Complexity

Complexity in Medical Informatics

Lead Guest Editor: Panayiotis Vlamos
Guest Editors: Ilias Kotsireas and Dimitrios Vlackakis

Complexity in Medical Informatics

Complexity

Complexity in Medical Informatics

Lead Guest Editor: Panagiotis Vlamos
Guest Editors: Ilias Kotsireas and Dimitrios Vlachakis

Editorial Board

Contents

Hindawi
Complexity
Volume 2019, Article ID 8658124, 2 pages
https://doi.org/10.1155/2019/8658124

Editorial

Complexity in Medical Informatics

Panagiotis Vlamos,[1] **Ilias Kotsireas**,[2] **and Dimitrios Vlachakis**[3]

[1]*Bioinformatics and Human Electrophysiology Laboratory, Department of Informatics, Ionian University, Greece*
[2]*Department of Physics and Computer Science, Wilfrid Laurier University, Ontario, Canada*
[3]*Department of Biotechnology, Agricultural University of Athens, Greece*

Correspondence should be addressed to Panagiotis Vlamos; vlamos@ionio.gr

Received 4 December 2018; Accepted 26 May 2019; Published 1 July 2019

In the era of big data and personalized medicine, the need to develop alternative methods for analyzing large volumes of data generated by clinical IoT platforms has formed. Big data analysis utilizes several data mining algorithms to determine patterns and connections in large databases [1].

Research studies use numerous data mining methodologies (i.e., machine learning algorithms). Genetic algorithm (GA) is a random optimization method derived from Darwin's principle of survival of the fittest in natural genetics [2]. GA is a metaheuristic method used in data mining applications such as feature selection and classification. According to Goldberg, 2012, the main idea behind GAs was robustness. Since their development, GAs proved to be a robust search tool. The fact that their use is not restricted by the search space is thought to be one of their biggest advantages [3]. GAs are of interest since they are not only very potent tools but also widely applicable ones. Among other medical areas, GAs are particularly useful for neurology, pharmacotherapy, and healthcare management [4]. GAs can also be used to train artificial neural networks (ANNs). A study showed that GA, in combination with the Levenberg-Marquardt backpropagation (LM) algorithm, was the best algorithm for ANN training, with 96.5% general success [5].

Another prevailing data processing technique is neural networks (NNs), and as their name suggests, they compare to the way the brain processes information. They are (usually) a nonlinear statistical tool, used to model complex relationships between data inputs and outputs or to establish data patterns. Current uses of neural networks include image analysis, signal processing, and laboratory medicine. Regarding the field of laboratory medicine, researchers aim to improve scientific procedures by minimizing errors and decrease the workload of the laboratory staff, while maintaining/improving the safety of patients.

Figure 1 aptly describes the workflow of image processing. NNs play a significant role in each step. For step 1 (preprocessing), NNs can be utilized to solve the following problems: (1) optimization, (2) approximation, and (3) mapping. For data reduction (step 2), NNs aid in obtaining an image compression rate as high as possible. NNs are also trained (i) to implement feature extraction and (ii) pixel-based segmentation (step 3) and (iii) to locate objects based on pixel data (step 4). For step 5, NNs' possible uses include object classification (e.g., chromosomes) and camera image analysis. For optimization (step 6), NNs were employed for tasks regarding segmentation and recognition [6].

The complexity and difficulties in analyzing big data pushed science to search for new and highly efficient solutions to the said issue. These aforementioned approaches are utilized for the extraction of useful information for data analysis and decision-making purposes. The aim of this special issue is to examine the applicability of novel techniques and especially the accuracy of the neural network methodologies and genetic algorithms in multivariate analysis of clinical data.

Technological interfaces related to networking issues, such as data mining, machine and deep learning, and intelligent decision support systems, can be exploited to address challenges regarding the improvement of patient's safety, the enhancement of care outcomes, the promotion of a patient-centered care, the facilitation of translational research, the

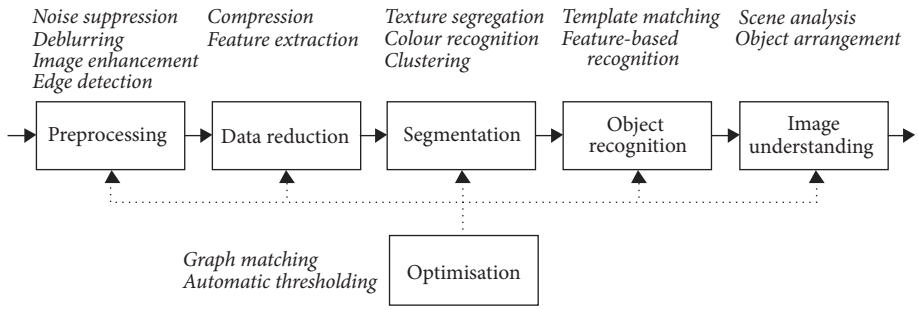


Figure 1: Image processing workflow (figure taken from [6]).

activation of precision medicine, and the improvement of education and skills in health informatics.

The topics of the accepted articles include but are not limited to the following: machine and deep learning approaches for health data; data mining and knowledge discovery in healthcare; clinical decision support systems; applications of the genetic algorithm in disease screening, diagnosis, and treatment planning; neurofuzzy system based on genetic algorithm for medical diagnosis and therapy support systems; applications of AI in healthcare; applications of artificial neural networks in medical science; electronic medical record and missing data; network and disease modeling (using administrative data); and health analytics and visualization.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this special issue.

Panagiotis Vlamos
Ilias Kotsireas
Dimitrios Vlachakis

References

[1] C. H. Lee and H. Yoon, "Medical big data: promise and challenges," *Kidney Research and Clinical Practice*, vol. 36, no. 1, pp. 3–11, 2017.

[2] T. Jun-shan, H. Wei, and Q. Yan, "Application of genetic algorithm in data mining," in *Proceedings of the 2009 First International Workshop on Education Technology and Computer Science*, pp. 353–356, Wuhan, Hubei, China, March 2009.

[3] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Boston, Mass, USA, 2012.

[4] A. Ghaheri, S. Shoar, M. Naderan, and S. S. Hoseini, "The applications of genetic algorithms in medicine," *Oman Medical Journal*, vol. 30, no. 6, pp. 406–416, 2015.

[5] S. Koçer and M. R. Canal, "Classifying epilepsy diseases using artificial neural networks and genetic algorithm," *Journal of Medical Systems*, vol. 35, no. 4, pp. 489–498, 2011.

[6] M. Egmont-Petersen, D. de Ridder, and H. Handels, "Image processing with neural networks—a review," *Pattern Recognition*, vol. 35, no. 10, pp. 2279–2301, 2002.

Hindawi
Complexity
Volume 2018, Article ID 4216813, 12 pages
https://doi.org/10.1155/2018/4216813

Research Article

Predicting Protein Interactions Using a Deep Learning Method-Stacked Sparse Autoencoder Combined with a Probabilistic Classification Vector Machine

Yanbin Wang,[1,2] **Zhuhong You**,[1] **Liping Li**,[1] **Li Cheng**,[1] **Xi Zhou**,[1] **Libo Zhang**,[3] **Xiao Li**,[1] **and Tonghai Jiang**[1]

[1]*Xinjiang Technical Institutes of Physics and Chemistry, Chinese Academy of Science, Urumqi 830011, China*
[2]*University of Chinese Academy of Sciences, Beijing 100049, China*
[3]*Institute of Software, Chinese Academy of Sciences, Beijing 100190, China*

Correspondence should be addressed to Zhuhong You; zhuhongyou@hotmail.com and Liping Li; lipingli@ms.xjb.ac.cn

Received 1 February 2018; Revised 10 May 2018; Accepted 13 June 2018; Published 10 December 2018

Academic Editor: Panayiotis Vlamos

Protein-protein interactions (PPIs), as an important molecular process within cells, are of pivotal importance in the biochemical function of cells. Although high-throughput experimental techniques have matured, enabling researchers to detect large amounts of PPIs, it has unavoidable disadvantages, such as having a high cost and being time consuming. Recent studies have demonstrated that PPIs can be efficiently detected by computational methods. Therefore, in this study, we propose a novel computational method to predict PPIs using only protein sequence information. This method was developed based on a deep learning algorithm-stacked sparse autoencoder (SSAE) combined with a Legendre moment (LM) feature extraction technique. Finally, a probabilistic classification vector machine (PCVM) classifier is used to implement PPI prediction. The proposed method was performed on human, unbalanced-human, *H. pylori*, and *S. cerevisiae* datasets with 5-fold cross-validation and yielded very high predictive accuracies of 98.58%, 97.71%, 93.76%, and 96.55%, respectively. To further evaluate the performance of our method, we compare it with the support vector machine- (SVM-) based method. The experimental results indicate that the PCVM-based method is obviously preferable to the SVM-based method. Our results have proven that the proposed method is practical, effective, and robust.

1. Introduction

Most important molecular processes in cells are performed by different types of protein interactions. Thus, one of the main objectives of functional proteomics is to determine the protein-protein interactions of organisms. With continuous research and the development of technique, it is now possible to detect protein interactions on a large scale by using high-throughput experimental techniques. Such research is obviously very important, because the research of PPIs is closely related to many functions of complex life systems, and these functions are not determined by the characteristics of the individual components. For example, molecular cell signaling is carried out through protein interactions. This process is not only the basis of many life functions, but it is also related to many diseases. In addition, the study of protein interactions has been of great value in the development of new drugs and in the prevention and diagnosis of disease.

As some high-throughput experimental techniques have been successfully applied to postgenomic era PPI research tasks, a large number of different species of PPI data have been collected, and some databases have been created to systematically collect and store experimentally determined PPIs [1–3]. Even though experimentally validated PPI data drives research and development of proteomics, they often have high false positives and false negatives [4–7]. In addition, because the experimental method has some unavoidable defects, such as having a high cost and being time consuming, the researchers have only verified a small part of the whole

PPI network even after a long period of effort. With advances in mathematical and computational methods [8–12], computer technology has been applied in more and more fields. Vlachakis et al. proposed computational methods to simulate catalytic mechanisms, complete drug design, and model protein three-dimensional structures [13–17]. Vlamos et al. developed several intelligent disease diagnosis applications and hybrid models for vulnerability detection [18–25]. Some researchers also introduced computational methods into the medical field and developed several automated diagnostic models [26, 27]. Therefore, using a machine learning algorithm to develop an efficient and accurate automatic discriminative system to predict new protein interactions has important practical significance.

To date, a variety of protein information has been used to build PPI prediction models based on machine learning algorithms. Protein information that can be used includes, but is not limited to, physicochemical information, structural information, evolutionary information, and protein domains. However, these methods have some limitations when they are used. For example, some computational methods using genomic information predict protein interactions by calculating a set of gene presence or absence patterns. The main factor limiting these methods is that they can only be applied to fully sequenced genomic data [28, 29]. Recently, methods that directly extract information from a protein primary sequence have attracted much attention. Methods that use only protein sequence information are more general than methods that rely on some additional information about proteins. Many researchers are working on the development of sequence-based computational models to predict new PPIs. Hamp and Rost developed a computational method for predicting PPIs based on profile-kernel support vector machines combined with evolutionary profiles [30]. An et al. proposed a PPI prediction method that combines the local phase quantization and relevance vector machine [31]. Yang et al. used a new local descriptor to describe the interaction between the contiguous and discontinuous regions of the protein sequence, which is able to obtain more protein interaction information from sequences [32]. Zhang et al. introduced two ensemble methods to predict PPIs. These ensemble methods are based on undersampling techniques and fusion classifiers [33]. You et al. proposed a prediction framework for detecting PPIs using a low-rank approximation-kernel extreme learning machine [34]. Several other sequence-based computational methods have been reported in previous work [35–38]. These sequence-based methods show that the individual information of the amino acid sequence is sufficient to determine the interaction of the protein. However, these methods usually use physical, chemical, or structural information, and even the fusion of all of these types of information as features of the protein sequence. Therefore, the feature extraction steps of these methods are not efficient. In addition, the above information can only represent each specific protein sequence but does not contain knowledge related to protein interactions. Therefore, even these methods combined with advanced classification algorithms have a difficulty in producing enough accuracy.

Compared with the physicochemical information, the evolutionary information of proteins can reflect the potential interactions between proteins. Therefore, we consider the evolutionary information of the protein as a feature of the protein sequence. Extracting the evolutionary information of a protein is challenging as there is currently no strategy that can efficiently obtain the evolutionary information of a protein. We hypothesize that there is a potential relationship between the conservation of amino acid residues during evolution and the interaction of proteins. Based on this hypothesis, we propose an efficient protein evolution feature extraction scheme, which used a deep learning algorithm combined with Legendre moments (LMs) and position weight matrix (PWM). Specifically, we first convert the protein sequence into a PWM containing the amino acid residue conservative score. Then, we use LMs to extract important evolutionary information from the PWM and generate the feature vector $\overrightarrow{\mathbf{F}}$. Last but not least, this feature $\overrightarrow{\mathbf{F}}$ was further optimized by using SSAE deep neural networks to eliminate noise, obtain primary information, and reduce feature dimensions. In addition, in response to the challenges posed by big data and imbalanced datasets, a sparse model, PCVM, was used to perform classification. Our contributions can be summarized as follows:

(1) We propose a method to predict PPIs quickly, efficiently, and accurately.

(2) We have abandoned the traditional materialized information and structural information, considered the evolutionary information associated with PPIs as a feature of the protein sequence, and proposed a feature extraction strategy to quickly and efficiently extract the evolutionary information of the protein and improve the prediction performance.

(3) We confirm that sparse classification algorithms can greatly benefit prediction of PPIs and present results showing that they can provide a benefit in dealing with large-scale data and unbalanced data (as is the case with PCVM).

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 describes the datasets and methods used in this paper. Section 4 shows the results of the experiment. Section 5 concludes the paper.

2. Related Work

The study of the PPI prediction model is mainly divided into two parts. One is the development of protein sequence feature extraction strategies, and the other is the application of classification algorithms. This section briefly reviews related research.

2.1. Sequence-Based Feature Extraction Algorithm. Previous methods of extracting sequence features were mainly the direct use of physicochemical information or amino acid sequence structure information or evolutionary information of proteins. Since the amino acid composition model has

been proposed, many subsequent works have been carried out for the composition model. Chou [39] proposed a feature extraction method called pseudoamino acid composition. This feature extraction method greatly increases the information content of the amino acid sequences contained in the features. It does not only consider the composition of amino acids, but also processes the amino acid position information. Another excellent research was done by Shen et al. [40]. In that study, 20 amino acids were clustered into 7 classes based on their dipole and side chain volumes, and then the features of the protein pairs were extracted based on the amino acid class. Combined with the SVM classifier, this method has a prediction accuracy of 83.9% on human PPIs. In a study by Guo et al. [41], an autocovariance-based method was developed to extract the interaction information of discontinuous amino acid fragments in a sequence. The method replaces the protein sequence with a digital sequence based on the physicochemical properties, and the replaced digital sequence is regarded as a group of information for analysis.

Different from the previous classical computational method, we did not use the traditional sequence-coding scheme and did not consider the physicochemical information of the protein sequence. Our method uses the evolutionary information of the protein sequence indirectly (using Legendre moments to extract feature vectors on the PSSM matrix containing evolutionary information), trying to use image-processing ideas to complete the task of PPI prediction; this is a direction in which only a few people are exploring. The introduction of our method and the satisfactory results produced on several gold standard datasets have greatly encouraged the scholars who explored on this direction. The advantage of this method is that the feature extraction strategy is simple and efficient, does not require complicated sequence coding, and does not need to consider the physicochemical information of the protein. Compared with traditional feature extraction methods, this method greatly improves the accuracy of PPI prediction and saves time and computational overhead.

In addition, the deep learning algorithm has shown extraordinary performance in many fields, but its ability has not been effectively verified in the PPI prediction task. A deep learning algorithm-stacked sparse autoencoder was used to reconstruct a protein feature vector in our work. This algorithm uses sparse network structures and adds sparseness restrictions on neurons. This not only allows us to obtain low-dimensional, low-noise protein feature vectors, but also improves the efficiency of the network. The results of our method applied to the test set demonstrate once again that deep learning algorithms can be used to assist in solving bioinformatics problems.

2.2. Classifier. The support vector machine (SVM) is one of the most commonly used classification algorithms in the PPI prediction model [42–44]. However, the SVM approach has some obvious drawbacks: (1) As the dataset becomes larger, the support vector increases rapidly. (2) Cross-validation-based kernel parameter optimization strategy consumes a large amount of computing resources. Another widely used classifier is the relevance vector machine (RVM) [45–47], which effectively avoids the disadvantages of SVM. It was developed to take advantage of the Bayesian inference and the prior weight following a zero-mean Gaussian distribution. However, the RVM has the potential to produce some unreliable vectors that lead to system error decisions. Because the weights of the negative class and the positive class are given by the zero-mean Gaussian prior, partial training samples that do not interact might be assigned confident weights or vice versa.

In order to avoid the problems of the above classifiers, we used the probability classification vector machine (PCVM) method to perform PPI classification, which provides different priors for different types of samples. The positive class is associated with a right-truncated Gaussian and the negative class is associated with a left-truncated Gaussian. The PCVM method has the following advantages: (1) PCVM produces sparse predictive models and has better efficiency in the testing phase. (2) PCVM provides probabilistic results for each output. (3) PCVM uses the EM algorithm to automatically find the optimal initial point, which saves time and improves the performance of the system.

3. Materials and Methodology

3.1. Datasets. To evaluate the performance of the proposed method, there are a total of 4 different PPI datasets used in our experiments, two of which are human, one is *S. cerevisiae*, and one is *H. pylori*.

The first human PPI dataset we used was from Pan et al. [48], which was downloaded from the Human Protein Reference Database (HPRD). After the self-interaction and repetitive interactions were removed, the remaining 36,630 PPI pairs formed the final gold standard positive (GSP) dataset. For the selection of gold standard negative (GSN) datasets, we followed the previous work [48] and generated GSN datasets from the Swiss-Prot version 57.3 database according to the following criteria: (1) Protein sequences annotated by uncertain terms are removed. (2) Multiple unlocalized protein sequences are deleted. (3) Protein sequences that may be only "fragments" or containing "fragments" are deleted.

After strictly following the above steps, 1773 human proteins were screened out. Noninteracting protein pairs are then constructed by randomly pairing proteins from different subcellular compartments. In addition, another golden negative dataset was downloaded, which was used in the study by Smialowski et al. [49]. The final GSN dataset was constructed by combining the above two negative datasets, which consisted of 36,480 noninteracting protein pairs. Therefore, the entire gold standard dataset (GSD) consists of 73,110 protein pairs, of which almost half is from the positive dataset and half is from the negative dataset.

Due to the fact that there are serious imbalances in the dataset in real-world tasks, this can lead to a failure of the PPI prediction model. Considering this issue, we have constructed another set of human datasets with an unbalanced number of positive and negative samples to evaluate the stability and robustness of our proposed method. This unbalanced human PPI dataset consists of 3899 positive samples and 13,000 negative samples.

The third PPI gold standard dataset we used was from downloaded datasets from the *S. cerevisiae* core subset of the database of interacting proteins (DIP). We strictly followed the work of Guo et al. [41] to construct the *S. cerevisiae* dataset. Finally, we obtained a gold standard dataset containing 11,188 protein pairs, of which 5594 positive protein pairs form a GSP dataset and 5594 negative protein pairs form a GSN dataset.

