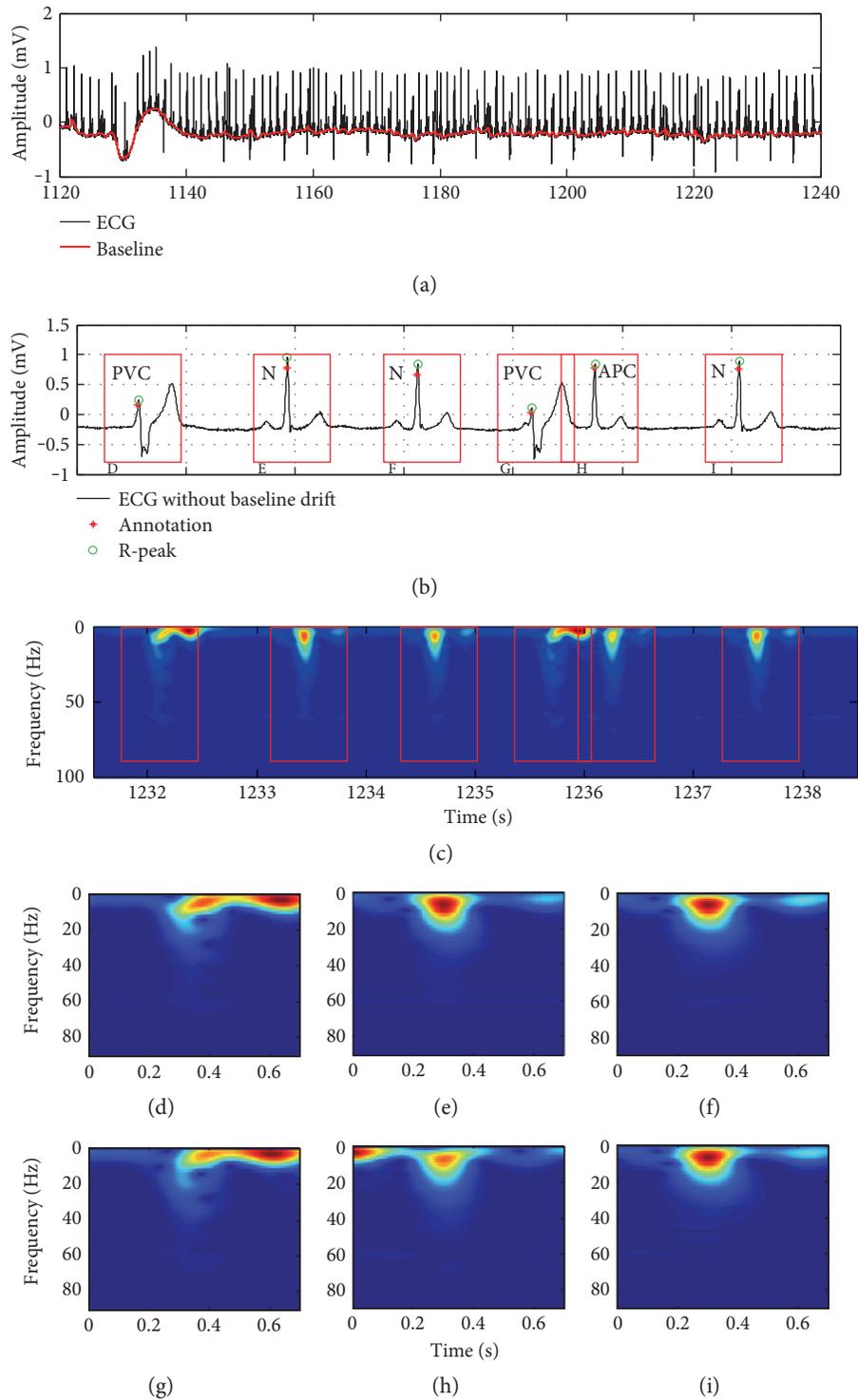


FIGURE 2: ECG preprocessing, sample heartbeat waveform from number 201 record lead II, including normal (N), premature ventricular contraction (PVC), and atrial premature contraction (APC) AAMI heartbeat classes. (a) Baseline drift elimination; (b) QRS-complex detection; (c) MFSWT spectrogram corresponding to the waveform in (b); (d)~(i) MFSWT spectrograms corresponding to each segmented heartbeat by 700 ms windows in (b). In the absence of other special instructions, all spectrograms were normalized to [0, 1].

and baseline drift were eliminated by using two median filters with window sizes of 200 ms and 600 ms [10] as shown in Figure 2(a). Although many algorithms are used for QRS

detection [30], the derivative-based algorithm with a characteristic steep slope of the QRS complex was chosen to detect R-peaks in view of its high accuracy. Figure 2(b) provides

details of the comparison of the detected R-peaks and the annotation points where all six R-peaks can be identified. 700 ms windows (red boxes in Figure 2), centered at the detected R-peaks (300 ms before and 400 ms after), were used to segment each heartbeat. As shown in Figures 2(b) and 2(c), the normal, premature ventricular contraction (PVC), and atrial premature contraction (APC) beats are segmented, and the corresponding MFSWT spectrograms were shown in Figures 2(b)~2(i).

3.2. Modified Frequency Slice Wavelet Transform. Frequency slice wavelet transform (FSWT) is essentially an extension of the short-time Fourier transform in frequency domain [31]. FSWT achieves good performance in transient vibration response analysis and damping modal identification [32]. However, low-frequency biosignals are not well represented by original FSWT due to its defined window size of frequency slice function (FSF) changes sharply in low-frequency area. To accurately locate the components of heartbeat signal in time-frequency plane, “modified frequency slice wavelet transform (MFSWT)” was proposed. MFSWT follows the rules of producing time-frequency representation from the frequency domain but incorporating a set of bound signal-adaptive FSFs which serve as a set of dynamic frequency filters, which can well represent signal in time-frequency domain.

Assume that $\hat{f}(u)$ is Fourier transform of $f(t)$. The MFSWT is expressed in frequency domain as

$$W_f(t, \omega) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{f}(u) \hat{p}^* \left(\frac{u - \omega}{q(\hat{f}(u))} \right) e^{iut} du, \quad (1)$$

where t and ω are observed time and frequency, respectively. “*” represents the conjugation operator. $\hat{p}(x) = e^{-x^2/2}$ is selected as the FSF in (1), and $\hat{p}(0) = 1$. The shape of FSF is like an inverted bell. q is defined as a scale function of $\hat{f}(u)$ and enables the transform with signal-adaptive property.

$$q = \delta + \text{sign}(\nabla |\hat{f}(u)|), \quad (2)$$

where δ is the frequency which corresponds to the maximum $|\hat{f}(u)|$. $\nabla(\cdot)$ is differential operators, and $\text{sign}(\cdot)$ means signum function, which returns 1 if the input is greater than zero, 0 if it is zero, or -1 if it is less than zero. According to (2), q changes slowly with $|\hat{f}(u)|$ and generates FSFs as a function of $|\hat{f}(u)|$. As dynamic frequency filters, FSFs were used to estimate the energy distribution of different frequency bands. Similar to the scale used for different size objects in microscopy, narrow window size of FSFs corresponds to the small values of $|\hat{f}(u)|$ and wide window size of FSFs corresponds to the large values of $|\hat{f}(u)|$. Due to the effect of the adaptive FSF, energy of signal components with large $|\hat{f}(u)|$ can be reinforced in time-frequency spectrogram. Taking advantages of the slowly changing FSFs and energy enhancement of frequency filtering, MFSWT achieves accurate time-frequency representation of the heartbeat signal.

FSFs in MFSWT meet $\hat{p}(0) = 1$ according to the proof in [32]. The reversed MFSWT can be expressed by

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} W_f(\tau, \omega) e^{i\omega(t-\tau)} d\tau d\omega. \quad (3)$$

In this study, MFSWT is used as a tool to generate heartbeat spectrogram for SDA feature extraction. Figure 3 shows an example. The comparison between the original and the reconstructed heartbeat signal is shown in Figure 3(a). Percentage root-mean-square difference (PRD) equal to zero indicates that signal can be exactly reconstructed by the reversed MFSWT from the spectrogram. Heartbeat time-frequency spectrograms of MFSWT, Wigner-Ville distribution (WVD), continuous wavelet transform (CWT), and FSWT are shown in Figures 3(b)~3(e), respectively. As outlined in Figure 3(b), accurate locations of P-, QRS-, and T-waves and power noise components in time-frequency spectrogram, which correspond well to the signal in the time domain, were achieved by using the MFSWT. In comparison to WVD, CWT, and FSWT, the spectrogram of MFSWT has better interpretability. Additionally, without troublesome parameter selection, MFSWT is easier to use than other methods. The goal of machine learning is to replace humans for pattern recognition, considering MFSWT spectrograms are more readily accepted by clinicians, which are adopted as time-frequency images for DL classification in the present work.

3.3. Stacked Denoising Auto-Encoder. Auto-encoder (AE) is the basic unit of stacked denoising auto-encoder, which can capture the maximum possible information contained in a given sample, while minimizing the reconstruction error rate. A basic encoder is a function that takes an input $V \in R^{d_v}$ to a hidden representation $h \in R^{d_h}$, which can be stated as

$$h = s_{\text{AE}}(WV + b), \quad (4)$$

where W is a $d_v \times d_h$ weight matrix, $b \in R^{d_h}$ is a bias, and s_{AE} is a nonlinear logistic sigmoid activation function $s(x) = 1/(1 + e^{-x})$.

The decoder maps the output of the hidden layer h back to the reconstruction \hat{V} by a similar transformation.

$$\hat{V} = s_{\text{DC}}(W'h + b'), \quad (5)$$

where s_{DC} is also a logistic sigmoid activation function and $W' \in R^{d_h \times d_v}$ and $b' \in R^{d_v}$ are two parameters of decoder. Let $W' = W^T$ be referred to as tied weights.

The parameters of AE are optimized if the average reconstruction error is minimized, which corresponds to minimize the following objective function:

$$O(\{W, b, b'\}) = \sum_{V \in D_n} L(V, \hat{V}), \quad (6)$$

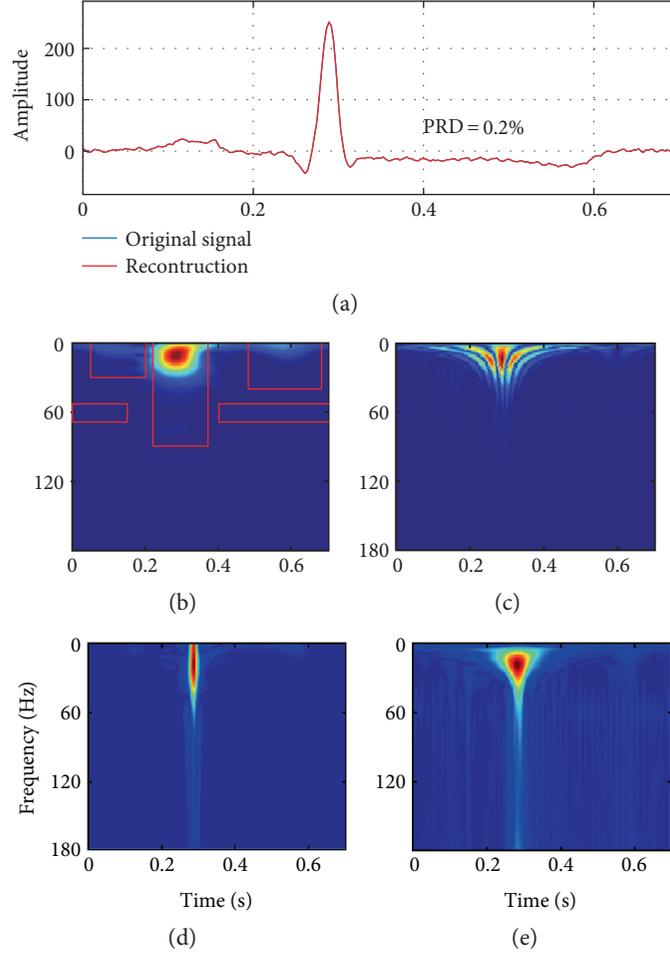


FIGURE 3: Heartbeat signals and corresponded spectrograms: (a) original heartbeat and reconstructed heartbeat; (b) MFSWT; (c) CWT @ mexh wavelet (d) WVD; and (e) FSWT @ $\kappa = 4$. All spectrograms were normalized to $[0, 1]$.

where L is a reconstruction error. The function of cross-entropy loss can be used as L if input samples D_n for training are in $[0, 1]$.

$$L(V, \hat{V}) = -\sum_{k=1}^{d_V} [V_k \log(\hat{V}_k) + (1 - V_k) \log(1 - \hat{V}_k)]. \quad (7)$$

For robust feature extraction, Vincent et al. [26] proposed denoising auto-encoders (DAE). DAE is trained to reconstruct the input from a corrupted version of the input. Thus, the model has an antinoise property. The objective function of DAE can be written as in (8), which can be optimized by the stochastic gradient descent method [21].

$$O(\{W, b, b'\}) = \sum_{\tilde{V} \in D_n} E_{\tilde{V} \sim q(\tilde{V}|V)} [L(V, \hat{V})]. \quad (8)$$

In (8), E is the expectation, and the corrupted version \tilde{V} of V produces $q(\tilde{V}|V)$ by a corruption process. Stochastic corruption process, which randomly sets a fraction P of inputs to zero, is used as the corruption process in this work, and an example is shown in Figure 4. The parameter P controls the degree of regulation.

SDA is achieved by stacking multiple DAEs with their corresponding decoders. Here, the SDA was used for primary feature extraction and initialization of deep neural network weights. The schematic view of a three-layered SDA is shown in Figure 5. Each layer of SDA is a DAE, and unsupervised layer-by-layer training minimizes the reconstruction error of each DAE. Figure 5(b) shows part of first layer weightings of unsupervised trained SDA model. Details of weights marked with a red box in Figure 5(b) are shown in Figure 5(d), which demonstrates that spectral features of heartbeat are captured by the trained SDA model and are stored as the weights. The extracted features of number 100 record first heartbeat are shown in Figure 5(c). In this case, the number of final abstracted features is 256, and the output of the bottleneck layer is sparsity, which helps the subsequent discriminant classification.

3.4. DNN Classifier. Since heartbeat classification is a multiple output task, a softmax regression layer with “N, S, V, and F” output is added on top of the bottleneck layer as shown in Figure 5(a). Then, the SDA encoder and the softmax layer are combined to form a DNN classifier. $W\{i\}$, b_i , and $h\{i\}$ are the weights, bases, and outputs of each hidden layers,

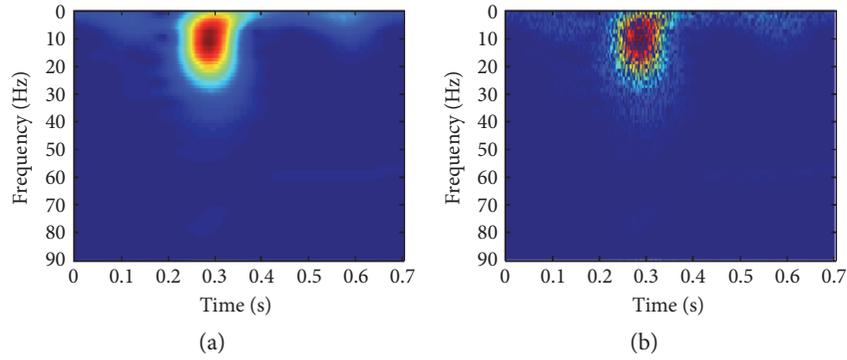


FIGURE 4: Stochastic corruption process in SDA model training. (a) Original spectrogram. (b) Result of stochastic corruption process ($P = 0.5$).