The last PPI dataset uses the pair of *H. pylori* proteins described by Martin et al. [50], which includes 1458 positive sample pairs and 1458 negative sample pairs.

3.2. Position Weight Matrix. In this article, we use the position weight matrix (PWM) to derive evolutionary information from protein sequences. A PWM for a query protein is a $Y \times 20$ matrix $\mathbf{M} = \{m_{ij},\ i = 1, \dots, Y, j = 1, \dots, 20\}$, where Y represents the size of the protein sequence and the number of columns of the $\mathbf{M}$ matrix denotes 20 amino acids. In order to construct PWM, a position frequency matrix is first created by calculating the presence of each nucleotide on each position. This frequency matrix can be represented as $\mathbf{p}(a, c)$, where u means position and k is the kth nucleotide. The PWM can be expressed as $M_{ij} = \sum_{k=1}^{20} \mathbf{p}(a, c) \times \mathbf{w}(b, c)$, where $\mathbf{w}(b, c)$ is a matrix whose elements represent the mutation value between two different amino acids. Consequently, high scores represent highly conservative positions, and low points represent a weak conservative position. It's an extremely useful tool for predicting protein disulfide connectivity, protein structural classes, subnuclear localization, and DNA or RNA binding sites. Here, we also employ PWMs to detect PPIs. In this paper, each protein is interpreted as PWMs using the position-specific iterated BLAST (PSI-BLAST). The PSI-BLAST has two important parameters, e value and iteration number, which were set at 0.001 and 3, respectively [51–53].

3.3. Legendre Moments. Legendre moments (LMs) are typical orthogonal moments, whose kernel function is the Legendre polynomial. It has been widely involved in a lot of applications, such as image analysis, computer vision, and remote sensing [54–58]. Here, we use the Legendre moment to extract the evolutionary information of the protein indirectly from the PWM and generate a 961-dimensional eigenvector. The two-dimensional discrete form of the LM is represented as

$$L_{mn} = \mu_{mn} \sum_{i=1}^{K} \sum_{j=1}^{L} h_{mn}(x, y) g(x_i, y_i), \tag{1}$$

where $g(x, y)$ is defined as a set of discrete points (x_i, y_i), $-1 \le x_i,\ y_i \le +1$. K represents the number of columns of the PWM matrix, L represents the sum of each column of PWM matrix.

$$h_{mn}(x, y) = \int_{x_i - \Delta x/2}^{x_i + \Delta x/2} \int_{y_i - \Delta y/2}^{y_i + \Delta y/2} R_m(x) R_n(y) dx dy. \tag{2}$$

The integral terms in (2) are frequently estimated by zeroth-order approximation; in other words, the values of Legendre polynomials are assumed to be constant over the intervals $[x_i - \Delta x/2, x_i + \Delta x/2]$ and $[y_i - \Delta y/2, y_i + \Delta y/2]$. In this case, the set of approximated LMs is defined as:

$$L'_{mn} = \frac{(2m + 1)(2n + 1)}{KL} \sum_{i=1}^{K} \sum_{j=1}^{L} R_m(x_i) R_n(y_i) g(x_i, y_i). \tag{3}$$

3.4. Stacked Sparse Autoencoder. Deep learning is a new field in machine learning research. Its motivation lies in building and simulating the neural network of the human brain for analytical learning. It imitates the mechanism of the human brain to interpret data. In this paper, the deep structure stacked sparse autoencoder (SSAE) is adopted for feature reduction and reconstruction [59–62]. SSAE forms a more abstract high-level representation feature by combining low-level features to discover the distributed feature representation of protein feature data.

The SSAE is an unsupervised network that is a large-scale nonlinear system composed by multilayer neuron cells in which the outputs of the current layer neuron are fed to the connectivity layer neuron. In this work, the aim of SSAE is to learn a distinctive representation for the Legendre moment (LM) feature. The underlying purposes are noise elimination and dimensionality reduction. The process of feature reconstruction is layer by layer in SSAE. The first layer is in charge of rough integration original input. The second layer is responsible for extracting and integrating the features learned earlier. Higher successive layers will be inclined to produce low-dimensional, low-noise, and high-cohesion features. In this paper, the SSAE was used to reduce the LM feature to 200 dimensional.

SSAE or Sparse autoencoder network is mainly made up of two parts, the encoding part and the decoding part [63], where the encode network compresses high-dimensional into low-dimensional attributes. The decoding network is responsible for restoring the original input layer by layer, and the network structure is symmetrical with the structure of the encoding network. In the coding stage, the primary data x is mapped onto a hidden layer. This process can be represented as

$$z = \sigma_1(w_1 x + b_1). \tag{4}$$

Here, σ_1 is a nonlinear function, w_1 is the weight of the encoding part and b_1 is the bias. After that, the original data is reconstructed by the decoding network:

$$x' = \sigma_2(w_2 z + b_2), \tag{5}$$

where w_2 is the weight of decoding network and b_2 is the bias. The purpose of SAE is to make the output as close as possible to the input by minimizing loss function:

$$\theta = \arg\min \left[\frac{1}{n} \sum_{i=1}^{n} L\left(x_i, x'_i\right) + \beta \sum_{j=1}^{S_2} KL\left(\rho \| \widehat{\rho}_j\right) \right], \tag{6}$$

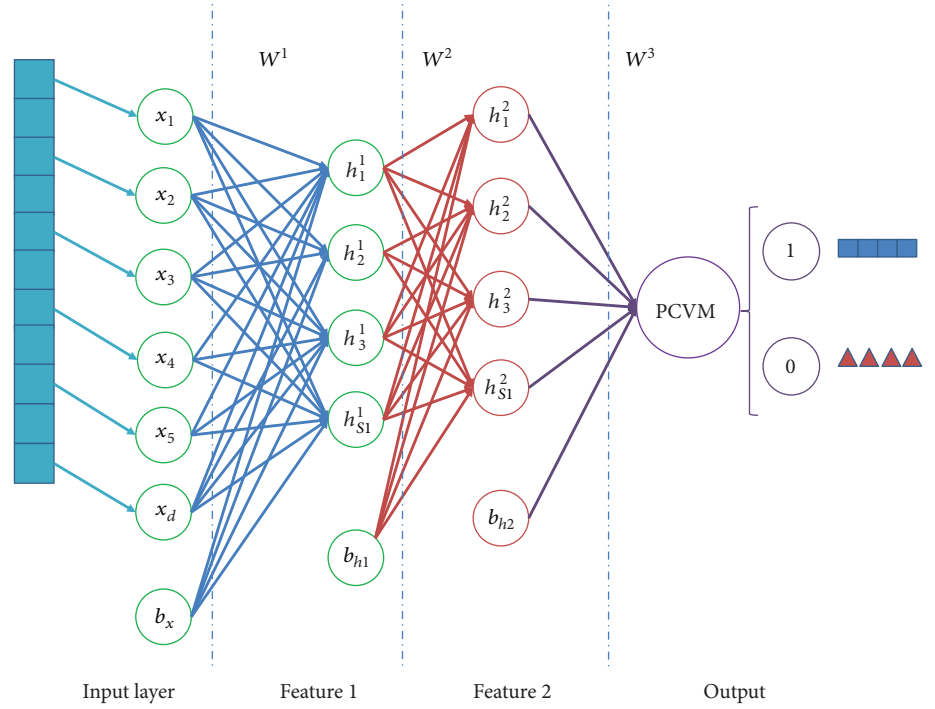


Figure 1: Stacked sparse autoencoder with two hidden layer structures.

$$\theta = \frac{1}{2N}\sum_{i=1}^{N}\left\|x_i' - x_i\right\|^2 + \beta\sum_{j=1}^{S_2} KL\left(\rho\|\widehat{\rho}_j\right), \tag{7}$$

where N is the number of hidden layer nodes, β is the weight of the sparse penalty item, $\widehat{\rho}_j$ represents the average activation value of the hidden layer element, and ρ is the sparse parameter.

Figure 1 shows a SSAE network with two hidden layers, of which the decoding part has not been shown, in order to highlight the feature reduction function of the network. Similar to the sparse autoencoder (SAE), the key to the training model is to learn the parameters $\theta = (W, b)$, which allows the model to have minimum input and output deviation. Once the optimal parameters θ are obtained, the SSAE yield function $R^{d_x} \to R^{d_{h(2)}}$ that transforms original data to a low-dimensional space.

3.5. Probabilistic Classification Vector Machines. The design of feature extraction strategies and the selection of classifiers are two crucial parts in developing an excellent PPI prediction model. In the previous description, we developed a new deep learning-based amino acid sequence feature extraction method. Here, we use the stronger PCVM classifier to replace the Softmax layer of the stacked sparse autoencoder to achieve the output of our model. Like most classification models, the goal of a PCVM [64–66] is to generate a model $f(X; W)$ by learning a set of labeled data $\{X, Y\}$. The model is determined by parameters W learned and expressed as

$$f(X; W) = \sum_{i=1}^{N} w_i \varnothing_{i,\theta}(x) + b, \tag{8}$$

where the $W = (w_1, \ldots, w_N)^T$ denotes the parameter of the model, $\varnothing_{i,\theta}(x)$ is a set of primary functions, and b represents the bias. A Gaussian cumulative distribution function $\beth(x)$ is used for obtaining the binary outputs. The function is defined as

$$\beth(d) = \int_{-\infty}^{d} N(r \mid 0, 1). \tag{9}$$

After incorporating (7) with (8), the model becomes

$$K(X; W, b) = \beth\left(\sum_{i=1}^{N} w_i \varnothing_{i,\theta}(x) + b\right) = \beth(\Phi_\theta(X)W + b). \tag{10}$$

Each weight w_i is assigned a prior by a truncated Gaussian distribution, as follows:

$$p(W \mid \boldsymbol{\alpha}) = \prod_{i=1}^{N} p(w_i \mid \alpha_i) = \prod_{i=1}^{N} N_t\left(w_i \mid 0, \alpha_i^{-1}\right), \tag{11}$$

where the bias b is assigned a zero-mean Gaussian prior, as follows:

$$p(b \mid \beta) = N\left(b \mid 0, \beta^{-1}\right), \tag{12}$$

where the $N_t(w_i \mid 0, \alpha_i^{-1})$ is a truncated Gaussian, and α_i denotes the inverse of the variance. The EM algorithm is used for obtaining all parameters of a PCVM model [67].

4. Results

4.1. Evaluation Criteria. In this work, the following criteria, such as accuracy (Accu), precision (Prec), sensitivity (Sens), and Matthews's correlation coefficient (Mcc), are used to assess the proposed method. Accuracy is used to describe the overall system error. Since the key task of PPI prediction is to correctly predict the interacting protein pairs, the sensitivity and accuracy indicators are used to assess the model's ability to predict positive data. In addition, data imbalance exists in real PPI prediction tasks. In view of this situation, we used an unbalanced PPI dataset in this paper. Therefore, Mcc is used to evaluate the reliability and stability of the model when dealing with unbalanced data. When the model appears "preference prediction" (i.e., the dataset is very unbalanced, the model can only correctly predict negative data), the Mcc score is lower. When the model is strong and robust, the indicator score is high. These indicators are defined as

$$\text{Accu} = \frac{\text{TN} + \text{TP}}{\text{FP} + \text{TP} + \text{FN} + \text{TN}},$$

$$\text{Sens} = \frac{\text{TP}}{\text{TN} + \text{TP}},$$

$$\text{Prec} = \frac{\text{TP}}{\text{TP} + \text{FP}},$$

$$\text{Mcc} = \frac{(\text{TN} \times \text{TP}) - (\text{FN} \times \text{FP})}{\sqrt{(FN + TP) \times (FP + TN) \times (FP + TP) \times (FN + TN)}}, \quad (13)$$

where TP means those samples, true interacting with each other, are predicted correctly. FP represents those samples, true noninteracting with each other, are judged to be interacting. TN represents those samples, true noninteracting with each other, are predicted correctly. FN represents those samples, true interacting with each other, are judged to be noninteracting. Furthermore, the ROC (receiver operating characteristic) is portrayed to appraise the performance of a set of classification results [68] and the AUC (area under ROC) is computed as an important evaluation indicator.

4.2. Assessment of Prediction. In this paper, the proposed sequence-based PPI predictor is implemented using a MATLAB platform. All the simulations are carried out on a computer with a 3.1 GHz 8-core CPU, 16 GB memory, and a Windows operating system. In order to make the prediction system independent of the training data, each PPI dataset is segmented into five parts by the five-fold cross-validation method. The performance of the PCVM-based method on human, unbalanced-human, *H. pylori*, and *S. cerevisiae* datasets are exposed in Tables 1–4. The corresponding ROC curves are depicted in Figures 2–6, respectively.

Analyzing Table 1 allows drawing the conclusion that the PCVM-based method yields a satisfactory result on the human dataset, where the accuracy of each fold is above 98% and the accuracy standard deviations of five experiments are only 0.2%. The corresponding average sensitivity, precision, and Mcc are 98.47%, 98.67%, and 97.19%, respectively. Their standard deviations are 0.3%, 0.5%, and 0.3%, respectively. The average AUC (Figure 2) of the five experiments reached 0.9984. The high accuracies and AUC show that the PCVM-based approach has a strong classification ability in identifying PPIs. The low standard deviations illustrate that this model is robust and stable.

When predicting PPIs on the unbalanced-human dataset (Table 2), the method produced an average accuracy of 97.71%, sensitivity of 91.87%, precision of 98.10%, and AUC of 0.9971, respectively.

When applied on the *H. pylori* dataset with the smallest training set, the PCVM-based methods also yielded a high average prediction accuracy of 93.76%, high precision of 92.10%, high sensitivity of 95.77%, and high Mcc of 88.26%, respectively (Table 3). The standard deviations of Accu, Sens, Prec, and Mcc in the five experiments are 0.1%, 2.0%, 1.9%, and 0.2%, respectively. Moreover, the average AUC on the *H. pylori* dataset reached 0.9860.

Table 1: 5-Fold cross-validation results using the proposed method on the human dataset.

Testing set	Accu (%)	Sens (%)	Prec (%)	Mcc (%)
1	98.50	98.87	98.13	97.04
2	98.69	98.53	98.89	97.41
3	98.31	98.35	98.22	96.68
4	98.69	98.51	98.88	97.41
5	98.69	98.11	99.23	97.41
Average	98.58 ± 0.2	98.47 ± 0.3	98.67 ± 0.5	97.19 ± 0.3

Table 2: 5-Fold cross-validation results using the proposed method on the unbalanced-human dataset.

Testing set	Accu (%)	Sens (%)	Prec (%)	Mcc (%)
1	97.57	91.71	97.67	93.23
2	97.78	92.44	98.00	93.86
3	97.72	92.20	97.12	93.32
4	97.75	91.26	99.19	93.78
5	97.75	91.76	98.50	93.74
Average	97.71 ± 0.1	91.87 ± 0.5	98.10 ± 0.8	93.59 ± 0.3

Table 3: 5-Fold cross-validation results using the proposed method on the *H. pylori* dataset.

Testing set	Accu (%)	Sens (%)	Prec (%)	Mcc (%)
1	94.00	96.76	92.28	88.62
2	93.65	95.73	91.50	88.10
3	93.65	92.52	94.77	88.11
4	93.83	95.67	92.58	88.38
5	93.66	98.18	89.37	88.10
Average	93.76 ± 0.1	95.77 ± 2.0	92.10 ± 1.9	88.26 ± 0.2

Table 4: The prediction performance comparison of PCVM with SVM.

Model	Testing set	Accu (%)	Sens (%)	Prec (%)	MCC (%)
PCVM	1	96.83	97.37	96.44	93.85
	2	96.33	97.33	95.22	92.93
	3	96.33	96.86	96.02	92.93
	4	96.60	96.85	96.33	93.44
	5	96.64	97.75	95.19	93.11
	Average	96.55 ± 0.2	97.23 ± 0.3	95.84 ± 0.5	93.25 ± 0.3
SVM	1	94.46	93.68	95.36	89.53
	2	93.70	90.32	96.46	88.13
	3	93.92	92.49	95.49	88.58
	4	92.76	91.99	93.33	86.56
	5	93.53	92.99	93.92	87.89
	Average	93.67 ± 0.6	92.29 ± 1.2	94.91 ± 1.2	88.13 ± 1.0

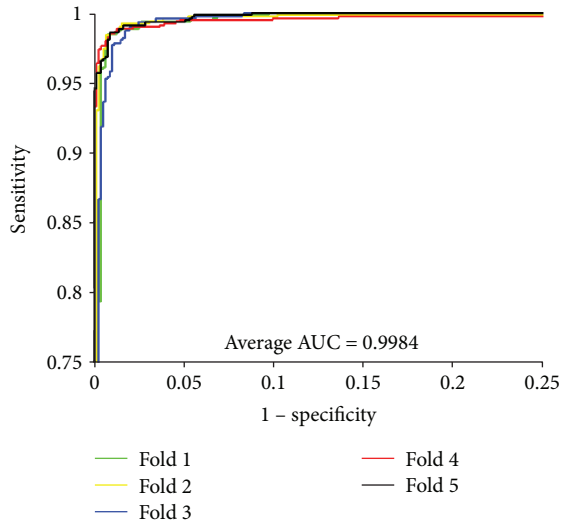


Figure 2: ROC curves performed by the proposed approach on the human dataset.

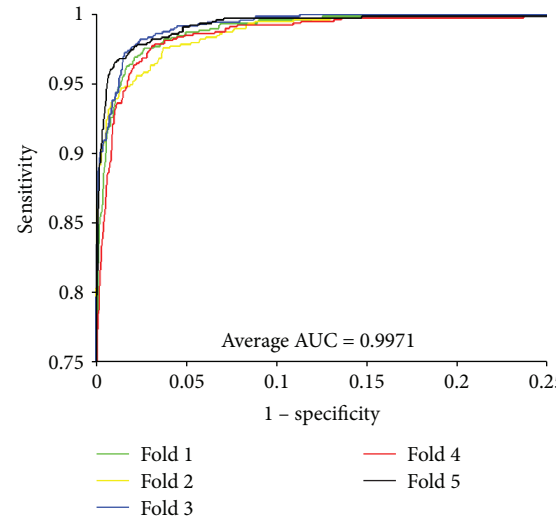


Figure 3: ROC curves performed by the proposed approach on the unbalanced-human dataset.

4.3. Comparison with the SVM-Based Approach. In order to highlight the feasibility of our classifier, the state-of-the-art SVM classifier was used to compare with PCVM. To make it fair, the same feature extraction scheme and the same *S. cerevisiae* dataset were used in this experience. The LIBSVM tool [69] is available for SVM classification, and the grid search approach was adopted for optimizing SVM model parameters c and g.

The classification results of the PCVM and SVM classifiers on the *S. cerevisiae* dataset are listed in Table 4, and the ROC curves of SVM are displayed in Figures 5 and 6. As we have seen, the average result of the PCVM method achieved 96.55% Accu, 97.23% Sens, 95.84% Prec, and 93.25% Mcc. The standard deviation of these indicators in five experiments are 0.2%, 0.3%, 0.5%, and 0.3%, respectively. The average results of the SVM method yielded 93.67% Accu, 92.29% Sens, 94.91% Prec, and 88.13% Mcc. The standard deviations are 0.6%, 1.2%, 1.2%, and 1.0%, respectively. In comparison with SVM, the PCVM classifier achieves significantly better results on this gold standard dataset. From Figures 5 and 6, the average AUC of the SVM classifier is 0.9856, which is significantly lower than those of PCVM of 0.9963. Higher AUC values clearly illustrate that the PCVM method is more accurate and more reliable for detection PPIs. The improved classification performance of the PCVM classifier compared with the SVM classifier can be explained by two reasons: (1) The number of PCVM basis functions is less than the number of training points, resulting in a reduction in the computational effort involved. (2) PCVM uses truncated Gauss priors to flexibly assign a priori information about weights, thus ensuring the generation of reliable support vectors.