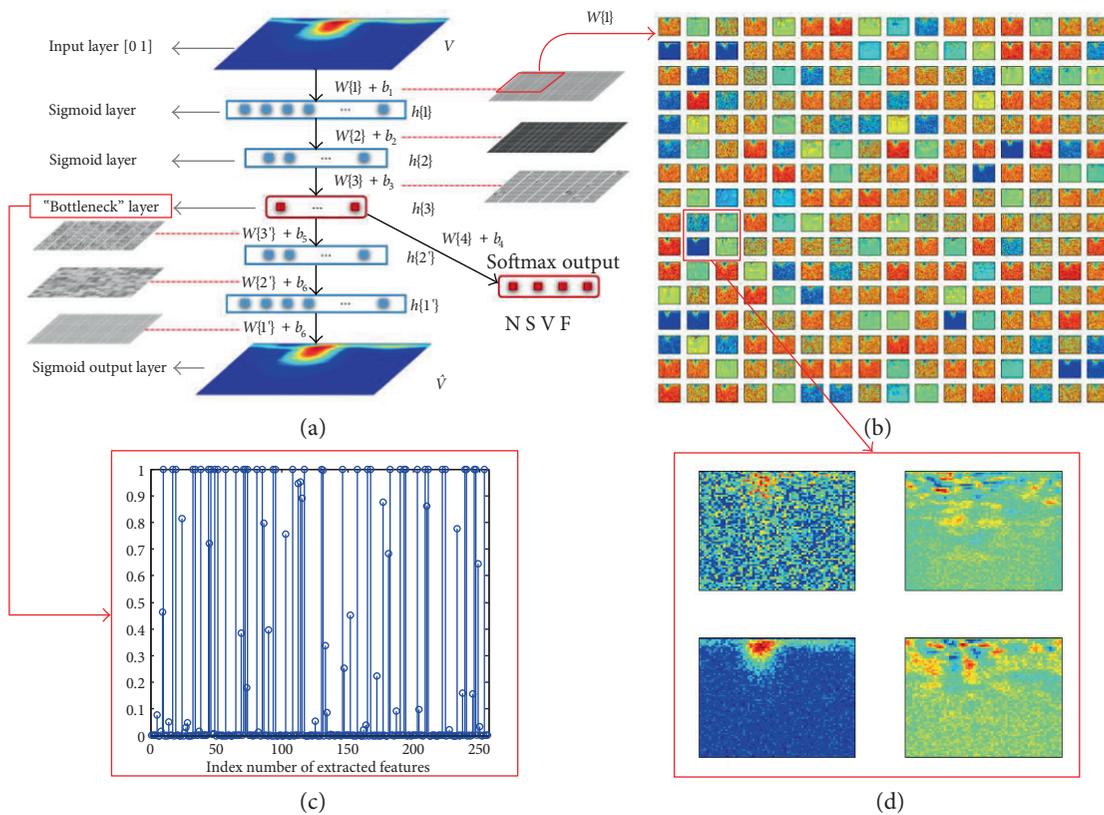


FIGURE 5: Schematic view of SDA for feature learning and DNN-based classification: (a) three layers of SDA and four layers of DNN; (b) part of the first layer weights of unsupervised trained SDA model; (c) extracted features of number 100 record first heartbeat; and (d) details of the weights marked with a red box.

respectively. The heartbeat class can be identified when the final hidden layer output feeds into the last softmax layer.

4. Patient-Specific DNN Classifier Training

Due to existing interpatient signal variability, different patients' beats of the same class are different; it is difficult to train a common interpatient model, which can perfectly classify heartbeats from other patients. This problem can be overcome by the use of patient-specific technique [7, 8, 11]. Here, the patient-specific approach was adopted in the

present study. We used a small beginning part of individual samples in model training to maximize performance in individual heartbeat classification.

As shown in Figure 6, the whole patient-specific DNN training consists of three sequential stages: (i) SDA model training, (ii) common interpatient classifier training, and (iii) patient-specific classifier training. The first stage is training the SDA model. Its purpose is to estimate initial parameters of a DNN classifier from the trained SDA. The second stage is referred as fine-tuning [16–18]. After using encoder layers of SDA and a softmax layer forming a DNN

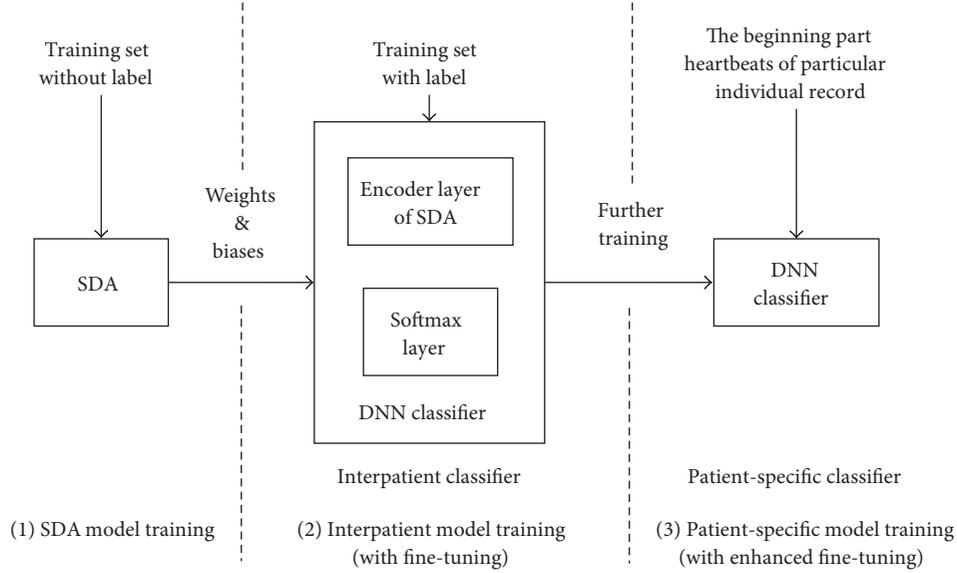


FIGURE 6: The workflow of patient-specific DNN training.

classifier, fine-tuning is used to minimize the heartbeat classification prediction error with samples of training set listed in Section 2. After fine-tuning, an interpatient DNN classifier is achieved. In the third stage, newly annotated heartbeats of the testing set (i.e., first 300 beats) are used for further fine-tuning based on the trained interpatient classifier. In this stage, the same algorithm in the second stage is adopted but with different train samples. The program will stop until classification performance has no further improvement or the maximum number of iterations is reached. At this stage, personalized heartbeat DNN classifiers can be generated. The last stage called as enhanced fine-tuning in our work aims to further parameter adjustment of the trained interpatient classifier to address the individual variations.

5. Experimental Results and Discussion

Evaluation of the trained classifier by the AAMI standard was done with MIT-BIH arrhythmia database. In preprocessing, all MFSWT spectrograms of each heartbeat were normalized to $[0, 1]$, and records in the database were divided into equal training and testing sets as described previously. The evaluation results were compared to those reported by other systems [7–14].

Four widely used metrics, that is, sensitivity (SE), specificity (SP), positive predictive value (PPV), and accuracy (ACC), were used (and defined next) for the assessment of classification performance:

$$\begin{aligned}
 SE_i &= \frac{TP_i}{(TP_i + FN_i)}, \\
 SP_i &= \frac{TN_i}{(TN_i + FP_i)}, \\
 PPV_i &= \frac{TP_i}{(TP_i + FP_i)}, \\
 ACC_i &= \frac{(TP_i + TN_i)}{(TP_i + TN_i + FP_i + FN_i)},
 \end{aligned} \tag{9}$$

where TP_i (true positive) equals the number of i th class heartbeats correctly classified, TN_i (true negative) is the number of heartbeats not belonging to i th class and not classified in the i th class, FP_i (false positive) equals the number of heartbeats erroneously classified into i th class, and FN_i (false negative) equals the number of i th class heartbeats classified in a different class. SE_i and SP_i , respectively, reflect the classifier's sensitivity and specificity in i th prediction, and PPV_i defines the percentage of positive correct predictions. ACC_i is the ratio between all correctly and incorrectly predicted heartbeats. Since the data set is imbalanced, the geometric mean (g -mean) [13], estimated by the geometrical mean of heartbeat class predicted sensitivities, was also selected as a performance measure.

$$g\text{-mean}_x = \sqrt{x^+ \cdot x^-}, \tag{10}$$

where x^+ and x^- are the predicted SE or PPV of the positive and negative classes, respectively.

Grid searching was used to identify the optimal parameters. The number of layers was changed from 0 to 3. Considering the feature abstract characteristic of SDA, it is a good idea to make the size of the hidden layer output smaller than the input size for each AE or DAE. Seven conditions of the number of neurons in the first hidden layer are A (64), B (128), C (256), D (512), E (1024), F (2048), and G (4096). The number of neurons of next layer was set as half of the current layer if SDA is a multilayer structure. After encoder layers, softmax layer maps the abstracted features to four types of heartbeats. For example, in a 3-layer SDA model, the number of the first layer is 512 and then the numbers of neurons of each layer in DNN model are 512-256-128-4. The beginning 300 beats of test records were used for personalized classifier training. Experimental results were based on the remaining heartbeats of testing sets and shown in Figure 7. The SVEB and VEB classification results from other works are shown in Table 4 as benchmarks in Figure 7, and

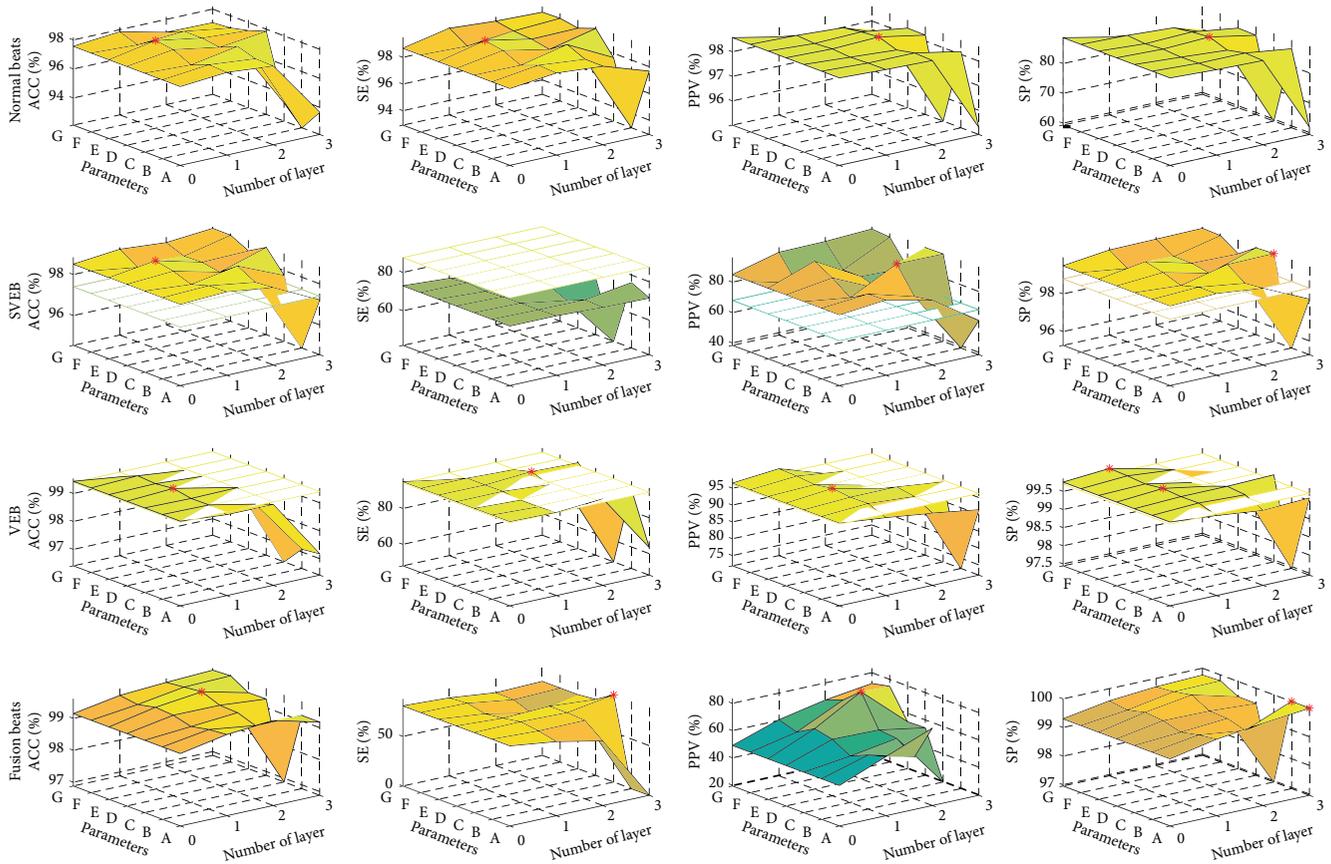


FIGURE 7: Heartbeat classification results by patient-specific models with different parameters. The best results of SVEB and VEB classification from other works in Table 4 are shown as benchmarks; the best results of the proposed models are marked by asterisks. A~G represent the sizes of the first hidden layer which are 64, 128, 256, 512, 1024, 2048, and 4096, and the number of neurons of the next layer was set as half of the current layer if SDA is a multilayer structure.