4.4. Compare with Previous Studies. Some other computational approaches for predicting PPI have been reported in previous studies. These highlight the advantages of the

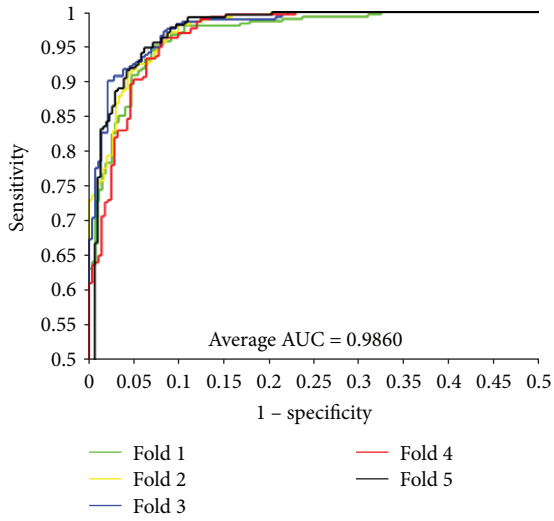


Figure 4: ROC curves performed by the proposed approach on the *H. pylori* dataset.

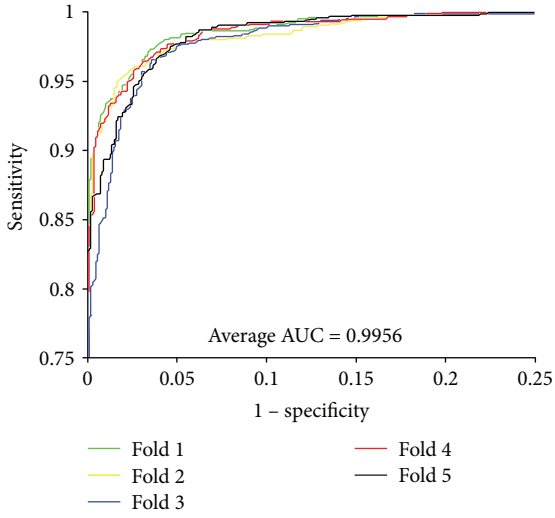


Figure 5: ROC curves performed by the proposed approach on the *S. cerevisiae* dataset.

proposed approach, which was compared with the existing approaches that attract wide attention on the same PPI datasets, respectively. We can see from Table 5 that our method also produces better results than other existing methods. The performance of the several different approaches on the *H. pylori* dataset is presented in Table 6. As seen from the Table 6, our proposed approach produces better performances than the four other main methods. The 93.76% prediction accuracy is much higher than any of the several other methods. Table 7 shows the results of comparing with several other different methods that achieved an average prediction accuracy of less than 93.92% on the *S. cerevisiae* dataset, while our PCVM-based approach obtained an average prediction accuracy of 96.55% with the lowest standard deviation of 0.2%. Meanwhile, the sensitivity of 97.23% is also far better than those of the other methods.

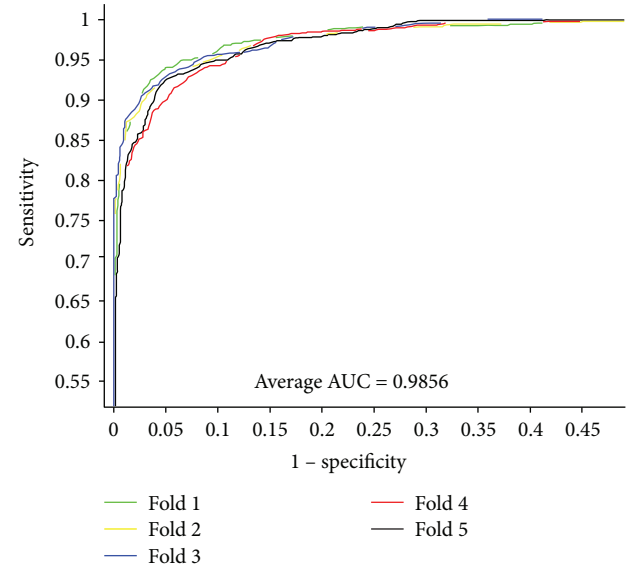


Figure 6: ROC curves performed by the SVM-based approach on the *S. cerevisiae* dataset.

Table 5: Performance comparison of different methods on the human dataset.

Model	Accu (%)	Sens (%)	Prec (%)	MCC (%)
LDA + RF [70]	96.40	94.20	N/A	92.80
LDA + RoF [70]	95.70	97.60	N/A	91.80
LDA + SVM [70]	90.70	89.70	N/A	81.30
AC + RF [70]	95.50	94.00	N/A	91.40
AC + RoF [70]	95.10	93.30	N/A	91.10
AC + SVM [70]	89.30	94.00	N/A	79.20
Proposed method	97.71	91.87	98.10	93.59

Extensive experiments indicate that the method we employ can sufficiently meet the needs of large-scale protein detection and can be used as a meaningful adjunct application for proteomics investigation.

5. Conclusion

The function and activity of proteins are usually regulated by other proteins that interact with it. In order to understand biological processes, we need to develop a tool that gives us an insight into the knowledge of protein interactions. Although many efforts have been taken to develop the method for detecting PPIs, the accuracy and robustness of most existing methods still have potential room to be improved. Hence, we explore a fresh and efficient computational system based on protein sequences using a PCVM classifier combined with Legendre moments and a stacked sparse autoencoder. Four strictly screened PPI datasets are used to assess the prediction ability of our devised approach and the prediction outcomes display that the approach provides practical predictive capability for PPI detection. In

Table 6: Performance comparison of different methods on the *H. pylori* dataset.

Model	Accu (%)	Sens (%)	Prec (%)	MCC (%)
Phylogenetic bootstrap [71]	75.80	69.80	80.20	N/A
Boosting [71]	79.52	80.30	81.69	70.64
Signature products [72]	83.40	79.90	85.70	N/A
HKNN [73]	84.00	86.00	84.00	N/A
Proposed method	93.76	95.77	92.10	88.26

Table 7: Performance comparison of different methods on the *S. cerevisiae* dataset.

Model	Testing set	Accu (%)	Sens (%)	Prec (%)	MCC (%)
Guo [41]	ACC	89.33 ± 2.67	89.93 ± 3.68	88.87 ± 6.16	N/A
	AC	87.36 ± 1.38	87.30 ± 4.68	87.82 ± 4.33	N/A
Yang [32]	Code1	75.08 ± 1.13	75.81 ± 1.20	74.75 ± 1.23	N/A
	Code2	80.04 ± 1.06	76.77 ± 0.69	82.17 ± 1.35	N/A
	Code3	80.41 ± 0.47	78.14 ± 0.90	81.66 ± 0.99	N/A
	Code4	86.15 ± 1.17	81.03 ± 1.74	90.24 ± 1.34	N/A
You [74]	PCA-EELM	87.00 ± 0.29	86.15 ± 0.43	87.59 ± 0.32	77.36 ± 0.44
Wong [75]	PR-LPQ + RF	93.92 ± 0.36	91.10 ± 0.31	96.45 ± 0.45	88.56 ± 0.63
Proposed method	PCVM	96.55 ± 0.2	97.23 ± 0.3	95.84 ± 0.5	93.25 ± 0.3

a subsequent comparative experiment, the prediction performance by our approach is obviously better than that of an SVM-based method and previous methods. We also found that prediction quality continues to improve with increasing dataset size. This finding underscores the value of this model to train and apply very large datasets, and suggests that further performance gains may be had by increasing the data size. Therefore, this proposed method is a reliable, efficient, and powerful PPI prediction model. It can be adopted to guide the validation of relevant experiments and to be an auxiliary tool for proteomics research.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors state no conflict of interest.

Authors' Contributions

Yanbin Wang, Zhuhong You, Liping Li, and Li Cheng considered the algorithm, arranged the datasets, and performed the analyses. Xi Zhou, Libo Zhang, Xiao Li, and Tonghai Jiang wrote the manuscript. All authors read and approved the final manuscript.

Acknowledgments

This work is supported in part by the National Science Foundation of China (Grant nos. 61722212 and 61572506) and in part by the Pioneer Hundred Talents Program of the Chinese Academy of Sciences. The authors would like to thank the editors and anonymous reviewers for their constructive advice.

References

[1] L. Licata, L. Briganti, D. Peluso et al., "MINT, the molecular interaction database: 2012 update," *Nucleic Acids Research*, vol. 40, no. D1, pp. D857–D861, 2012.

[2] G. D. Bader, I. Donaldson, C. Wolting, B. F. F. Ouellette, T. Pawson, and C. W. V. Hogue, "BIND—the biomolecular interaction network database," *Nucleic Acids Research*, vol. 29, no. 1, pp. 242–245, 2001.

[3] I. Xenarios, E. Fernandez, L. Salwinski et al., "DIP: the database of interacting proteins: 2001 update," *Nucleic Acids Research*, vol. 29, no. 1, pp. 239–241, 2001.

[4] O. Puig, F. Caspary, G. Rigaut et al., "The tandem affinity purification (TAP) method: a general procedure of protein complex purification," *Methods*, vol. 24, no. 3, pp. 218–229, 2001.

[5] M. Koegl and P. Uetz, "Improving yeast two-hybrid screening systems," *Briefings in Functional Genomics and Proteomics*, vol. 6, no. 4, pp. 302–312, 2008.

[6] U. Rüetschi, A. Rosén, G. Karlsson et al., "Proteomic analysis using protein chips to detect biomarkers in cervical and amniotic fluid in women with intra-amniotic inflammation," *Journal of Proteome Research*, vol. 4, no. 6, pp. 2236–2242, 2005.

[7] J. Sun, J. Xu, Z. Liu et al., "Refined phylogenetic profiles method for predicting protein-protein interactions," *Bioinformatics*, vol. 21, no. 16, pp. 3409–3415, 2005.

[8] I. Kotsireas, R. Melnik, and B. West, "Advances in mathematical and computational methods: addressing modern

challenges of science, technology, and society," in *AIP Conference Proceedings*, p. 1, Melville, NY, USA, 2011.

[9] I. Kotsireas, E. Lau, and R. Voino, "Exact implicitization of polynomial curves and surfaces," *ACM SIGSAM Bulletin*, vol. 37, no. 3, p. 78, 2003.

[10] I. Kotsireas and E. Zima, "Abstracts of WWCA 2011 in honor of Herb Wilf's 80th birthday," *ACM Communications in Computer Algebra*, vol. 45, no. 1/2, pp. 92–99, 2011.

[11] I. Kotsireas and E. Volcheck, "ANTS VI: algorithmic number theory symposium poster abstracts," *ACM SIGSAM Bulletin*, vol. 38, no. 3, pp. 93–107, 2004.

[12] I. Kotsireas, "Proceedings of the 2011 International Workshop on Symbolic-Numeric Computation," in *ISSAC '11 International Symposium on Symbolic and Algebraic Computation (Co-located with FCRC 2011)*, p. 18, San Jose, CA, USA, 2011.

[13] D. Vlachakis, A. Pavlopoulou, G. Tsiliki et al., "An integrated in silico approach to design specific inhibitors targeting human poly(A)-specific ribonuclease," *PLoS One*, vol. 7, no. 12, article e51113, 2012.

[14] D. Vlachakis, G. Tsiliki, A. Pavlopoulou, M. G. Roubelakis, S. C. Tsaniras, and S. Kossida, "Antiviral stratagems against HIV-1 using RNA interference (RNAi) technology," *Evolutionary Bioinformatics*, vol. 9, article EBO.S11412, 2013.

[15] D. Vlachakis, D. Tsagrasoulis, V. Megalooikonomou, and S. Kossida, "Introducing Drugster: a comprehensive and fully integrated drug design, lead and structure optimization toolkit," *Bioinformatics*, vol. 29, no. 1, pp. 126–128, 2013.

[16] D. Vlachakis, V. L. Koumandou, and S. Kossida, "A holistic evolutionary and structural study of *Flaviviridae* provides insights into the function and inhibition of HCV helicase," *PeerJ*, vol. 1, article e74, 2013.

[17] D. Vlachakis, D. G. Kontopoulos, and S. Kossida, "Space constrained homology modelling: the paradigm of the RNA-dependent RNA polymerase of dengue (type II) virus," *Computational and Mathematical Methods in Medicine*, vol. 2013, Article ID 108910, 9 pages, 2013.

[18] P. Vlamos, K. Lefkimmiatis, C. Cocianu, L. State, and Z. Luo, "Artificial intelligence applications in biomedicine," *Advances in Artificial Intelligence*, vol. 2013, Article ID 219137, 2 pages, 2013.

[19] P. Vlamos, V. Chrissikopoulos, and M. Psiha, "Building vulnerability: an interdisciplinary concept," *Key Engineering Materials*, vol. 628, pp. 193–197, 2014.

[20] P. Vlamos, A. Pateli, and M. Psiha, "Hybrid model for measurement of building vulnerability," *Key Engineering Materials*, vol. 628, pp. 237–242, 2014.

[21] P. Vlamos, "On the monotony of certain sequences," *Octogon Mathematical Magazine*, vol. 10, pp. 370-371, 2002.

[22] P. Vlamos and S. Tefarikis, "Numerical solution of partial differential equations," *The Mathematical Gazette*, vol. 50, pp. 179–449, 2005.

[23] A. Alexiou, M. Psiha, and P. Vlamos, "An integrated ontology-based model for the early diagnosis of Parkinson's disease," in *IFIP Advances in Information and Communication Technology*, pp. 442–450, Springer, Berlin, Heidelberg, 2012.

[24] A. Alexiou, M. Psiha, and P. Vlamos, *Towards an Expert System for Accurate Diagnosis and Progress Monitoring of Parkinson's Disease*, Springer International Publishing, 2015.

[25] A. T. Alexiou, P. Maria, J. Rekkas, and P. Vlamos, "A stochastic approach of mitochondrial dynamics," *World Academy of Science, Engineering and Technology*, vol. 55, pp. 77–80, 2011.

[26] A. Athanasios, P. Maria, T. Georgia, and V. Panayiotis, "Automated prediction procedure for Charcot-Marie-Tooth disease," in *13th IEEE International Conference on BioInformatics and BioEngineering*, pp. 1–4, Chania, Greece, 2013.

[27] M. Psiha and P. Vlamos, "Modeling neural circuits in Parkinson's disease," *Advances in Experimental Medicine and Biology*, vol. 822, pp. 139–147, 2015.

[28] R. Jansen, H. Yu, D. Greenbaum et al., "A Bayesian networks approach for predicting protein-protein interactions from genomic data," *Science*, vol. 302, no. 5644, pp. 449–453, 2003.

[29] T. N. Tran, K. Satou, and B. H. Tu, "Using inductive logic programming for predicting protein-protein interactions from multiple genomic data," in *Knowledge Discovery in Databases: PKDD 2005*, vol. 3721 of Lecture Notes in Computer Science, pp. 321–330, Springer, Berlin, Heidelberg, 2005.

[30] T. Hamp and B. Rost, "Evolutionary profiles improve protein-protein interaction prediction from sequence," *Bioinformatics*, vol. 31, no. 12, pp. 1945–1950, 2015.

[31] H. C. Yi, Z. H. You, D. S. Huang, X. Li, T. H. Jiang, and L. P. Li, "A deep learning framework for robust and accurate prediction of ncrna-protein interactions using evolutionary information," *Molecular Therapy - Nucleic Acids*, vol. 11, pp. 337–344, 2018.

[32] L. Yang, J. F. Xia, and J. Gui, "Prediction of protein-protein interactions from protein sequence using local descriptors," *Protein & Peptide Letters*, vol. 17, no. 9, pp. 1085–1090, 2010.

[33] Y. Zhang, D. Zhang, G. Mi et al., "Using ensemble methods to deal with imbalanced data in predicting protein-protein interactions," *Computational Biology and Chemistry*, vol. 36, pp. 36–41, 2012.

[34] Z.-H. You, M. C. Zhou, X. Luo, and S. Li, "Highly efficient framework for predicting interactions between proteins," *IEEE Transactions on Cybernetics*, vol. 47, no. 3, pp. 731–743, 2017.

[35] K. T. Simons, C. Kooperberg, E. Huang, and D. Baker, "Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions," *Journal of Molecular Biology*, vol. 268, no. 1, pp. 209–225, 1997.

[36] Y. Wang, Z. You, X. Li, X. Chen, T. Jiang, and J. Zhang, "PCVMZM: using the probabilistic classification vector machines model combined with a Zernike moments descriptor to predict protein-protein interactions from protein sequences," *International Journal of Molecular Sciences*, vol. 18, no. 5, 2017.

[37] C. von Mering, R. Krause, B. Snel et al., "Comparative assessment of large-scale data sets of protein-protein interactions," *Nature*, vol. 417, no. 6887, pp. 399–403, 2002.

[38] T. Berggard, S. Linse, and P. James, "Methods for the detection and analysis of protein-protein interactions," *Proteomics*, vol. 7, no. 16, pp. 2833–2842, 2007.

[39] K. C. Chou, "Pseudo amino acid composition and its applications in bioinformatics, proteomics and system biology," *Current Proteomics*, vol. 6, no. 4, pp. 262–274, 2009.

[40] J. Shen, J. Zhang, X. Luo et al., "Predicting protein-protein interactions based only on sequences information," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, no. 11, pp. 4337–4341, 2007.

[41] Y. Guo, L. Yu, Z. Wen, and M. Li, "Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences," *Nucleic Acids Research*, vol. 36, no. 9, pp. 3025–3030, 2008.

[42] W. Zhou, H. Yan, X. Fan, and Q. Hao, "Prediction of protein-protein interactions based on molecular interface features and the support vector machine," *Current Bioinformatics*, vol. 8, no. 1, pp. 3–8, 2013.

[43] L. Hua and P. Zhou, "Combining protein-protein interactions information with support vector machine to identify chronic obstructive pulmonary disease related genes," *Molecular Biology*, vol. 48, no. 2, pp. 287–296, 2014.

[44] S. Dohkan, A. Koike, and T. Takagi, "Prediction of protein-protein interactions using support vector machines," in *Proceedings. Fourth IEEE Symposium on Bioinformatics and Bioengineering*, pp. 165–173, Taichung, Taiwan, 2004.

[45] L.-P. Li, Y.-B. Wang, Z.-H. You, Y. Li, and J.-Y. An, "PCLPred: a bioinformatics method for predicting protein-protein interactions by combining relevance vector machine model with low-rank matrix approximation," *International Journal of Molecular Sciences*, vol. 19, no. 4, 2018.

[46] J.-Y. An, F.-R. Meng, Z.-H. You, Y.-H. Fang, Y.-J. Zhao, and M. Zhang, "Using the relevance vector machine model combined with local phase quantization to predict protein-protein interactions from protein sequences," *BioMed Research International*, vol. 2016, Article ID 4783801, 9 pages, 2016.

[47] J. Y. An, F. R. Meng, Z. H. You, X. Chen, G. Y. Yan, and J. P. Hu, "Improving protein-protein interactions prediction accuracy using protein evolutionary information and relevance vector machine model," *Protein Science*, vol. 25, no. 10, pp. 1825–1833, 2016.