the best results of our work are marked by asterisks. With proper parameters, the proposed method outperforms the benchmarks in all other measures except SE of SVEB. The experimental results confirm the efficiency of the proposed patient-specific deep architectural framework. However, to maintain good stable performance, the size of neurons can be selected with a narrower range while the number of layers is added. Based on the classification results and generalization risk consideration, it is advised that one hidden layer within 1024~2048 neurons could achieve acceptable heartbeat classification.

0 to 300 heartbeats of individual samples were used in patient-specific models training to explore the relationship between performance improvement and the added number of personal samples. Experiments were based on a trained interpatient DNN classifier, which includes one encoder layer with 1024 neurons and a softmax layer. The relationship is shown in Figure 8. Accuracy, SE, and PPV increase as the number of individual heartbeats increase. Accuracy and measures of SVEB and VEB become stable when the number of beats exceeds 80. Because major F beats of the testing set are in number 213 record and appear after 2 min, the best SE and PPV of F were achieved in the range of 240~300 beats. The proposed method would produce an interpatient classifier if no individual samples were used in

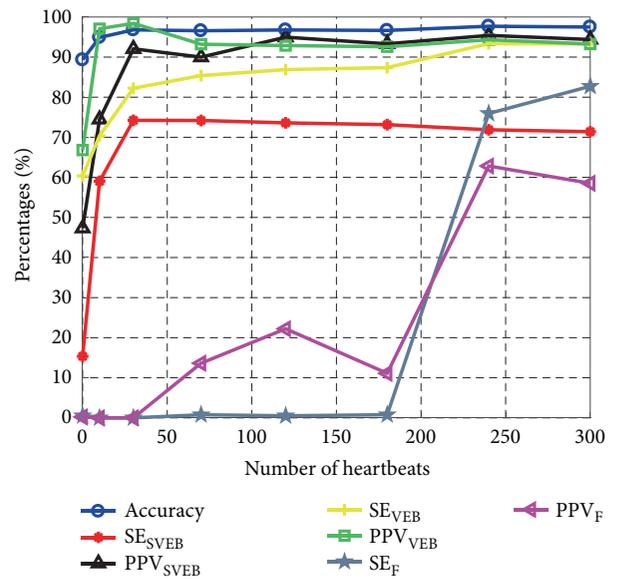


FIGURE 8: Relationships between classification performance and the number of individual samples were used for model training.

TABLE 2: Performance of deep architectural classifier on testing set with reference (performance is determined by interpatient scenario; all testing sets were used).

		Predicted				Total	SE (%)
		N	S	V	F		
True	N	41873	300	947	810	43930	95.3
	S	1520	282	9	25	1836	15.4
	V	1240	13	1943	23	3219	60.4
	F	376	1	9	2	388	0.5
	Total	45009	596	2908	860	49373	42.9
	PPV (%)	93.0	47.3	66.8	0.2	51.8	Accuracy = 89.3%
Accuracy = $(TP_N + TP_S + TP_V + TP_F)/\text{number of testing heartbeats}$							

TABLE 3: Performance of deep architectural classifier on testing set with reference (performance is determined by trained personalized classifier, testing set excluding the first 300 heartbeats).

		Predicted				Total	SE (%)
		N	S	V	F		
True	N	37622	68	175	119	37984	99.0
	S	448	1143	7	3	1601	71.4
	V	106	0	2644	85	2835	93.3
	F	52	0	9	292	353	82.7
	Total	38228	1211	2835	499	42773	86.6
	PPV (%)	98.4	94.4	93.3	58.5	86.1	Accuracy = 97.5%
Accuracy = $(TP_N + TP_S + TP_V + TP_F)/\text{number of testing heartbeats}$							

fine-tuning. Confusion matrix of interpatient heartbeat classification (Table 2) demonstrates an 89.3% heartbeat classification accuracy. Although SE and PPV of N reached 95.3% and 93.0%, respectively, the other measure values of S, V, and F are low. From the results, it can infer that the constructed interpatient deep architectural classifier has poor ability to process interpatient signal variability.

The overall classification assessment results obtained by considering each of the four classes (“N, S, V, and F”) are summarized in Table 3 when 300 beats are used. The numbers of correct prediction of N, S, V, and F beats are 37,622, 1143, 2644 and 292, respectively. The overall accuracy of the heartbeat classification reached 97.5%. The results demonstrate that performance of the classifier can be efficiently improved by using relatively small individual samples, the patient-specific classifier can well cope with interpatient variations.

Comparisons of the proposed model and the state-of-the-art methods [7–14] were summarized in Tables 4 and 5, which demonstrate that the proposed model achieves better recognition in patient-specific heartbeat classification scenario, with ACCs of SVEB and VEB rates of 98.8% and 99.1%, respectively. Specifically, a PPV of SVEB of 94.4% indicates that the proposed model has the high-level capability of identifying SVEB. The 71.4% SE of SVEB is superior to most other reported studies (see Table 4). The results presented in Table 4 confirm that the proposed model can satisfactorily identify SVEB and VEB. Evaluation results for all four-class heartbeat recognitions are outlined in Table 5, and classification confusion matrix in Table 3. The

results relating to SE and PPV of all types are close to or surpass those obtained with current state-of-the-art methods. Accuracy, g_mean_{SE} , and g_mean_{PPV} reached 97.5%, 85.9%, and 84.4%, respectively. Similarly, using 1-D CNNs for patient-specific ECG classification that also achieved superior performance was reported [15]. These make us have good reasons to believe that deep learning methods have great potential in patient-specific ECG signal analysis.

The reasons for the superior performance of the proposed method are multifactorial. In the first instance, MFSWT transforms ECG signal from time domain to time-frequency domain. The distinguishable differences of heartbeat signals are well preserved in MFSWT spectrogram, facilitating both SDA feature extraction and the following personalized DNN classifier training. Secondly, using data-driven SDA instead of expert human involvement could avoid the improper feature extraction for classification. Thirdly, deep architectural patient-specific classifier improves the accuracy of individual heartbeat prediction. The main limitation of the present study is that it needs extra individual annotated beats. According to the experimental results, patient-specific samples are important to the proposed method. As the results shown in Figure 8, not using individual samples in the model training, the system may entirely or partially fail to classify S, V, and F beats. However, using a few annotated individual beats is possible in clinical practice. Once the patient-specific classifier has been trained, it can continually provide accurate heartbeat classification services for individual patients in WBSN-enabled long-term automatic ECG monitoring scenario. Except for individual training

TABLE 4: Classification metrics compared to the state-of-the-art SVEB and VEB classification (percentage, %).

Methods	SVEB				VEB			
	ACC	SE	PPV	SP	ACC	SE	PPV	SP
Proposed (a)	98.8	71.4	94.4	99.8	99.1	93.3	93.3	99.5
Kiranyaz et al. [15]*	96.4	64.6	62.1	98.6	98.6	95	89.5	98.1
Chazal and Reilly [11]*	95.9	87.7	47.0	96.2	99.4	94.3	96.2	99.7
Jiang and Kong [8]*	96.6	50.6	68.0	98.8	97.7	86.6	89.4	98.9
Ince et al. [7]*	97.3	63.5	53.7	98.3	98.0	84.6	86.7	99.0
Proposed (b)	96.2	15.4	47.3	99.3	95.5	60.4	66.8	97.9
Mar et al. [12]	93.3	83.2	33.5	93.7	97.4	86.8	75.9	98.1
Alvarado et al. [10]	97.0	86.2	56.7	97.5	99.1	92.4	93.4	99.5
Ye et al. [9]	97.4	56.4	55.1	98.6	94.6	84.7	59.5	95.4
Zhang et al. [13]	93.3	79.1	36.0	93.9	98.6	85.5	92.7	99.5

*Patient-specific method: require expert intervention.

(a) indicates the patient-specific heartbeat classification scenario. Classifiers were trained by using the first 300 beats of individual patient.

(b) indicates the interpatient heartbeat classification scenario.

TABLE 5: Classification metrics compared to the state-of-the-art four-class heartbeat recognitions (percentage, %).

Methods	Accuracy	g-mean		N		S		V		F	
		SE	PPV	SE	PPV	SE	PPV	SE	PPV	SE	PPV
Proposed (a)	97.5	85.9	84.4	99.0	98.4	71.4	94.4	93.3	93.3	82.7	58.5
Chazal and Reilly [11]*	93.9	87.2	59.8	94.3	99.4	87.7	47.0	94.3	96.2	74.0	29.1
Jiang and Kong [8]*	94.5	62.7	83.8	98.7	96.2	50.6	68.0	86.6	89.4	35.8	84.2
Ince et al. [7]*	93.6	74.5	76.9	97.0	97.0	62.1	56.7	83.4	86.5	61.4	73.4
Proposed (b)	89.3	14.6	16.2	95.3	93.0	15.4	47.3	60.4	66.8	0.5	0.2
Mar et al. [12]	89.0	79.3	45.2	94.2	99.2	86.2	56.7	92.4	93.4	66.4	17.7
Alvarado et al. [10]	93.6	84.0	55.2	94.2	99.2	86.2	56.7	92.4	93.4	66.4	17.7
Ye et al. [9]	88.2	62.6	37.0	90.0	98.2	56.4	55.1	84.7	59.5	35.8	5.8
Zhang et al. [13]	88.3	86.7	46.2	88.9	99.0	79.1	36.0	85.5	92.8	93.8	13.7

*Patient-specific method: require expert intervention.

(a) indicates the patient-specific heartbeat classification scenario. Classifiers were trained by using the first 300 beats of individual patient.

(b) indicates interpatient heartbeat classification scenario.

samples, the computation may limit the usage of the proposed method. The personalized DNN model training requires the extensive computation (~1 hour @ Intel i7 4720H, 32 GB RAM, GTX970M laptop), though small computation is needed for prediction (<0.02 ms @ Intel i7 4720H, 32 GB RAM, GTX970M laptop). However, the envisaged significant expansion of machine and network processing abilities will facilitate increased usage of the proposed method.

6. Conclusions

A novel framework based on time-frequency representation and patient-specific DL architectural model for heartbeat classification is proposed. The model performance was validated by evaluation on MIT-BIH arrhythmia database. The results confirmed an overall superior performance with an accuracy of 97.5%. Superior classification results have been achieved by using one encoder layer of SDA with 1024 neurons and one softmax-formed DNN model. The advantages

of the proposed framework include its automatic feature extraction, patient-adaptive nature, and low classification error. The proposed patient-specific DNN classifier is simple and effective. Therefore, it is a potential choice for individual automatic heartbeat classification used in WBSN-enabled ECG monitoring.

Conflicts of Interest

The authors declare that they have no competing interest.