[48] X. Y. Pan, Y. N. Zhang, and H. B. Shen, "Large-scale prediction of human protein-protein interactions from amino acid sequence based on latent topic features," *Journal of Proteome Research*, vol. 9, no. 10, pp. 4992–5001, 2010.

[49] P. Smialowski, P. Pagel, P. Wong et al., "The Negatome database: a reference set of non-interacting protein pairs," *Nucleic Acids Research*, vol. 38, Supplement 1, pp. D540–D544, 2010.

[50] S. Martin, D. Roe, and J. L. Faulon, "Predicting protein–protein interactions using signature products," *Bioinformatics*, vol. 21, no. 2, pp. 218–226, 2004.

[51] L. Li, Y. Liang, and R. L. Bass, "GAPWM: a genetic algorithm method for optimizing a position weight matrix," *Bioinformatics*, vol. 23, no. 10, pp. 1188–1194, 2007.

[52] J. Korhonen, P. Martinmäki, C. Pizzi, P. Rastas, and E. Ukkonen, "MOODS: fast search for position weight matrix matches in DNA sequences," *Bioinformatics*, vol. 25, no. 23, pp. 3181-3182, 2009.

[53] J. Yang and S. A. Ramsey, "A DNA shape-based regulatory score improves position-weight matrix-based recognition of transcription factor binding sites," *Bioinformatics*, vol. 31, no. 21, pp. 3445–3450, 2015.

[54] P.-T. Yap and R. Paramesran, "An efficient method for the computation of Legendre moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1996–2002, 2005.

[55] K. M. Hosny, "Exact Legendre moment computation for gray level images," *Pattern Recognition*, vol. 40, no. 12, pp. 3597–3605, 2007.

[56] J. D. Zhou, H. Z. Shu, L. M. Luo, and W. X. Yu, "Two new algorithms for efficient computation of Legendre moments," *Pattern Recognition*, vol. 35, no. 5, pp. 1143–1152, 2002.

[57] B. Fu, J. Zhou, Y. Li, G. Zhang, and C. Wang, "Image analysis by modified Legendre moments," *Pattern Recognition*, vol. 40, no. 2, pp. 691–704, 2007.

[58] G. A. Papakostas, E. G. Karakasis, and D. E. Koulouriotis, "Accurate and speedy computation of image Legendre moments for computer vision applications," *Image and Vision Computing*, vol. 28, no. 3, pp. 414–423, 2010.

[59] J. Xu, L. Xiang, Q. Liu et al., "Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images," *IEEE Transactions on Medical Imaging*, vol. 35, no. 1, pp. 119–130, 2016.

[60] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. A. Manzagol, "Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[61] A. Sankaran, P. Pandey, M. Vatsa, and R. Singh, "On latent fingerprint minutiae extraction using stacked denoising sparse autoencoders," in *IEEE International Joint Conference on Biometrics*, pp. 1–7, Clearwater, FL, USA, 2014.

[62] C. Tao, H. Pan, Y. Li, and Z. Zou, "Unsupervised spectral-spatial feature learning with stacked sparse autoencoder for hyperspectral imagery classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 12, pp. 2438–2442, 2015.

[63] Y. B. Wang, Z. H. You, X. Li et al., "Predicting protein-protein interactions from protein sequences by a stacked sparse autoencoder deep neural network," *Molecular BioSystems*, vol. 13, no. 7, pp. 1336–1344, 2017.

[64] H. Chen, P. Tino, and X. Yao, "Probabilistic classification vector machines," *IEEE Transactions on Neural Networks*, vol. 20, no. 6, pp. 901–914, 2009.

[65] Z. Xue, X. Yu, Q. Fu, X. Wei, and B. Liu, "Hyperspectral imagery classification based on probabilistic classification vector machines," in *Eighth International Conference on Digital Image Processing (ICDIP 2016)*, C. M. Falco and X. Jiang, Eds., Chengu, China, 2016.

[66] H. Chen, P. Tino, and X. Yao, "Efficient probabilistic classification vector machine with incremental basis function selection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 356–369, 2014.

[67] T. L. Bailey and C. Elkan, "Fitting a mixture model by expectation maximization to discover motifs in biopolymers," *Proceedings. Internation Conference on Intelligent Systems for Molecular Biology*, vol. 2, pp. 28–36, 1994.

[68] J. Huang and C. X. Ling, "Using AUC and accuracy in evaluating learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 3, pp. 299–310, 2005.

[69] C. C. Chang and C. J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, 2011.

[70] B. Liu, F. Liu, L. Fang, X. Wang, and K. C. Chou, "repDNA: a Python package to generate various modes of feature vectors for DNA sequences by incorporating user-defined physicochemical properties and sequence-order effects," *Bioinformatics*, vol. 31, no. 8, pp. 1307–1309, 2015.

[71] J. R. Bock and D. A. Gough, "Whole-proteome interaction mining," *Bioinformatics*, vol. 19, no. 1, pp. 125–134, 2003.

[72] S. Martin, D. Roe, and J.-L. Faulon, "Predicting protein-protein interactions using signature products," *Bioinformatics*, vol. 21, no. 2, pp. 218–226, 2005.

[73] L. Nanni and A. Lumini, "An ensemble of K-local hyperplanes for predicting protein-protein interactions," *Bioinformatics*, vol. 22, no. 10, pp. 1207–1210, 2006.

[74] Z. H. You, Y. K. Lei, L. Zhu, J. Xia, and B. Wang, "Prediction of protein-protein interactions from amino acid sequences with ensemble extreme learning machines and principal component analysis," *BMC Bioinformatics*, vol. 14, Supplement 8, article S10, 2013.

[75] L. Wong, Z. H. You, S. Li, Y. A. Huang, and G. Liu, "Detection of protein-protein interactions from amino acid sequences using a rotation forest model with a novel PR-LPQ descriptor," in *Advanced Intelligent Computing Theories and Applications*, pp. 713–720, Springer, Cham, Switzerland, 2015.

Hindawi
Complexity
Volume 2018, Article ID 9193248, 17 pages
https://doi.org/10.1155/2018/9193248

Research Article

Application of Data Mining Technology on Surveillance Report Data of HIV/AIDS High-Risk Group in Urumqi from 2009 to 2015

Dandan Tang,[1] **Man Zhang**,[2] **Jiabo Xu**,[3] **Xueliang Zhang**,[4] **Fang Yang**,[5] **Huling Li**,[1] **Li Feng**,[1] **Kai Wang**,[4] **and Yujian Zheng**[1]

[1]*College of Public Health, Xinjiang Medical University, Urumqi 830011, China*
[2]*Department of Information Engineering, Xinjiang Institute of Engineering, Urumqi, 830000, China*
[3]*College of Medical Engineering and Technology, Xinjiang Medical University, Urumqi 830011, China*
[4]*Department of Medical Engineering, The Affiliated Tumor Hospital, Xinjiang Medical University, Urumqi 830011, China*
[5]*Department of AIDS/STD Control and Prevention, Urumqi Center for Disease Control and Prevention, Urumqi, Xinjiang 830026, China*

Correspondence should be addressed to Kai Wang; wangkaimath@sina.com and Yujian Zheng; 147854307@qq.com

Received 29 May 2018; Accepted 17 September 2018; Published 10 December 2018

Guest Editor: Panayiotis Vlamos

Objective. Urumqi is one of the key areas of HIV/AIDS infection in Xinjiang and in China. The AIDS epidemic is spreading from high-risk groups to the general population, and the situation is still very serious. The goal of this study was to use four data mining algorithms to establish the identification model of HIV infection and compare their predictive performance. *Method*. The data from the sentinel monitoring data of the three groups of high-risk groups (injecting drug users (IDU), men who have sex with men (MSM), and female sex workers (FSW)) in Urumqi from 2009 to 2015 included demographic characteristics, sex behavior, and serological detection results. Then we used *age*, *marital status*, *education level*, and other variables as input variables and whether to infect HIV as output variables to establish four prediction models for the three datasets. We also used confusion matrix, accuracy, sensitivity, specificity, precision, recall, and the area under the receiver operating characteristic (ROC) curve (AUC) to evaluate classification performance and analyzed the importance of predictive variables. *Results*. The final experimental results show that random forests algorithm obtains the best results, the diagnostic accuracy for random forests on MSM dataset is 94.4821%, 97.5136% on FSW dataset, and 94.6375% on IDU dataset. The k-nearest neighbors algorithm came out second, with 91.5258% diagnostic accuracy on MSM dataset, 96.3083% diagnostic accuracy on FSW dataset, and 90.8287% diagnostic accuracy on IDU dataset, followed by support vector machine (94.0182%, 98.0369%, and 91.3571%). The decision tree algorithm was the poorest among the four algorithms, with 79.1761% diagnostic accuracy on MSM dataset, 87.0283% diagnostic accuracy on FSW dataset, and 74.3879% accuracy on IDU. *Conclusions*. Data mining technology, as a new method of assisting disease screening and diagnosis, can help medical personnel to screen and diagnose AIDS rapidly from a large number of information.

1. Introduction

Acquired immunodeficiency syndrome (AIDS) is a malignant infectious disease with a very high fatality rate caused by human immunodeficiency virus (HIV) [1]. It alters the immune system making people much more vulnerable to infections and diseases [2]. Up to now, the HIV/AIDS epidemic has been one of the most important and crucial public health problems facing both developed and developing nations. Since the first case of HIV infection of China discovered in 1985, the number of the infected patients has been increasing year by year. The spread trend of AIDS in China has not been fundamentally controlled; AIDS prevention and control situation in Xinjiang is even severer. Xinjiang Uygur Autonomous Region is one of the provinces hardest hit by AIDS in China. The first HIV/AIDS case in Xinjiang was reported in 1995. At the end of 2011, the cumulative total of HIV/AIDS cases reported in Xinjiang accounted for

7.7% of all cumulative total of HIV/AIDS cases in the country, ranking the fifth position in China [3]. The total number of HIV/AIDS reported cases from 2004 to 2015 had been accumulated to 14,696, and it accounted for 5.56% of the total number of AIDS patients reported in China. There were also 3830 people died of HIV, which took up 4.56% of the total death cases induced by AIDS. The reported AIDS cases increased from 20 to 1868 with the average annual growth rate of 28.74, and the reported deaths increased from 5 to 680 with the average annual growth rate of 28.74 in the past decades, which were higher than that of the national average annual growth level [4]. Urumqi, the capital of Xinjiang Uygur Autonomous Region, is one of the main districts of AIDS infection in Xinjiang, and its AIDS epidemic has been consistently high. The largest group of HIV infection is injecting drug users in Urumqi. But in the late 2011, the proportion of the sexual route of transmission of infection is more than the intravenous drug users sharing syringes; the infection became the first way. More and more sexual partners, men and men crowd into the spread of AIDS high-risk groups [5, 6]. The situation of stemming the spread of HIV in persons at high risk of exposure and blocking the AIDS epidemic moving from high-risk groups to the general population proliferation is still very flinty. Therefore, HIV infection continues to be a major global public health issue.

Data mining is a newly developing technology based on machine study in artificial intelligence and database, and it can be classified into two categories: unsupervised learning and supervised learning [7]. Data mining is the process of selecting, exploring, and modeling large amounts of data, which aims at discovering unknown patterns or relationships and infer prediction rules from the data [8]. In the recent years, great advancement has been achieved in the medical research of data mining. Studies have applied data mining to analyze volumes of data, explore unknown factors of disease, develop predictive models, and produce meaningful reports in different medical research fields [9–11]. In the new period, the study of prevention, diagnosis, and treatment of HIV disease entered a new phase. A lot of domestic and foreign researchers have done on using the data mining technology to discover the relationship of the AIDS patient's potential factors and the result of treatment based on HIV surveillance data or comprehensive clinical data [12]. Oliveira et al. built multilayer artificial neural networks (MLP), naive Bayesian classifiers (NB), support vector machines (SVM), and the k-nearest neighbor algorithm (KNN) in order to identify the main factors influencing reporting delays of HIV-AIDS cases within the Portuguese surveillance system. The results of this study strongly suggested that MLP provided the best results, with a higher classification accuracy (approximately 63%), precision (approximately 76%), and recall (approximately 60%) [13]. Wang et al. had developed three computational modeling methods to predict virological response to therapy from HIV genotype and other clinical information. The comparison results showed that an artificial neural network (ANN) models were significantly inferior to random forests (RF) and support vector machines (SVM) [14]. Hai-Lei, et al. constructed a 133 HIV carriers forecasting model based on support vector machines (SVM), and the HIV carriers were found in the port of a province in China during the period of 2004–2009. The overall accuracy rate of forecasting model was 90.60%, and its sensibility and specificity were 90.29% and 90.90%, respectively [15]. Hailu compared the prediction of the different data mining technologies, which were used to develop the HIV testing prediction model. Four popular data mining algorithms (decision tree, naive Bayes, neural network, and logistic regression) were used to build the model that predicted whether an individual was being tested for HIV. The final experimentation results indicated that the decision tree (random tree algorithm) performed the best with an accuracy of 96% [16].

However, in previous studies, few researches considered the use of data mining methods to construct predictive mathematical models of AIDS high-risk group based on several potential risk factors for surveillance report data. This paper aims at using data mining technology to identify the main factors influencing on the status of AIDS high-risk group infection (including injecting drug user (IDU), female sex worker (FSW), and men who have sex with men (MSM)) on surveillance report data in Urumqi and compare the prediction power of the different forecast models based on data mining technology. In order to accomplish this objective, several data mining classification models were considered, namely, random forests (RF), support vector machine (SVM), k-nearest neighbors (KNN), and decision tree (DT), using a 10-fold cross-validation technique. The classification performance was evaluated in terms of a confusion matrix, accuracy, sensitivity, specificity, precision, recall, and AUC values of the receiver operating characteristic (ROC) curves.

2. Materials and Methods

2.1. Study Population. The target populations that met the inclusion criteria in this paper were selected from the data between 2009 and 2015 that the sentinel surveillance of CDC at all levels in Urumqi was reported to China CDC Information System. There are three populations at higher risk of HIV exposure that were considered, including FSW which was defined as women who engaged in commercial sex trade during the investigation; IDU was defined as who takes oral, inhaling, or injecting heroin, cocaine, opium, morphine, marijuana, k-powder, methamphetamine, ecstasy, leprosy, etc.; and MSM was defined as people who have had intercourse or oral sex in the past years.

2.2. Data Source. The data applied in this paper consisted of three datasets from the higher risk of HIV/AIDS exposure populations collected between 2009 and 2015 by the Urumqi CDC. The three datasets are FSW dataset that included 9090 FSWs and 53 attributes, MSM dataset that included 5304 MSM and 57 attributes, and IDU dataset that included 7337 IDUs and 56 attributes. The collected data had three core survey questionnaires: FSW questionnaire, MSM questionnaire, and IDU questionnaire. The survey items included demographic characteristics (age-at-birth, gender,

marital status, nation, place of household registration and educational level, etc.), serological detection results (antibody detection of HIV, syphilis, and HCV), high-risk behaviors factors (drug abuse behavior and sexual behavior), and AIDS prevention strategies and measures (the awareness of AIDS/HIV prevention knowledge, the conditions of prevention, and intervention service and situation of test-accepting).

2.3. Data Preprocessing. Data preprocessing plays an important role in the data mining tasks. Data preprocessing contains many kinds of methods for different preprocessing purposes, including data cleaning, data transformation, and data reduction [17]. In this study, we have selected some appropriate methods to optimize the original dataset. First, the attributes unrelated to the data mining goal were removed in advance, such as questionnaire ID, investigation date, and area codes. And the attributes with a large number of missing values were also removed. Second, the data grouping technique was used to simplify the data mining task. In the multiple distinct values of some attributes, such as age, a numerical variable was discretize into different category groups based on WHO standard for age classification. Ethnicity, originally with 56 distinct values, were converted into three distinct categories according to the constituent ratio of different nationalities as Hans, Uygurs, and others. In addition, simple statistical computations were performed with the R language and software environment, version 3.4.3, to analyze the distribution of the attributes. The dependent variable (T03C) was a binary outcome variable of people who has been tested for HIV with two categories: 0 and 1, where 0 means the HIV test results were negative and 1 means the HIV test results were positive. The results of the attributes description are presented in Tables 1, 2, and 3.

Table 1 shows a total of 5304 MSM respondents tested for HIV. Among them, 377 (7.11%) were detected as HIV positive and 4927 (92.9%) were detected as HIV negative. Table 2 shows a total of 9090 FSW respondents who had received a HIV test; 9041 (99.5%) were HIV-positive, while only 49 (0.5%) were HIV negative. Table 3 shows 7337 IDU respondents who had accepted a HIV test; the HIV negative and positive were 6087 (83%) and 1250 (17%), respectively. These results indicate that there is a need of balancing these two classes of the three datasets. In this article, we employed the Synthetic Minority Over-sampling Technique (SMOTE) [18] to dispose unbalanced samples. In SMOTE algorithm, majority class samples use the undersampling method and minority class samples use the oversampling technique. It potentially performs better than simple oversampling and it is widely used [19, 20].

2.4. Attribute Selection. In a data mining task, the selection of the input attributes is usually a highly important step to improve the classification ability of the models, to reduce the classier complexity, to save the computational time, and to simplify the obtained results. Filtering and wrapper are two main different approaches to select a subset of attributes from all of the attributes used in machine learning. Filtering is to make an independent assessment based on the data general characteristics. Wrapper is to select a feature subset using the evaluation function based on a machine learning algorithm [21]. In this paper, the wrapper methods based on random forests (RF) was used to select the attributes as the inputs of the classification model. RF algorithm is an ensemble learning method based on the aggregation of a large number of decision trees and has proved to be very powerful in many different applications [22–24]. A feature selection based on the random forest classifier has been found to provide multivariate feature importance scores, which are relatively easy to obtain and have been successfully applied to high dimensional data [25, 26]. The quantification procedures of the variable importance scores can be described as follows: computing the variable importance score and permuting score, then selecting the features that have more contribution to classification model, and building models through the feature evaluation criteria of random forest algorithm. The Gini importance considers conditional higher-order interactions among the variables and might be a preferable ranking criterion than a univariate measure [27, 28] and is the feature importance evaluation criteria of random forest algorithm which was used in this study.

2.5. Classification Models

2.5.1. Random Forests (RF). The first algorithm for random decision forests was created by Ho (1995) [29], and its extension version was developed by Breiman [30]. The RF is an ensemble learning method based on decision tree and has been successfully used in several types of classification and regression, especially for accurate identification of disease diagnosis problems [31–33]. RF builds a large number of decision trees using a bootstrap sample with replacement from the training set and predicts the class of each tree according to the test set, and the final RF prediction class is presented based on the majority of the votes [34]. It has been shown to give excellent performance on numerical and categorical data.

2.5.2. Support Vector Machine (SVM). Support vector machine, a novel type of learning machine derived from statistical learning theory, constructs a hyperplane or set of hyperplanes in high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection, function estimation, and high-dimensional pattern recognition problems [35–38]. The SVM mainly deals with the problems of binary classification. In addition to performing linear classification, SVM can efficiently perform nonlinear classification through kernel techniques [39] implicitly mapping their inputs into high-dimensional feature spaces. SVM categorization model can be constructed in two ways, as follows: (1) converting the input space into higher dimensional feature space by a nonlinear mapping function. (2) Building the separating hyperplane based on maximum distance from the closest points of the training set [40].

Table 1: Details of the attributes of the MSM dataset.