Acknowledgments

This research was supported in part by the National Natural Science Foundation of China (NNSFC) under Grant 61601124 and Grant 61571113, the research project of Fujian University of Technology under Grant GY-Z160058, and China Scholarship Council.

References

- [1] World Health Organization, "Cardiovascular disease," 2013, http://www.who.int/cardiovascular_diseases/en/index.html.
- [2] World Health Organization, "From burden to "best buys": reducing the economic impact of non-communicable disease in low-and middle-income countries," *Program on the Global Demography of Aging*, 2011, www.who.int/nmh/publications/best_buys_summary/en/.
- [3] D. W. Smith, D. Nowacki, and J. K. Li, "ECG T-wave monitor for potential early detection and diagnosis of cardiac arrhythmias," *Cardiovascular Engineering*, vol. 10, no. 4, pp. 201–206, 2010.
- [4] K. Luo, J. Li, and J. Wu, "A dynamic compression scheme for energy-efficient real-time wireless electrocardiogram biosensors," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 9, pp. 2160–2169, 2014.
- [5] T. Chen, E. B. Mazomenos, K. Maharatna, S. Dasmahapatra, and M. Niranjani, "Design of a low-power on-body ECG classifier for remote cardiovascular monitoring systems," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 1, pp. 75–85, 2013.
- [6] A. Amann, R. Tratnig, and K. Unterkofler, "Detecting ventricular fibrillation by time-delay methods," *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 1, pp. 174–177, 2007.
- [7] T. Ince, S. Kiranyaz, and M. Gabbouj, "A generic and robust system for automated patient-specific classification of ECG signals," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 5, pp. 1415–1426, 2009.
- [8] W. Jiang and S. G. Kong, "Block-based neural networks for personalized ECG signal classification," *IEEE Transactions on Neural Networks*, vol. 18, no. 6, pp. 1750–1761, 2007.
- [9] C. Ye, B. V. K. V. Kumar, and M. T. Coimbra, "Heartbeat classification using morphological and dynamic features of ECG signals," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 10, pp. 2930–2941, 2012.
- [10] A. S. Alvarado, C. Lakshminarayan, and J. C. Principe, "Time-based compression and classification of heartbeats," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 6, pp. 1641–1648, 2012.
- [11] P. de Chazal and R. B. Reilly, "A patient-adapting heartbeat classifier using ECG morphology and heartbeat interval features," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 12, pp. 2535–2543, 2006.
- [12] T. Mar, S. Zauneder, J. P. Martínez, M. Llamedo, and R. Poll, "Optimization of ECG classification by means of feature selection," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 8, pp. 2168–2177, 2011.
- [13] Z. Zhang, J. Dong, X. Luo, K. Choi, and X. Wu, "Heartbeat classification using disease-specific feature selection," *Computers in Biology and Medicine*, vol. 46, pp. 79–89, 2014.
- [14] H. Huang, J. Liu, Q. Zhu, R. Wang, and G. Hu, "A new hierarchical method for inter-patient heartbeat classification using random projections and RR intervals," *Biomedical Engineering Online*, vol. 13, no. 1, p. 90, 2014.
- [15] S. Kiranyaz, T. Ince, and M. Gabbouj, "Real-time patient-specific ECG classification by 1-D convolutional neural networks," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 3, pp. 664–675, 2016.
- [16] M. Llamedo and J. P. Martínez, "An automatic patient-adapted ECG heartbeat classifier allowing expert assistance," *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 8, pp. 2312–2320, 2012.
- [17] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [18] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in Neural Information Processing Systems 19 (NIPS' 06)*, pp. 153–160, MIT Press Cambridge, MA, USA, 2007.
- [19] C. P. Marc Aurelio Ranzato, S. Chopra, and Y. LeCun, "Efficient learning of sparse representations with an energy-based model," in *Advances in Neural Information Processing Systems 19 (NIPS' 06)*, pp. 1137–1144, MIT Press Cambridge, MA, USA, 2007.
- [20] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.
- [21] L. Deng and D. Yu, "Deep learning: methods and applications," *Foundations and Trends® in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.
- [22] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [23] G. Hinton, L. Deng, D. Yu et al., "Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.
- [24] B. Ergen, Y. Tatar, and H. O. Gulcur, "Time-frequency analysis of phonocardiogram signals using wavelet transform: a comparative study," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 15, no. 4, pp. 371–381, 2012.
- [25] A. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.
- [26] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol, "Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion," *The Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.
- [27] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.
- [28] IEC, "IEC International Standard 601-2-47 Ed. 2 Medical electrical equipment part 2-47: particular requirements for the basic safety and essential performance of ambulatory electrocardiographic systems," 2012, https://en.wikipedia.org/wiki/IEC_60601.
- [29] C. Liu, P. Li, C. Di Maria, L. Zhao, H. Zhang, and Z. Chen, "A multi-step method with signal quality assessment and fine-tuning procedure to locate maternal and fetal QRS complexes from abdominal ECG recordings," *Physiological Measurement*, vol. 35, no. 8, pp. 1665–1683, 2014.
- [30] B. U. Kohler, C. Hennig, and R. Orglmeister, "The principles of software QRS detection," *IEEE Engineering in Medicine and Biology Magazine*, vol. 21, no. 1, pp. 42–57, 2002.

- [31] Z. H. Yan, A. Miyamoto, Z. W. Jiang, and X. L. Liu, "An overall theoretical description of frequency slice wavelet transform," *Mechanical Systems and Signal Processing*, vol. 24, no. 2, pp. 491–507, 2010.
- [32] Z. H. Yan, A. Miyamoto, and Z. W. Jiang, "Frequency slice algorithm for modal signal separation and damping identification," *Computers & Structures*, vol. 89, no. 1-2, pp. 14–26, 2011.