Variables	Description	Category	Total number	Percentage (%)
A01B	Monitoring sites	Sayibak District	2981	56.2
		Xinshi District	624	11.8
		Shuimogou District	361	6.8
		Tianshan District	1338	25.2
A06	Sample source	Bar/dancehall/ tearooms/club	783	14.8
		Bath/sauna/ pedicure/massage	669	12.6
		Park/public toilet/ grassland	188	3.5
		Network recruiting	3605	68.0
		Others	59	1.1
B01	Age	1(15–17)	27	0.5
		2(18–28)	2704	51.0
		3(29–40)	2140	40.3
		4(41–48)	362	6.8
		5(49–55)	59	1.1
		6(56–65)	8	0.2
		7(>66)	4	0.1
B02	Marital status	Unmarried	4377	82.5
		Married	602	11.3
		Cohabitation	76	1.4
		Divorced or widowed	249	4.7
B03	The location of household register	Xinjiang Uygur Autonomous Region	4620	87.1
		Others	684	12.9
B04	Nation	Hans	4546	85.7
		Uygurs	333	6.3
		Others	425	8.0
B05	Inhabit time	<3 months	123	2.3
		3–6 months	59	1.1
		7–12 months	96	1.8
		1–2 years	332	6.3
		>2 years	4694	88.5
B06	Educational level	Illiteracy	8	0.2
		Primary school	43	0.8
		Junior middle school	346	6.5
		High school or technical school	1081	20.4
		College or above	3826	72.1
C08	Knowledge and awareness of HIV	No	186	3.5
		Yes	5118	96.5
D01	Have you ever had anal sex with a person of the same sex in the last six months	No	356	6.7
		Yes	4948	93.3
D03	Did you use a condom for sex with the same sex last time	No	1169	22.0
		Yes	4135	78.0

Table 1: Continued.

Variables	Description	Category	Total number	Percentage (%)
E01	Have you had any commercial sex with people of the same sex last 6 months	No	5024	94.7
		Yes	280	5.3
F01	Did you have sex with the opposite sex last 6 months	No	4801	90.5
		Yes	503	9.5
G01	Did you take drugs	No	5270	99.4
		Yes	34	0.6
H01	Have you ever been diagnosed with an STD in the last year	No	5168	97.4
		Yes	136	2.6
I01	Have you ever received a condom promotion or HIV/AIDS counselling and testing to prevent HIV/AIDS	No	633	11.9
		Yes	4671	88.1
I02	Have you ever received a community medication to maintain or providing or exchanging cleaning needles to prevent HIV/AIDS	No	5268	99.3
		Yes	36	0.7
I03	Have you ever received a companion education to prevent HIV/AIDS	No	1709	32.2
		Yes	3595	67.8
J01	Has HIV been tested in the last year	No	1726	32.5
		Yes	3578	67.5
T04C	Syphilis test results	No	4979	93.9
		Yes	325	6.1
T05C	Hepatitis test results	Yes	39	0.7
		No	5265	99.3
T03C	HIV test results	No	4927	92.9
		Yes	377	7.1

2.5.3. K-Nearest Neighbors (KNN). The k-nearest neighbors algorithm (KNN) is the simplest but more powerful nonparametric classification method of all data mining methods, since it is a type of instance-based or lazy learning algorithm [41]. KNN classifier has been widely used in many fields, such as text classification, pattern recognition, and disease detection and diagnosis, based on the advantages such as simplicity, high efficiency, and easy to implement [42, 43]. KNN arithmetic idea mainly considers three points: the value of k, distance measurement, and decision rules of classification. The k, as a user-defined constant, will directly affect the KNN classification performance. And the distance metric measures commonly use Euclidean distance, Manhattan distance, and Minkowski distance. The decision rules of classification depend on the majority voting.

2.5.4. Decision Trees (DT). A decision tree is a kind of commonly used data mining method with many advantages such as easy to understand, readable, and quick classification [44]. A decision tree is the organization of the nodes that make decisions like a tree, which consists of decision nodes, branches, and leaf nodes. Each decision node represents a data category or attributes to be classified, and each leaf node represents a result [45]. The whole decision-making process starts from the root decision node, and from top to bottom, it is determined until the classification results are given. There are three commonly used typical decision tree algorithms in data mining at present, such as ID3 algorithm, C4.5 algorithm, and CART algorithm [46].

2.6. Performance Evaluation. In this paper, a confusion matrix and some indicators including accuracy, sensitivity, specificity, precision, recall, and the receiver operating characteristic (ROC) curve were used to appraise the performance of the four classification models. A 10-fold cross-validation was applied to RF, SVM, KNN, and DT validation. A confusion matrix consists of the parts shown in Table 4. In Table 4, TP (true positive) is the positive records of the correct classification, TN (true negative) is the negative records of the correct classification, FP (false positive) is the positive records of the incorrect classification, and FN (false negative) is the negative records of the incorrect classification.

Several important measures, such as accuracy, sensitivity, specificity, precision, and recall, can be calculated by using the confusion matrix. The accuracy is the number of samples correctly classified. The sensitivity is a description of measuring the proportion of correctly classified positive samples. The specificity is a description of measuring the proportion of correctly classified negative samples. The precision is a description of the number of positive samples to the proportion of all predicted positive samples. The recall is a description of the ratio of positive samples

Table 2: Details of the attributes of the FSW dataset.

Variables	Description	Category	Total number	Percentage (%)
A01B	Monitoring sites	Sayibak District	2557	28.1
		Xinshi District	1099	12.1
		Economic Development District	522	5.7
		Shuimogou District	2653	29.2
		Tianshan District	2259	24.9
A06	Sample source	Sauna/bath center	778	8.6
		Nightclub	3157	34.7
		Karaoke hall/ ballroom/bar	2388	26.3
		Guesthouse/hotel	551	6.1
		Foot washing room/ hair salon	1551	17.1
		Roadside shop/little dine	656	7.2
		Street	9	0.1
B01B	Age	1(15–17)	109	1.2
		2(18–28)	6479	71.3
		3(29–40)	2028	22.3
		4(41–48)	394	4.3
		5(49–55)	66	0.7
		6(56–65)	7	0.1
B02	Marital status	Unmarried	5288	58.2
		Married	2359	26.0
		Cohabitation	1051	11.6
		Divorced or widowed	392	4.3
B03	The location of household register	Xinjiang Uygur Autonomous Region	4947	54.4
		Others	4143	45.6
B04	Nation	Hans	7405	81.5
		Uygurs	785	8.6
		Others	900	9.9
B05	Educational level	Illiteracy	118	1.3
		Primary school	949	10.4
		Junior middle school	3738	41.1
		High school or technical school	3383	37.2
		College or above	902	9.9
B06	How long were you working here this time	>=1 year	3180	35.0
		6–12 months	1930	21.1
		1–6 months	2773	30.5
		<1 months	1207	13.3
C08	Knowledge and awareness of HIV	No	401	4.4
		Yes	8689	95.6
D01	Did you use condoms with your guests the last time	No	932	10.3
		Yes	8158	89.7
D02	How often did you use condoms when you have sex with a guest last month	Never used	190	2.1

Table 2: Continued.

Variables	Description	Category	Total number	Percentage (%)
		Sometimes used	2160	23.8
		Every time used	6740	74.1
E01	Did you take drugs	No	9029	99.3
		Yes	61	0.7
F01	Have you ever been diagnosed with an STD in the last year	No	9060	99.7
		Yes	30	0.3
G01	Have you ever received a condom promotion or HIV/AIDS counselling and testing to prevent HIV/AIDS	No	504	5.5
		Yes	8586	94.5
G02	Have you ever received a community medication to maintain or providing or exchanging cleaning needles to prevent HIV/AIDS	No	8950	98.5
		Yes	140	1.5
G03	Have you ever received a companion education to prevent HIV/AIDS	No	2485	27.0
		Yes	6632	73.0
H01	Has HIV been tested in the last year	No	4429	48.7
		Yes	4661	51.3
T04C	Syphilis test results	No	8904	98.0
		Yes	186	2.0
T05C	Hepatitis test results	No	8986	98.9
		Yes	104	1.1
T03C	HIV test results	No	9041	99.5
		Yes	49	0.5

to the total number of positive samples. The accuracy, sensitivity, specificity, precision, and recall are defined as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100\%, \quad (1)$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\%, \quad (2)$$

$$\text{Specificity} = \frac{\text{TP}}{\text{TN} + \text{FP}} \times 100\%, \quad (3)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100\%, \quad (4)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100\%. \quad (5)$$

The ROC curve is originally derived from statistical decision theory, which can comprehensively describe the classification performance of the classifiers with different discriminant thresholds [47]. The vertical axis of the ROC curve is TP rate, and the horizontal axis is FP rate. However, in a practical application, the AUC (the area under the ROC curve) is often used to evaluate the performance of the classifier.

3. Experimental Results

R is an open source programming language and software environment for statistical computing and graphics. Based on the R language environment, the implementation of each algorithm in this experiment is carried out. Here, we used SMOTE (DMwR), randomForest (randomForest), ksvm (kernlab), kknn (kknn), and rpart (rpart) packages. All experiments were validated with a 10-fold cross-validation technique in order to present a more stable accuracy rate after applying the four classification models. Some evaluation indexes were used to compare the classification performance of four data mining algorithms.

Table 5 shows the three original datasets and the three artificial datasets obtained using SMOTE algorithm. It is evident that the original datasets are biased; the imbalance rate of each original datasets is 13.0689, 184.5102, and 4.8696, respectively. In order to achieve the data balance to avoid the result bias, we used SMOTE algorithm combining the oversampling the minority class and undersampling the majority class techniques. We apply the function SMOTE in the DMwR package in R software. The three main parameters of function SMOTE are perc.over, perc.under, and k. The parameter perc.over and perc.under control the amount of oversampling of the minority classes and undersampling of the majority classes, respectively. The parameter k controls the way of the new examples created. For the parameters in the SMOTE algorithm, the value of k was set to 5. For the initial dataset of MSM with 377 minority samples and 4927 majority samples, we set the parameters perc.over = 1200 and perc.under = 110, respectively. Firstly, the number of minority samples was increased; a total of $1200 \times 377/100$ new minority samples were generated. The original minority samples and the new minority samples consisted of the new dataset. Secondly, sampling the majority sample, we obtain a new sample of the majority, which is $(110/100) \times 1200 \times 377/100$. We put the new sample of the majority into the new dataset which was created

Table 3: Details of the attributes of the IDU dataset.

Variables	Description	Category	Total number	Percentage (%)
A01B	Monitoring sites	Sayibak District	2147	32.9
		Xinshi District	892	12.2
		Shuimogou District	1802	24.6
		Tianshan District	1922	26.2
		Toutun River District	56	0.8
		Urumqi County	248	3.4
A06	Sample source	Compulsory detoxification setting	1617	22.0
		Community	5063	69.0
		Methadone clinic (urine test positive)	657	9.0
B02	Age	1(15–17)	49	0.7
		2(18–28)	1719	23.4
		3(29–40)	3493	47.6
		4(41–48)	1721	23.5
		5(49–55)	305	4.2
		6(56–65)	43	0.6
		7(>66)	5	0.1
B01	Gender	Male	6549	89.3
		Female	788	10.7
B03	Marital status	Unmarried	2586	35.2
		Married	3241	44.2
		Cohabitation	225	3.1
		Divorced or widowed	1285	17.5
B04	The location of household register	Xinjiang Uygur Autonomous Region	6762	92.2
		Others	575	7.8
B05	Nation	Hans	2452	33.4
		Uygurs	3880	52.9
		Others	1005	13.7
B06	Educational level	Illiteracy	377	5.1
		Primary school	1561	21.3
		Junior middle school	3231	44.0
		High school or technical school	1673	22.8
		College or above	495	6.7
C08	Knowledge and awareness of HIV	No	139	1.9
		Yes	7198	98.1
D01	How many drugs did you use at present	1 kind	6618	90.2
		2 kinds	646	8.8
		3 kinds	57	0.8
		4 kinds	13	0.2
		5 kinds	2	0.0
		6 kinds	1	0.0
D02	Did you take drugs	Sometimes used	2160	23.8
		Every time used	6740	74.1
		No	1812	24.7

Table 3: Continued.

Variables	Description	Category	Total number	Percentage (%)
E01	Have you ever had sex last month	Yes	5525	75.3
		No	4562	62.2
F01	Have you ever had sex with a commercial partner in the last year	Yes	2775	37.8
		No	6516	88.8
G01	Have you ever received a condom promotion or HIV/AIDS counselling and testing to prevent HIV/AIDS	Yes	821	11.2
		No	1453	19.8
G02	Have you ever received a community medication to maintain or providing or exchanging cleaning needles to prevent HIV/AIDS	Yes	5884	80.2
		No	3056	41.7
G03	Have you ever received a companion education to prevent HIV/AIDS	Yes	4281	58.3
		No	3353	45.7
H01	Has HIV been tested in the last year	Yes	3984	54.3
		No	3080	42.0
T04C	Syphilis test results	Yes	4257	58.0
		No	7093	96.7
T05C	Hepatitis test results	Yes	244	3.3
		No	3389	46.2
T03C	HIV test results	Yes	3948	53.8
		No	6087	83.0
		Yes	1250	17.0

Table 4: Confusion matrix for the two-class problem.

	Predicted negative	Predicted positive
Actual negative	TN	FP
Actual positive	FN	TP

Table 5: Description of original data and balanced data.

Dataset	Minority class	Majority class	Samples in total	Imbalance rate
MSM (original)	377	4927	5304	13.0689
MSM (SMOTE)	4901	4976	9877	1.0153
FSW (original)	49	9041	9090	184.5102
FSW (SMOTE)	9849	9898	19,747	1.0049
IDU (original)	1250	6087	7337	4.8696
IDU (SMOTE)	6250	6300	12,550	1.008

above. Eventually, in this new dataset, both the minority sample and the majority sample were $(1 + 1200/100) \times 377$ and $(110/100) \times 1200 \times 377/100$, respectively. For the initial dataset of FSW with 49 minority samples and 9041 majority samples, we set the parameters perc.over = 20,000 and perc.under = 101. The oversampling and undersampling algorithms also were utilized in the MSM dataset. The result demonstrated the new dataset with minority samples $(1 + 20{,}000/100) \times 49$ and majority samples $(101/100) \times 20{,}000/49/100$. For the initial dataset of IDU with 1250 minority samples and 6087 majority samples, setting the parameters perc.over = 400 and perc.under = 216, the minority sample and the majority sample were $1 + 400/100 \times 1250$ and $216/100 \times 400 \times 1250/100$, respectively.

Figures 1, 2, and 3 describe the importance of the sorted variables of the three datasets (MSM dataset, FSW dataset, and IDU dataset) according to the Gini index criterion from RF. From Figure 1, for the MSM dataset, the most important variables are B01, B06, A01B, A06, and B05. The least important variables are I02, G01, H01, and D01. From Figure 2, for the FSW dataset, the most important variables are B01B, T05C, A06, B05, and B06. The least important variables are F01, G02, C08, E01, and D01. From Figure 3, for the IDU dataset, the most important variables are B02, A01, T05C, B06, and B05. The least important variables are C08, B04, T04C, F01, and D01. Finally, applying the rank + MeanDecreaseGini method of attribute selection method, variables were ranked based on their importance in classifying the HIV patients. We also asked the CDC doctors about the importance of lower-ranking attributes, combining the two methods agree that B01, B06, A01B, A06, B05, B04, B02, D03, I03, J01, I01, B03, F01, T04C, and E01 as the main subset of attributes important in predicting the HIV patients from MSM population, B01B, T05C, A06, B05, B06, B04, B02, A01B, D02, H01, T04C, G03, B03, and G01 as the main subset of attributes important in predicting the HIV patients from female sex workers population, and B02, A01, T05C, B06, B05, B03, A06, D02, H01, G02, G03, E01, G01, and B01 as the main subset of attributes important in predicting the HIV patients from drug users population. The detailed descriptions of the selected attributes were shown in Tables 6, 7, and 8.

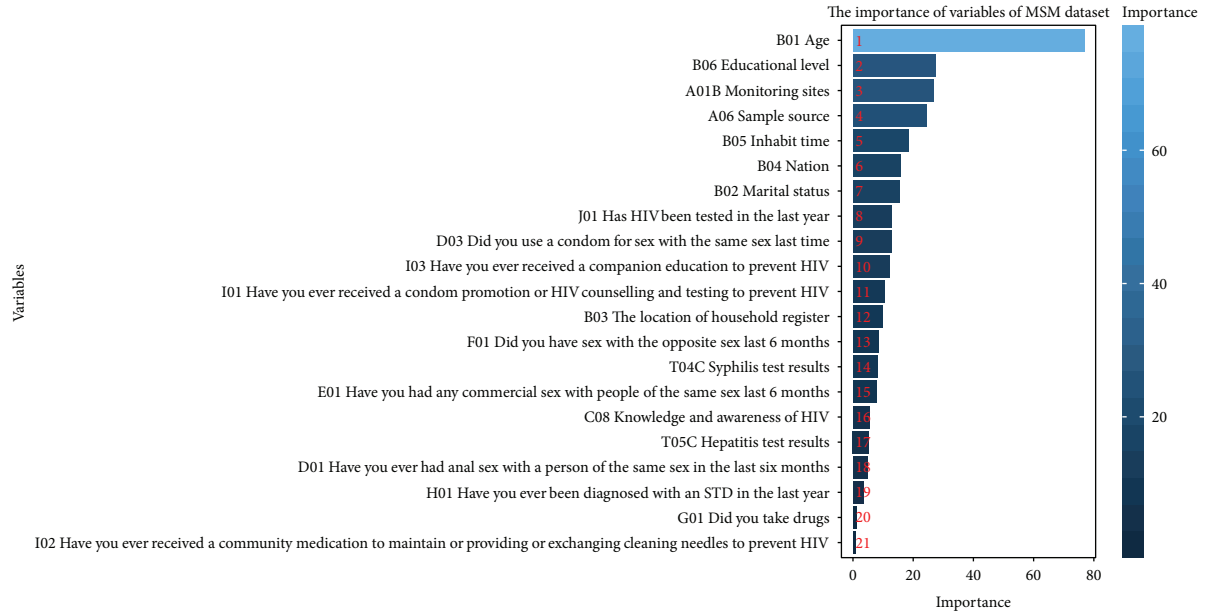


Figure 1: The importance of variables of MSM dataset.

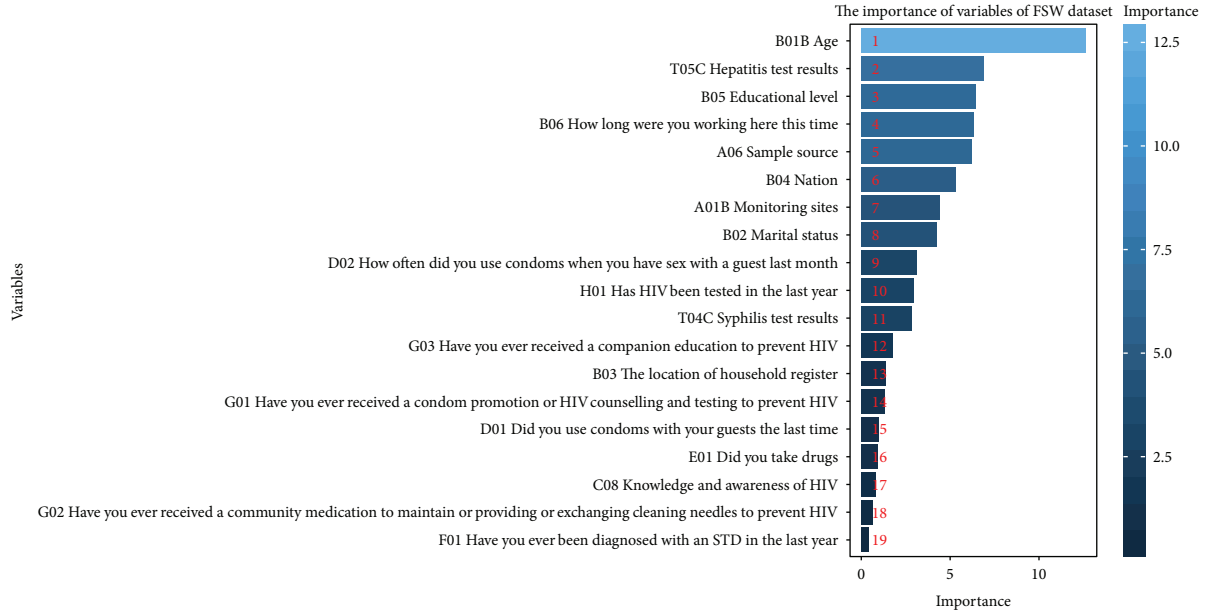


Figure 2: The importance of variables of FSW dataset.

Figures 4, 5, and 6 show the ROC curve obtained for the three datasets with the four classifiers. The AUC scores for RF, SVM, KNN, and DT on MSM dataset are 0.9802, 0.9401, 0.9747, and 0.7917; 0.9981, 0.9803, 0.9967, and 0.8702 on FSW dataset; and 0.9874, 0.9135, 0.9802, and 0.7438 on IDU dataset. It is obvious that RF performed significantly better than the other three classifiers. The AUC scores achieved for MSM dataset, FSW dataset, and IDU datasets are 0.9802, 0.9981, and 0.9874, respectively. The maximum value of the AUC (0.9981) was obtained for the FSW dataset with RF algorithm. Moreover, the value of AUC of DT algorithm with IDU dataset is 0.7438 which is the minimum of all AUC scores.

Figures 7, 8, and 9 depict the classification performance when the four classifiers are applied on MSM dataset, FSW dataset, and IDU dataset, respectively. The accuracy, precision, and recall for RF, SVM, KNN, and DT on the three datasets were compared. For the MSM dataset (Figure 7), the SVM model achieved a classification accuracy of 87.8404%, with a precision of 89.5130% and a recall of 85.5132%. The KNN model had a classification accuracy of 91.5258%, with a precision of 89.5130% and a recall of 85.5132%. For the

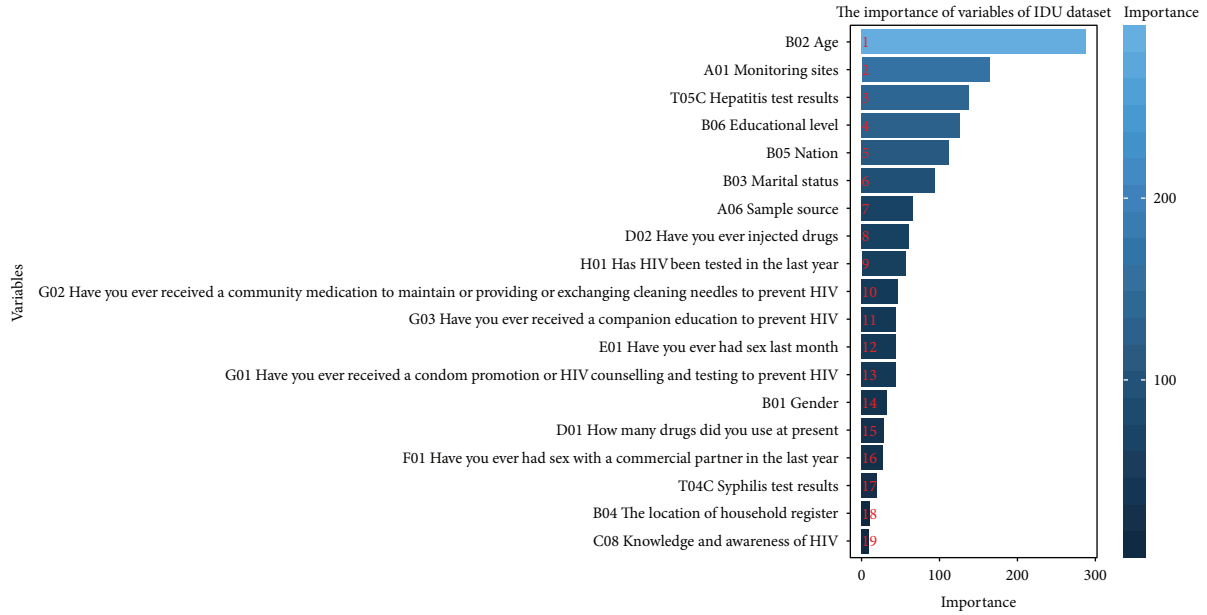


Figure 3: The importance of variables of IDU dataset.

Table 6: Selection attributes used in models of MSM dataset.

Rank	Attribute	MeanDecreaseGini
1	B01: age	76.8033
2	B06: educational level	27.1032
3	A01B: monitoring sites	26.0119
4	A06: sample source	23.9942
5	B05: inhabit time	18.3735
6	B04: nation	16.2218
7	B02: marital status	14.9883
8	D03: did you use a condom for sex with the same sex last time	12.7123
9	I03: have you ever received a companion education to prevent HIV/AIDS	12.2440
10	J01: has HIV been tested in the last year	12.1464
11	I01: have you ever received a condom promotion or HIV/AIDS counselling and testing to prevent HIV/AIDS	10.1819
12	B03: the location of household register	9.7513
13	F01: did you have sex with the opposite sex last 6 months	8.5185
14	T04C: syphilis test results	8.2889
15	E01: have you had any commercial sex with people of the same sex last 6 months	7.6851

decision tree, the accuracy, precision, and recall were 76.7440%, 77.6199%, and 74.6582%, respectively. The random forest algorithm performed best among the four evaluated models with an accuracy of 94.4821%, a precision of 98.5511%, and a recall of 90.2061%.

For the FSW dataset (Figure 8), the final experimental results demonstrated that the random forest algorithm showed the best with an accuracy of 97.5136%, and the precision and recall were 97.4638% and 91.6160%, respectively. The KNN model came out to be the second with a classification accuracy of 96.3083%, and the precision and recall were 97.4210% and 95.1163%, respectively, followed by SVM model with a classification accuracy of 93.3560%, the precision and recall equal to 94.1554% and 92.4155%, respectively. The decision tree has also performed the least classification accuracy of 85.0408%, and the precision and recall were 86.9467% and 82.3739%, respectively.

For the IDU dataset (Figure 9), the RF classifier showed the best predictive performances; the accuracy, precision, and recall gave 94.6375%, 97.4638%, and 91.6160%, respectively. In the SVM model, they were 83.4821%, 84.8141%, and 81.4080%, respectively. As shown in the confusion matrix in Table 10, the KNN learning algorithm scored an accuracy of 90.8287%; the precision and recall were 94.7831%, 86.3360%, respectively. Using the decision tree had a lower overall performance, with an accuracy of

Table 7: Selection attributes used in models of FSW dataset.

Rank	Variables	MeanDecreaseGini
1	B01B: age	12.6253
2	T05C: hepatitis test results	6.7033
3	A06: sample source	6.6001
4	B05: educational level	6.3421
5	B06: how long were you working here this time	6.1513
6	B04: nation	5.2128
7	B02: marital status	4.6192
8	A01B: monitoring sites	4.4660
9	D02: how often did you use condoms when you have sex with a guest last month	2.9029
10	H01: has HIV been tested in the last year	2.8776
11	T04C: syphilis test results	2.8470
12	G03: have you ever received a companion education to prevent HIV/AIDS?	1.6805
13	B03: the location of household register	1.4158
14	G01: have you ever received a condom promotion or HIV/AIDS counselling and testing to prevent HIV/AIDS	1.2143

Table 8: Selection attributes used in models of IDU dataset.

Rank	Variables	MeanDecreaseGini
1	B02: age	292.3608
2	A01: monitoring sites	166.8695
3	T05C: hepatitis test results	142.0867
4	B06: educational level	125.3663
5	B05: nation	112.2430
6	B03: marital status	92.2254
7	A06: sample source	63.6016
8	D02: have you ever injected drugs	58.3517
9	H01: has HIV been tested in the last year	55.1894
10	G02: have you ever received a community medication to maintain or providing or exchanging cleaning needles to prevent HIV/AIDS	45.0500
11	G03: have you ever received a companion education to prevent HIV/AIDS	44.9624
12	E01: have you ever had sex last month	43.5729
13	G01: have you ever received a condom promotion or HIV/AIDS counselling and testing to prevent HIV/AIDS	42.9014
14	B01: gender	33.0323

71.2271%, precision and recall were 69.8690% and 74.2400%, respectively.

The other performance metrics confusion matrixes, such as sensitivity and specificity, were also employed to measure the performance of different classifiers for the three datasets. As a whole, the RF classifier has the best performance as compared to the other three methods and has obtained higher accuracies 94.4821%, 97.5136%, and 94.6375% on MSM dataset, FSW dataset, and IDU dataset, respectively. The decision tree has also achieved the least classification accuracy 76.7440%, 85.0408%, and 71.2271% on MSM dataset, FSW dataset, and IDU dataset, respectively. The detailed classification outcomes of each model for the three datasets are shown in Tables 9, 10, and 11.

4. Discussion

The AIDS epidemic in Urumqi is still very serious. The increasing number of high-risk groups, such as prostitutes, male sex workers, and floating population, has exacerbated the difficulty of AIDS prevention and treatment. Data mining has been widely used in the field of diagnosis, evaluation, and other medical fields [48]. This study aimed at using four mature data mining algorithms (random forests, support vector machine, k-nearest neighbors, and decision tree) to build identification models for AIDS patients based on the sentinel monitoring data of HIV high-risk populations (MSM, FSWs, and IDUs) in Urumqi and compared the prediction power of the different models. However, considering

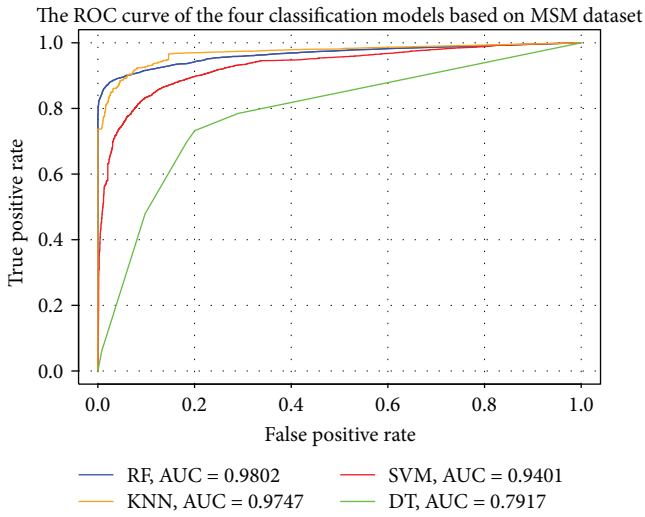


Figure 4: ROC curve of different classifiers for MSM dataset.

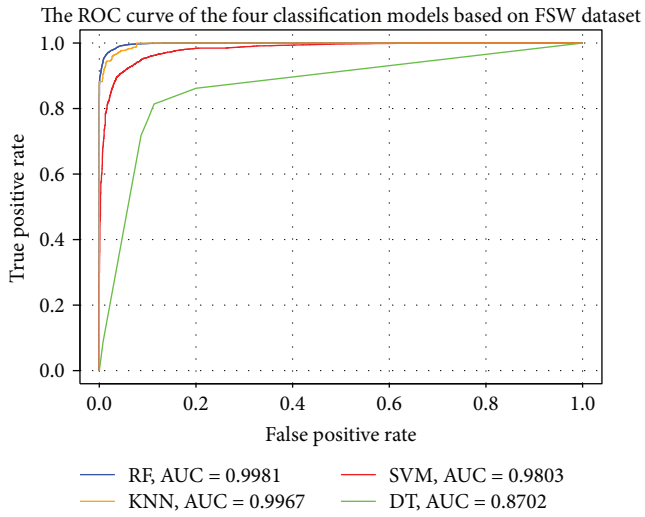


Figure 5: ROC curve of different classifiers for FSW dataset.

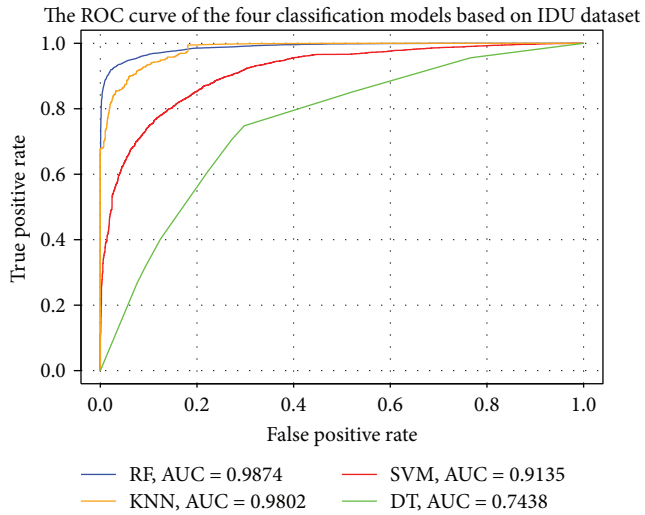


Figure 6: ROC curve of different classifiers for IDU dataset.

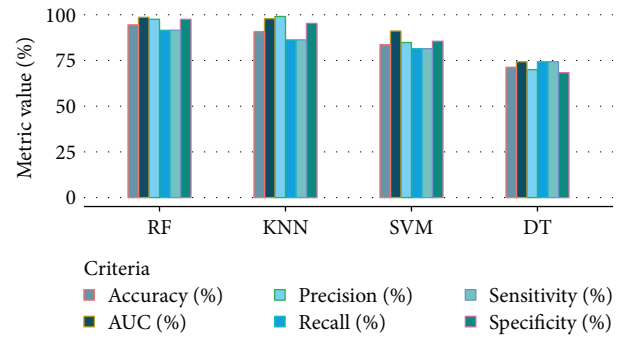


Figure 7: Performance of different classification models for MSM dataset.

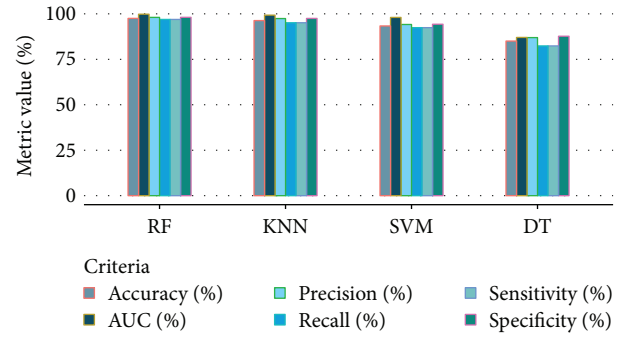


Figure 8: Performance of different classification models for FSW dataset.

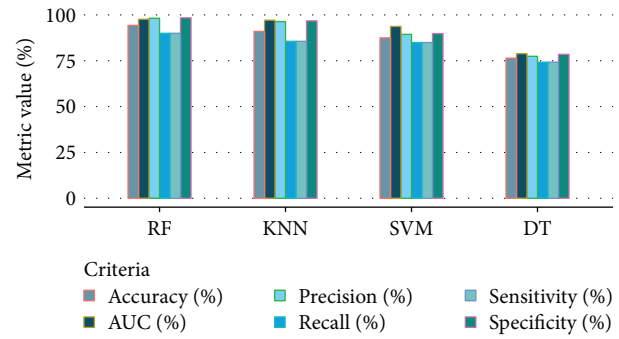


Figure 9: Performance of different classification models for IDU dataset.

that the major defect in the model build process is class imbalances, the SMOTE method has been used to simulate the data balance and overcome the problem of overfitting according to the previous research [49].

For all datasets, the final experimental results showed that RF algorithm obtains the best results; the diagnostic accuracy for RF on MSM dataset are 94.4821%, 97.5136% on FSW dataset, and 94.6375% on IDU dataset. The KNN algorithm came out second, with 91.5258% diagnostic accuracy on MSM dataset, 96.3083% diagnostic accuracy on FSW dataset, and 90.8287% diagnostic accuracy on IDU dataset, followed by SVM (94.0182%, 98.0369%, and

Table 9: Performance measures of the classifiers for MSM dataset.

Testing criteria	RF		SVM		KNN		DT	
Confusion matrix	4911	65	4485	491	4828	148	3921	1055
	480	4421	710	4191	689	4212	1242	3659
Accuracy (%)	94.4821		87.8404		91.5258		76.7440	
Sensitivity (%)	90.2061		85.5132		85.9416		74.6582	
Specificity (%)	98.6937		90.1326		97.0257		78.7982	
Precision (%)	98.5511		89.5130		96.6055		77.6199	
Recall (%)	90.2061		85.5132		85.9416		74.6582	
AUC (%)	98.0217		94.0182		97.4709		79.1761	

Table 10: Performance measures of the classifiers for IDU dataset.

Testing criteria	RF		SVM		KNN		DT	
Confusion matrix	6151	149	5389	911	6003	297	4299	2001
	524	5726	1162	5088	854	5396	1610	4640
Accuracy (%)	94.6375		83.4821		90.8287		71.2271	
Sensitivity (%)	91.6160		81.4080		86.3360		74.2400	
Specificity (%)	97.6349		85.5397		95.2857		68.2381	
Precision (%)	97.4638		84.8141		94.7831		69.8690	
Recall (%)	91.6160		81.4080		86.3360		74.2400	
AUC (%)	98.7495		91.3571		98.0208		74.3879	

Table 11: Performance measures of the classifiers for FSW dataset.

Testing criteria	RF		SVM		KNN		DT	
Confusion matrix	9709	189	9333	565	9650	248	8680	1218
	302	9547	747	9102	481	9368	1736	8113
Accuracy (%)	97.5136		93.3560		96.3083		85.0408	
Sensitivity (%)	96.9337		92.4155		95.1163		82.3739	
Specificity (%)	98.0905		94.2918		97.4944		87.6945	
Precision (%)	98.0588		94.1554		97.4210		86.9467	
Recall (%)	96.9337		92.4155		95.1163		82.3739	
AUC (%)	99.8114		98.0369		99.6712		87.0283	

91.3571%). The DT algorithm was the poorest of the four algorithms, with 79.1761% diagnostic accuracy on MSM dataset, 87.0283% diagnostic accuracy on FSW dataset, and 74.3879% accuracy on IDU. These results suggested that the four established data mining models can predict whether a person is infected with HIV. But compared with SVM, decision tree, and KNN, random forest model through a large number of random sample method balance the sampling error; the effect of classifying the results produces a large number of different test data. A comprehensive assessment is just a single test sample for fitting the results of the other three models more reliably [50].

This study based on the importance score of independent variables for random forest model identified the most important influencing factor for the HIV infection in the three high dangerous populations in Urumqi. For the MSM dataset, these variables are age, educational level, monitoring sites, sample source, inhabit time, nation, marital status, etc. Variables such as age show that the MSM population in Urumqi is mainly the young and middle-aged active population aged from 18 to 40 years old, accounting for 91.3%, which is similar to the monitoring results in Chengdu [51] and show that sexually active people are still the focus of AIDS prevention and treatment. The majority (82.5%) of the participants had never been married. More than half (56.2%) came from the Sayibak District, 68% of the participants were recruited through the network, and 72.1% had some college or higher education. Therefore, based on the epidemic characteristics of MSM population in Urumqi, personal characteristics and social factors should be taken into account comprehensively when education intervention measures are carried out for this population. For the FSW dataset, the results showed that

most of the female sex workers (FSWs) in Urumqi were young women under 30 years old, 58.2% were unmarried, 65% of female sex workers (FSWs) worked in a local workspace for less than a year, and more than half were primary school and junior middle school and had come mainly from nightclub, karaoke, ballroom, and bar. Therefore, we should focus on the actual epidemic characteristics of FSWs to take corresponding measures to publicize education and intervene. For the IDU dataset, the age of the 7337 participants ranged from l1 to 71 years, with more than half (94.5%) of them aged 18–48 years. Among them, 2586 (35.2%) were single, with 2147 (32.9%) participants coming from Sayibak District, and 5169(66.4%) participants were junior high school and below. Among the participants, 89.3% were male and 69% were from the community. These results can provide evidence for the prevention of HIV infection among drug users through the promotion of education, especially for adolescents, low cultural level population, floating population, drug abuse, sexual disorder, etc.

As we have shown above, data mining models can accurately identify diseases based on certain important attributes. These predictive models are valuable tools in the medical field. However, there are areas of concern in the development of predictive models: (1) the model should include all clinically relevant data, (2) the model should be tested on an independent sample, and (3) the model must make sense to the medical personnel who are supposed to make use of it. It has been shown that not all predictive models constructed using data mining techniques satisfy all of these requirements [52].

There are some limitations to this article. First, all individuals are recruited in Urumqi, which was limited by geographical and population characteristics. Therefore, the information bias may exist during the experiment process. If the study population could be expanded to more than one province or to the whole country, the model recognition effect would be better. Second, in the epidemiological investigation of HIV-infected persons, due to subjective, objective, and other reasons, respondents may provide unreal information, which leads to a certain influence on the analysis results. In the future, more feature selection methods, class imbalance processing methods, and data mining algorithms are expected to be tested.

5. Conclusion

In general, four prediction models were established and compared for predicting whether a person is infected with HIV. The results showed that the random forest model performed the best in classification accuracy. This study can provide some effective ways for medical staffs to quickly screen and diagnose AIDS from a large amount of information.

Data Availability

The (CSV) data used to support the findings of this study are restricted in order to protect patient privacy. Data are available from Kai Wang (wangkaimath@sina.com) for researchers who meet the criteria for access to confidential data.

Conflicts of Interest

The authors declare no conflict of interest.

Authors' Contributions

Dandan Tang, Kai Wang, and Yujian Zheng designed the project; Man Zhang, Jiabo Xu, and Xueliang Zhang participated in the data collection; Dandan Tang, Li Feng, and Huling Li performed the analysis of the data; Dandan Tang and Fang Yang wrote the manuscript. All authors contributed to the interpretation of the results, revised the manuscript critically, and approved the final version of the manuscript.

Acknowledgments

This project was supported by the National Natural Science Foundation of China (11461073, 11301451).

References

[1] O. Singh and E. C. Y. Su, "Prediction of HIV-1 protease cleavage site using a combination of sequence, structural, and physicochemical features," *BMC Bioinformatics*, vol. 17, Supplement 17, pp. 478–289, 2016.

[2] M. A. Nowak and A. J. Mcmichael, "How HIV defeats the immune system," *Scientific American*, vol. 273, no. 2, pp. 58–65, 1995.

[3] N. I. Ming-Jian, J. Chen, Y. Zhang et al., "Analysis of epidemic status of HIV/AIDS in Xinjiang," *Bulletin of Disease Control and Prevention*, vol. 27, no. 2, pp. 1–3, 2012.

[4] Q. Zheng, J. Wang, Y. Dong et al., "Analysis of monitoring data of AIDS in Xinjiang from 2004 to 2015," *Bulletin of Disease Control & Prevention*, vol. 32, no. 1, pp. 34–48, 2017.

[5] M. A. Ling, "HIV/AIDS epidemic in Urumqi from 1995 to 2011," *Modern Preventive Medicine*, vol. 109, pp. 2727–2729, 2013.

[6] M. A. Ling and Y. X. Wang, "Characteristics of man who have sex with men HIV/AIDS cases reported through internet based direct reporting system in Urumqi," *World Latest Medicine Information*, vol. 16, no. 52, pp. 1-2, 2016.

[7] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, Google Ebook, 2011.

[8] D. A. Adeniyi, Z. Wei, and Y. Yongquan, "Automated web usage data mining and recommendation system using k-nearest neighbor (knn) classification method," *Applied Computing and Informatics*, vol. 12, no. 1, pp. 90–108, 2016.

[9] H. B. Burke, P. H. Goodman, D. B. Rosen et al., "Artificial neural networks improve the accuracy of cancer survival prediction," *Cancer*, vol. 79, pp. 857–862, 1997.

[10] C. D. Chang, C. C. Wang, and B. C. Jiang, "Using data mining techniques for multi-diseases prediction modeling of hypertension and hyperlipidemia by common risk factors," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5507–5513, 2011.

[11] X. H. Meng, Y. X. Huang, D. P. Rao, Q. Zhang, and Q. Liu, "Comparison of three data mining models for predicting diabetes or prediabetes by risk factors," *The Kaohsiung Journal of Medical Sciences*, vol. 29, no. 2, pp. 93–99, 2013.

[12] L. Wang, "Application of data mining technology in diagnosis and treatment of AIDS," *Journal of Mathematical Medicine*, vol. 26, no. 1, pp. 97–99, 2013.

[13] A. Oliveira, B. M. Faria, A. R. Gaio, and L. P. Reis, "Data mining in HIV-AIDS surveillance system," *Journal of Medical Systems*, vol. 41, no. 4, p. 51, 2017.

[14] D. Wang, B. Larder, A. Revell et al., "A comparison of three computational modelling methods for the prediction of virological response to combination HIV therapy," *Artificial Intelligence in Medicine*, vol. 47, no. 1, pp. 63–74, 2009.

[15] W. U. Hai-Lei, J. S. Qian, and C. Zhang, "A HIV carrier forecasting model for quarantine based on support vector machines," *Practical Preventive Medicine*, vol. 17, no. 11, pp. 2152–2155, 2010.

[16] T. G. Hailu, "Comparing data mining techniques in HIV testing prediction," *Intelligent Information Management*, vol. 07, no. 3, pp. 153–180, 2015.

[17] A. Famili, W. Shen, R. Weber, and E. Simoudis, "Data preprocessing and intelligent data analysis," *Intelligent Data Analysis*, vol. 1, no. 1-4, pp. 3–23, 1997.

[18] R. Blagus and L. Lusa, "SMOTE for high-dimensional class-imbalanced data," *BMC Bioinformatics*, vol. 14, no. 1, pp. 106–116, 2013.

[19] L. Zhang, C. Zhang, R. Gao, R. Yang, and Q. Song, "Using the SMOTE technique and hybrid features to predict the types of ion channel-targeted conotoxins," *Journal of Theoretical Biology*, vol. 403, pp. 75–84, 2016.

[20] E. M. Karabulut and T. Ibrikci, "Effective automated prediction of vertebral column pathologies based on logistic model tree with smote preprocessing," *Journal of Medical Systems*, vol. 38, no. 5, p. 50, 2014.

[21] H. Liu, X. Shi, D. Guo, Z. Zhao, and Yimin, "Feature selection combined with neural network structure optimization for HIV-1 protease cleavage site prediction," *BioMed Research International*, vol. 2015, Article ID 263586, 11 pages, 2015.

[22] J. R. Bienkowska, G. S. Dalgin, F. Batliwalla et al., "Convergent random forest predictor: methodology for predicting drug response from genome-scale data applied to anti-TNF response," *Genomics*, vol. 94, no. 6, pp. 423–432, 2009.

[23] X. Chen and H. Ishwaran, "Random forests for genomic data analysis," *Genomics*, vol. 99, no. 6, pp. 323–329, 2012.

[24] M. Kotti, L. D. Duffell, A. A. Faisal, and A. H. Mcgregor, "Detecting knee osteoarthritis and its discriminating parameters using random forests," *Medical Engineering & Physics*, vol. 43, pp. 19–29, 2017.

[25] A. Hapfelmeier and K. Ulm, "A new variable selection approach using random forests," *Computational Statistics & Data Analysis*, vol. 60, pp. 50–69, 2013.

[26] M. Sandri and P. Zuccolotto, "Variable selection using random forests," in *Data Analysis, Classification and the Forward Search*, pp. 263–270, 2006.

[27] B. H. Menze, B. M. Kelm, R. Masuch et al., "A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data," *BMC Bioinformatics*, vol. 10, no. 1, pp. 213–216, 2009.

[28] A.-L. Boulesteix, A. Bender, J. Lorenzo Bermejo, and C. Strobl, "Random forest Gini importance favours SNPs with large minor allele frequency: impact, sources and recommendations," *Briefings in Bioinformatics*, vol. 13, no. 3, pp. 292–304, 2012.

[29] T. K. Ho, "Random decision forests," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, p. 278, Montreal, Quebec, Canada, August 1995.

[30] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[31] M. Dauwan, J. J. van der Zande, E. van Dellen et al., "Random forest to differentiate dementia with Lewy bodies from Alzheimer's disease," *Alzheimer's & Dementia: Diagnosis, Assessment & Disease Monitoring*, vol. 4, pp. 99–106, 2016.

[32] A. T. Azar, H. I. Elshazly, A. E. Hassanien, and A. M. Elkorany, "A random forest classifier for lymph diseases," *Computer Methods and Programs in Biomedicine*, vol. 113, no. 2, pp. 465–473, 2014.

[33] T. Shaikhina, D. Lowe, S. Daga, D. Briggs, R. Higgins, and N. Khovanova, "Decision tree and random forest models for outcome prediction in antibody incompatible kidney transplantation," *Biomedical Signal Processing and Control*, vol. 37, pp. 1025–1042, 2017.

[34] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland, "Data mining for credit card fraud: a comparative study," *Decision Support Systems*, vol. 50, no. 3, pp. 602–613, 2011.

[35] Y. Tian, X. Ju, Z. Qi, and Y. Shi, "Efficient sparse least squares support vector machines for pattern classification," *Computers & Mathematics with Applications*, vol. 66, no. 10, pp. 1935–1947, 2013.

[36] C. S. Lo and C. M. Wang, "Support vector machine for breast MR image classification," *Computers & Mathematics with Applications*, vol. 64, no. 5, pp. 1153–1162, 2012.

[37] Y. Guo, L. Yu, Z. Wen, and M. Li, "Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences," *Nucleic Acids Research*, vol. 36, no. 9, pp. 3025–3030, 2008.

[38] H. Yang, L. Chan, and I. King, "Support vector machine regression for volatile stock market prediction," in *Intelligent Data Engineering and Automated Learning — IDEAL 2002*, vol. 2412, pp. 391–396, 2002.

[39] C. K. I. Williams, "Learning with kernels: support vector machines, regularization, optimization, and beyond," *Publications of the American Statistical Association*, vol. 98, pp. 489–489, 2002.

[40] V. P. Gladis Pushpa Rathi, "A novel approach for feature extraction and selection on MRI images for brain tumor classification," *International Conference on Computer Science, Engineering and Applications.*, vol. 10, no. 5, pp. 225–234, 2012.

[41] M. Akhil Jabbar, B. L. Deekshatulu, and P. Chandra, "Classification of heart disease using k-nearest neighbor and genetic algorithm," *Procedia Technology*, vol. 10, pp. 85–94, 2013.

[42] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[43] E. A. Aydın and M. K. Keleş, "Breast cancer detection using k-nearest neighbors data mining method obtained from the bow-tie antenna dataset," *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 27, no. 6, 2017.

[44] F. I. Alam, F. K. Bappee, M. R. Rabbani, and M. M. Islam, "An optimized formulation of decision tree classifier," *Communications in Computer and Information Science*, vol. 361, pp. 105–118, 2013.

[45] J. R. Neto, Z. M. de Souza, S. R. de Medeiros Oliveira et al., "Use of the decision tree technique to estimate sugarcane productivity under edaphoclimatic conditions," *Sugar Tech.*, vol. 19, no. 6, pp. 662–668, 2017.

[46] K. Boonchuay, K. Sinapiromsaran, and C. Lursinsap, "Decision tree induction based on minority entropy for the class imbalance problem," *Pattern Analysis and Applications*, vol. 20, no. 3, pp. 769–782, 2017.

[47] T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.

[48] Y. U. Chang-Chun, H. E. Jia, S. C. Fan et al., "Application of data mining in medical field," *Academic Journal of Second Military Medical University*, vol. 24, pp. 1250–1252, 2003.

[49] D. M. Herrera-Ibatá, A. Pazos, R. A. Orbegozo-Medina, F. J. Romero-Durán, and H. González-Díaz, "Mapping chemical structure-activity information of HAART-drug cocktails over complex networks of AIDS epidemiology and socioeconomic data of U.S. counties," *Bio Systems*, vol. 132-133, pp. 20–34, 2015.

[50] T. A. Almeida, R. M. Silva, and A. Yamakami, "Machine learning methods for spamdexing detection," *International Journal of Information Security Science*, vol. 2, pp. 1–22, 2016.

[51] Y. Feng, Z. Wu, R. Detels et al., "HIV/STD prevalence among men who have sex with men in Chengdu, China and associated risk factors for HIV Infection," *Journal of Acquired Immune Deficiency Syndromes*, vol. 53, Supplement 1, pp. S74–S80, 2010.

[52] D. Delen, G. Walker, and A. Kadam, "Predicting breast cancer survivability: a comparison of three data mining methods," *Artificial Intelligence in Medicine*, vol. 34, no. 2, pp. 113–127, 2005.

Hindawi
Complexity
Volume 2018, Article ID 8651930, 13 pages
https://doi.org/10.1155/2018/8651930

Research Article

Tensor Decomposition for Multiple-Instance Classification of High-Order Medical Data

Thomas Papastergiou, Evangelia I. Zacharaki, and Vasileios Megalooikonomou

Computer Engineering and Informatics Department, University of Patras, Rio, Achaia 26504, Greece

Correspondence should be addressed to Thomas Papastergiou; papastergiou@ceid.upatras.gr

Received 1 June 2018; Accepted 30 August 2018; Published 6 December 2018

Academic Editor: Panayiotis Vlamos

Multidimensional data that occur in a variety of applications in clinical diagnostics and health care can naturally be represented by multidimensional arrays (i.e., tensors). Tensor decompositions offer valuable and powerful tools for latent concept discovery that can handle effectively missing values and noise. We propose a seamless, application-independent feature extraction and multiple-instance (MI) classification method, which represents the raw multidimensional, possibly incomplete, data by means of learning a high-order dictionary. The effectiveness of the proposed method is demonstrated in two application scenarios: (i) prediction of frailty in older people using multisensor recordings and (ii) breast cancer classification based on histopathology images. The proposed method outperforms or is comparable to the state-of-the-art multiple-instance learning classifiers highlighting its potential for computer-assisted diagnosis and health care support.

1. Introduction

Nowadays, data tend to be large in volume and multiparametric in nature, especially in clinical diagnostics and health care. Applications that provide massive multidimensional data are vast. Some examples include monitoring patients by multisensor technologies [1, 2], noninvasive lesion detection and diagnosis using hyperspectral sampling [3], cancer diagnosis based on tissue microarray data [4, 5], color segmentation of skin lesions using histology-stained microscopic images [4, 5], classification of EEG signals for seizure detection [6], or for Alzheimer's disease analysis [7]. The main challenge is to extract discriminative features from high-dimensional data in a way that preserves their multidimensional structure while at the same time models the interdimensions' interaction. Traditional matrix representation techniques that represent high-dimensional data by flattening them to a matrix suffer many times from the curse of dimensionality that poses limitations on many two-dimensional approaches. By representing such data in a more natural way by multidimensional arrays (a.k.a. tensors) and using sophisticated high-order techniques, such as tensor decompositions, we can capture multiple interactions and couplings and simultaneously discover latent concepts that are present in the data [8]. Tensor-based techniques have been employed in the field of signal processing and machine learning for a variety of tasks [9] like in blind multiuser code-division multiple access (CDMA) communications, blind source separation, collaborative filtering-based recommender systems, Gaussian mixture parameter estimation, topic modeling, or, as mostly related to our work, multilinear discriminative subspace learning [10, 11], among many others. For an extensive overview of the underlying tensor theory and the aforementioned applications, we refer to the extensive review paper [9]. Tensor decomposition has also been applied recently for image restoration by grouping image patches [12] or for image compression and reconstruction [13, 14] by removing redundancy simultaneously in spatial and spectral domain. In contrast to multichannel signal or image data encoding that often benefits from tensor decomposition due to their structured nature, encoding of 3D geometrical meshes rather relies on traditional techniques, such as graph Fourier Transform [15]. A common aspect in most of the applications is the exploitation of sparsity in high-order structures. An overview

of some basic techniques that exploit sparsity in the recovery of low-rank higher-order tensors, followed by related applications, is provided in [16].

The second challenge in the analysis of current biomedical data comes in the learning phase that follows the data representation phase. Standard supervised learning implies that each example used for training a classification model, is represented as a feature vector with an associate class label attached. However, in many real-life applications, data tend to be complex, incorporating different concepts, and thus it is difficult to model each example as a single feature vector: e.g., medical images depicting different tissue types, biosignals tracking different activities, or molecules with conformations with different chemical properties. In these cases, a more efficient representation, which preserves as much information as possible, consists of a collection of feature vectors (denoted as instances), such as patches of an image, time windows of biosignals, or conformations of a molecule, each one covering a different aspect of the whole object. The challenge that arises for such representations is the lack of refined annotation for each individual feature vector, known as *multiple-instance learning* (MIL). Furthermore, some of the feature vectors describing an observation could provide none or sometimes even misleading information about the object's class (e.g., not all cells are malignant in a histopathology image with malignancy).

Besides the challenges inherited by the high-order structure and multivariate context, data partiality or incompleteness impose an additional burden. Missing data occur in real-life due to a variety of reasons including failure in the data acquisition processes (e.g., temporary malfunction of EEG electrodes [17]), costly experiments impeding the annotation of all samples, or due to noise or artifacts removal. In supervised learning paradigms, missing values must be removed from the data or imputed by statistical approaches [18] prior to inference. Another interesting approach when classifying data with missing values is based on the assumption that data are of low rank [19, 20], that there exist prototypes (i.e., components) and all the samples can be reconstructed by a mixture of them. For example in [19], the classification problem is treated as a matrix completion problem via rank minimization, while in [20], classification is performed using the low-rank assumption without any matrix completion step. For high-dimensional settings, dissimilarity-based classification is proposed in [21] where missing values are estimated via high-order decomposition and then classification is performed on the completed data.

The aim of this work is to define a generalized tensor-based multiple-instance learning framework (called TensMIL) for analyzing high-order, possibly incomplete data, avoiding the extraction of predefined or hand-crafted features. Our approach is formulated as a multistep minimization problem in which all parameters, internal and external, are learnt by supervision. In order to illustrate the wide applicability of TensMIL, we assess it in two distinct scenarios for multiple-instance classification using biomedical images and multichannel biosignals, respectively, and compare it against other state-of-the-art techniques. In order to place the method into the MIL context and better appreciate its differences from other approaches, we first provide a small overview of the related work in MIL and then proceed with more details and contributions of TensMIL.

In multiple-instance learning problems, bags (subjects) are described by multiple-feature arrays (instances) and labels are provided only for the bags, whereas the labels of the individual instances are unknown. Several methods have been proposed exploiting local or global information and implementing different classifiers or mapping functions. For a complete taxonomy on MIL algorithms, we refer to the work of Amores [22], as well as previous reviews by Foulds and Frank [23] or Dong [24]. At the first level of the taxonomy tree, the classification frameworks follow either the Instance Space (IS), Bag Space (BS), or Embedded Space (ES) paradigm.

The inference process for the methods in the IS paradigm is based on information that resides in the individual instances, i.e., an instance-level classifier is trained to separate the instances in positive or negative class. The obtained instance-level scores are then aggregated to summarize the information about the whole bag, usually based on one of the two assumptions [22, 23]: the *standard MI* assumption that states that every positive bag contains *at least one* positive instance and the *collective* (or *weighted collective*) assumption in which all instances in a bag contribute equally (or according to weights) to the bag's label [25]. The selected aggregation rule thus acts as a bag-level classifier. Although the assumption-based IS paradigm proves to be an effective heuristic in many application domains, very often, the relationship between instances in a bag and the bag-level class labels is unknown; therefore, the use of *concepts* was introduced to relax the strict view of predefined assumptions. A more refined hierarchy of assumptions was defined by Weidmann et al. [26] and presented by increasing generality from the standard MI (for a single concept), to the presence-based (for multiple concepts), threshold-based, and count-based assumption.

In contrast to the IS paradigm, where the (bag-level) classifier is obtained as an aggregation of local responses, the inference process of the methods in the BS and ES paradigms is performed in the space of bags. BS methods directly employ a distance or kernel function that operates on non-vectorial entities, such as the bags, in order to assess similarity between them. Since our proposed method relates less to this category of methods, we omit further discussion, but refer to [22] for additional details. In the ES paradigm, a set of concepts are identified by unsupervised learning and used as a vocabulary that describes classes of instances. A mapping function is then employed to map each bag into a feature vector $\mathbf{v}$ which aggregates the pertinent information about the bag. In the special case of histogram-based ES methods, the vector $\mathbf{v}$ describes the distribution (histogram) of the instances into the different classes of the vocabulary. The few ES methods that are not based on vocabularies or concepts' learning usually summarize statistics (for example, the minimum and maximum values) of the features of all the instances inside the bag. Another interesting approach is associating the bags with their most informative instances via instance selection. In this way, the bag space is mapped

to a reduced instance space, where IS classifiers or even classic non-MIL classifiers can be exploited. Recently a new multiple-instance learning algorithm with discriminative bag mapping (MILDM) [27] has been proposed, where informative instances are selected such that the bags are maximally distinguishable in the new mapping space.

In this paper, we propose a seamless method for feature extraction and MIL classification of high-dimensional data by modeling the data as n-dimensional arrays (i.e., tensors). Through tensor decomposition, we construct a high-dimensional dictionary that models the latent factors of the data as a number of $n-1$ dimensional rank-1 constructs. In this way, the coefficients that correspond to the instances' mode indicate the contribution of each latent factor to the representation of the corresponding instance, and thus they can serve as instance-level features. Subsequently, using these features, we train an instance-level classifier for predicting the hidden class label of each instance by a continuous score. We model each bag by the density function of the predicted labels and train a bag space classifier for the final classification task. Our motivation was to avoid strict predefined MIL rules, such as the standard MIL assumption; therefore, we extended the collective assumption, by learning the bag labels using the probability density function of the estimated hidden instances' labels.

The main contributions of our work are summarized as follows:

(1) TensMIL is based on a generalized feature extraction method for high-dimensional data using tensor decomposition, thus can be applied in multiple scenarios

(2) It performs well even with a very small number (e.g., 10%) of observed data

(3) Evaluation in the UCSB Breast Cancer benchmark dataset with full and with partial observed values showed that it outperforms or is comparable to existing state-of-the-art MIL algorithms

(4) To the best of our knowledge, we are the first to exploit the potential of physiological (such as respiratory and cardiac) signals in predicting aging-associated decline (frailty). The application of TensMIL revealed prognostic capabilities for frailty manifestation that previous methods failed to uncover

2. Materials and Methods

The proposed methodology is illustrated in the simplified schematic diagram in Figure 1 and consists mainly of three phases: (i) the data representation and feature extraction phase in which the data are mapped from the original high-dimensional space to a lower dimensional space using tensor decomposition, (ii) the multiple-instance learning phase in which sequential discriminative models are inferred to classify the data into different groups, and (iii) the optimization phase that is coupled with the previous phase for learning the hyperparameters. In the next sections, we describe

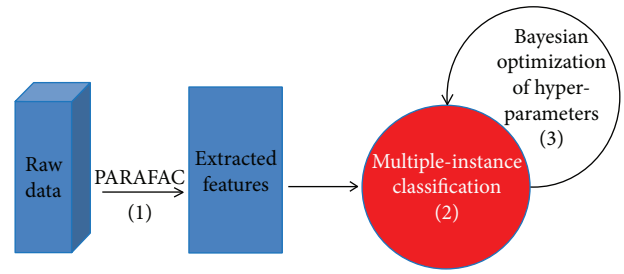


Figure 1: Schematic diagram of the proposed methodology.

analytically every phase starting from the use of tensor decomposition for feature extraction and proceeding with our proposed MIL framework.

The notation that we follow within this manuscript is as follows. We denote tensors by capital boldface Euler letters ($\boldsymbol{\mathcal{X}}$, $\boldsymbol{\mathcal{Y}}$, $\boldsymbol{\mathcal{Z}}$), matrices by capital boldface letters ($\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$), vectors by boldface lowercase letters ($\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{c}$), and scalars by lowercase letters (a, b, c). Entries of a matrix or a tensor are denoted by lowercase letters with subscripts (e.g., the $(i_1, i_2, \ldots, i_n)$ entry of an n-way tensor $\boldsymbol{\mathcal{X}}$ is denoted by $x_{i_1,i_2,\ldots,i_n}$). Columns of a matrix are denoted by a boldface capital letter with a subscript consisting of a star and a number (e.g., $\mathbf{A}_{*,1}$ denotes the first column of matrix $\mathbf{A}$).

2.1. Tensor Decomposition. We briefly outline the CANDECOMP/PARAFAC (CP) decomposition, a powerful tool originally introduced in [28, 29]. For preliminaries on tensors, we refer to the Supplementary Material (available here). Without loss of generality and for the sake of simplicity from now and on, we will refer to 3rd order tensors, although the proposed method can be generalized for high-order tensors. Let $\boldsymbol{\mathcal{X}}$ be a 3-way tensor of size $I \times J \times K$. With full data, a tensor $\boldsymbol{\mathcal{X}}$ can be decomposed into a set of matrices $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}$ of sizes $I \times R$, $J \times R$, $K \times R$, respectively, as follows:

$$\boldsymbol{\mathcal{X}} \approx \sum_{r=1}^{R} \mathbf{U}_{*,r} \circ \mathbf{V}_{*,r} \circ \mathbf{W}_{*,r}, \tag{1}$$

where R is the rank of the decomposition and "$\circ$" denotes the outer product of two arrays.

Let Ω be the set of the observed indices of tensor $\boldsymbol{\mathcal{X}}$. We can define an indicator tensor $\boldsymbol{\mathcal{W}}$ having the same size as the original tensor such that $\boldsymbol{\mathcal{W}}(i, j, k) = 1$, $\forall (i, j, k) \in \Omega$, and zero elsewhere. The tensor decomposition problem can then be formulated as follows:

$$\min_{\mathbf{U},\mathbf{V},\mathbf{W}} \left\| \boldsymbol{\mathcal{W}} \circledast \left(\boldsymbol{\mathcal{X}} - \sum_{r=1}^{R} \mathbf{U}_{*,r} \circ \mathbf{V}_{*,r} \circ \mathbf{W}_{*,r} \right) \right\|_{\mathrm{F}}^{2}, \tag{2}$$

where the "$\circledast$" denotes the Hadamard (element-wise) product. When Ω is equal to the set of indices of $\boldsymbol{\mathcal{X}}$, then we have a full- (nonmissing) value decomposition problem; otherwise, we have a decomposition problem with missing values.

For calculating the CP decomposition, we exploit the well-known Alternating Least Squares (ALS) method [30] when we deal with a full-value problem, and the two Proximal methods proposed in [31] when we deal with missing value problems. The methods proposed in [31]—GenProxSGD (nondistributed) and StrProxSGD (distributed, suitable for big data)—tackle the optimization problem in (2) by solving local minimization problems rather than solving the entire problem at once.

2.2. Generalized Feature Extraction. We propose here a general method for extracting instance-based features from raw data in which data are represented as an n-dimensional tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}$. The representation of the data is problem-specific, and we will discuss in a later section the representation of data for the two different problems that we tackle. Our objective is to calculate the latent factors of data via the CP decomposition of the raw data tensor, where instances are arranged in one dimension. The obtained factor matrix (the one corresponding to the instances) can be used as feature matrix in the instances' space. The other factor matrices correspond to the calculated high-order dictionary.

Formally, if $\mathcal{X} \in R^{I \times J \times K}$(instances are arranged across the first dimension), we can write slice-wise a rank-R CP decomposition of $\mathcal{X}$ presented in (1) as

$$\mathcal{X}_{i,*,*} \approx \sum_{r=1}^{R} u_{ir}(\boldsymbol{V}_{*r} \circ \boldsymbol{W}_{*r}), \tag{3}$$

where $\mathcal{X}_{i,*,*}$ represents a mode-1 slice of the tensor that corresponds to the ith instance. Equation (3) denotes that each instance can be approximated as a linear combination of R two-dimensional components, $\boldsymbol{V}_{*,r} \circ \boldsymbol{W}_{*,r} \in \mathbb{R}^{J \times K}$ which correspond to the latent factors of the data. Thus, we can choose as features representing an instance i, the R coefficients u_{ir}, $r = 1, 2, \ldots, R$, that correspond to the ith row of factor matrix $\boldsymbol{U}$ in (2). Furthermore, we can see the latent factors $\boldsymbol{V}_{*,r} \circ \boldsymbol{W}_{*,r}$ as a high-order dictionary describing the data. This procedure can be employed as is to tensors of order $N > 3$ yielding dictionaries of order $N - 1$ and is independent of the nature of the data per se.

2.3. Alternative Feature Extraction for New (Unseen) Data. The tensor-based feature extraction process in the proposed framework involves the decomposition of a common tensor constructed by the concatenation of training and testing samples, as described above. For reducing the computational cost, it might be desired to classify new testing data without repeating the whole tensor decomposition. We describe next an alternative approach to obtaining the low-dimensional feature representation in which a PARAFAC model is constructed only from the training data while the test data are represented by the estimated training model as follows. If $\mathcal{X}_{\text{train}} \approx \sum_{r=1}^{R} \mathbf{U}_{*,r} \circ \mathbf{V}_{*,r} \circ \mathbf{W}_{*,r}$ is a PARAFAC decomposition of rank R calculated for the training set and $\mathcal{X}_{\text{test}}$ is the tensor of test data, then it can be shown [30] that the PARAFAC calculation problem of (2) can be written in a mode-1 matricized form as

$$\min_{U,V,W} \left\| \mathbf{X}_{\text{train}(1)} - \mathbf{U}(\mathbf{W} \odot \mathbf{V})^{\mathbf{T}} \right\|_{\text{Fr}}^{2}. \tag{4}$$

We can formulate and solve a least squares minimization problem to find the "closest" representation of the test set based on the calculated dictionary of $\mathbf{U}$ and $\mathbf{W}$:

$$\min_{\tilde{U}} \left\| \mathbf{X}_{\text{test}(1)} - \tilde{\mathbf{U}}(\mathbf{W} \odot \mathbf{V})^{\mathbf{T}} \right\|_{\text{Fr}}^{2}. \tag{5}$$

It is easy to show [30] that the solution of the problem of (5) has the following closed form $\tilde{\boldsymbol{U}} = \boldsymbol{X}_{test(1)}(\boldsymbol{W} \odot \boldsymbol{V})(\boldsymbol{W}^{\boldsymbol{T}}\boldsymbol{W} \circledast \boldsymbol{V}^{\boldsymbol{T}}\boldsymbol{V})^{\dagger}$, where "†" is the Moore-Penrose pseudoinverse.

In the following, we describe the next phase of the methodology that involves the construction of the discriminative model by multiple-instance learning.

2.4. Problem Statement in Multiple-Instance Learning (MIL). We first briefly define formally the multiple-instance learning problem. A bag $B_i = \{\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}, \ldots, \boldsymbol{x}_{i,n_i}\}$ is a set of n_i feature vectors describing a subject. Let us denote $\mathbf{B} = \{B_i, i = 1, 2, \ldots, n\}$ as the set of all the bags. The cardinality of each bag B_i can vary across the bags. Each feature vector $\boldsymbol{x}_{i,j}$, where the first index refers to the corresponding bag and the second index to the feature of the bag it belongs to, is called an *instance*. All instances $\boldsymbol{x}_{i,j}$, $i = 1, 2, \ldots, n$, and $j = 1, 2, \ldots, n_i$ live in a d-dimensional feature space ($\boldsymbol{x}_{i,j} \in \mathbb{R}^d$), called *instance space*. Each bag comes with a label attached to it $Y_i \in \mathcal{Y} = \{1, \ldots, C\}$, $i = 1, 2, \ldots, n$, with $C = 2$ defining a binary classification problem and $C > 2$ defining a C class classification problem. $\mathcal{Y}$ denotes the set of all bag class labels.

The objective of a MIL problem is given a collection of n bags (subjects) with their appropriate labels $\{(B_i, Y_i), i = 1, 2, \ldots, n\}$ to learn a model that can predict the labels of new observations (bags).

2.5. Our MIL Framework (TensMIL). The proposed MIL framework follows the IS paradigm in which an instance-level classifier $f(\boldsymbol{x})$ is first constructed based on the label inheritance rule (i.e., all instances of a bag inherit the label of the bag). In order to make learning computationally feasible, it is generally necessary to reduce the hypothesis space by enforcing some MI assumption. However, in contrast to the classical IS-based methods that directly combine the instance-level responses through some predefined rule, we increase the generality and try to infer those assumptions based on the training set. Specifically, we extract the histogram of all instance-level responses within each bag and learn the distribution of those histograms from the training set. The instance-label responses refer to the output of the instance-level classifier $f(\mathbf{x})$ and are analogous to class prediction scores for each instance. The histogram extraction of the instance-level

responses corresponds to quantizing the responses within predefined bins that can be considered as clusters of low, medium, or high class-likeness. In that sense, our framework relates also to the *ES methods without vocabularies* with the difference that the representation is not based on the original (multiple) attributes of the instances, but on the instance-level responses (output of the first classifier). Our contribution lies in the fact that we do not rely on a few statistics, like the average, minimum, or maximum values, but incorporate a richer representation such as the histogram.

In mathematical terms, we formulate (similarly to previous work [32]) an optimization problem that we solve based on the following steps:

(i) First, the instance-level responses within each bag are estimated based on a function $f(\bullet|\boldsymbol{\theta}_f)$ that assigns a class prediction score (such as an abnormality score) to each instance in the bag given a set of parameters $\boldsymbol{\theta}_f$ (6), by initializing the unknown instance labels with the corresponding class label, i.e., $y_{i,j} = Y_i, \forall_i$:

$$\widehat{\boldsymbol{\theta}}_f = \arg\ \min_{\theta_f} \sum_{i=1}^{n} \sum_{j=1}^{n_i} l\left(f\left(\boldsymbol{x}_{i,j}|\boldsymbol{\theta}_f\right), y_{i,j}\right), \quad (6)$$

where $l : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$ is a loss function defined over the instance space. Upon estimation of $\widehat{\boldsymbol{\theta}}_f$, the function f will provide the predictions for the instance-level class labels, which is in contrast to the work in [32], where the unknown instance-level class labels are considered as optimization variables and are calculated in an iterative manner

(ii) Then, a mapping function $\mathscr{H}(\bullet|\theta_H)$ is applied from the instance space to the bag space and the mapped features are used as the new bag representation $\tilde{B}$ (7). In the proposed method, this mapping corresponds to the calculation of the density function of the class prediction scores and is obtained by histogram extraction:

$$\tilde{B}_i = \left\{\mathscr{H}\left(f\left(\mathbf{x}_{i,j}\middle|\widehat{\boldsymbol{\theta}}_f\right)|\theta_H\right) : \mathbf{x}_{i,j} \in B_i\right\}. \quad (7)$$

(iii) Finally, the classification function $F(\bullet|\boldsymbol{\theta}_F)$ for the whole bag is calculated by supervised learning as shown in the following equations:

$$\widehat{\boldsymbol{\theta}}_F = \arg\ \min_{\boldsymbol{\theta}_F} \sum_{i=1}^{n} L\left(F\left(\tilde{B}_i|\boldsymbol{\theta}_F\right), \mathrm{Y}_i\right), \quad (8)$$

$$\widehat{\mathrm{Y}}_i = F\left(\tilde{B}_i\middle|\widehat{\theta}_F\right), \quad (9)$$

where $L : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$ is a loss function defined over the bag space

More details on the individual steps are provided in the following sections.

2.5.1. Robust Estimation of the Instances' Hidden Labels. The medical applications usually concern classification problems of ordinal data, where the classes have a natural order, such as the grade of a tumor or the performance score in a clinical test. If class labels are used, they can be considered as a discrete approximation of the continuous score (e.g., malignancy); thus, the same techniques can be applied for discrete or continuous output variables. The binary classification is a special case of this problem, where the two classes lie on the two extremes (minimum and maximum) of the clinical score range.

In the first step (6), we use the squared error as loss function and train a full quadratic regression model (containing an intercept, linear terms, interactions, and squared terms) $f : \mathbb{R}^d \to \mathbb{R}$ in the instance space that predicts the hidden class labels $y_{i,j}$ for each instance. The quadratic regression model can be expressed as

$$f(\boldsymbol{x}) = \sum_{k=1}^{d} \sum_{m=1}^{d} \theta_{km} x_k x_m + \sum_{k=1}^{d} x_k, \quad (10)$$

where the parameters θ_{km} collectively form the vector $\boldsymbol{\theta}_f$ in (6), and d is the dimensionality of $\boldsymbol{x}$ employed in the regression. Since there is no available information about the instances' hidden class labels, the regression model is trained by using values for the dependent variable, the class labels of the corresponding bags, this means that $y_{i,j} = Y_i, \forall j = 1, \ldots, n_i$. Upon the calculation of f, which is common for all bags, we can estimate the instance labels as $\widehat{y}_{i,j} = f(\boldsymbol{x}_{i,j})$.

Since not all the instances of a bag i will belong to the bag's class Y_i, some of the instances will behave as outliers and will not fit well to the respective class. To eliminate the effect of such inconsistent data, we employ robust quadratic regression which uses iteratively reweighted least squares with a weighting function [33]. We used the logistic weighting function:

$$w_k = \frac{\tan h(r_k)}{r_k}, \quad r_k = \frac{\mathrm{resid}_k}{\left(\mathrm{tune}^* s^* \sqrt{1 - h_k}\right)} k = 1, 2, \ldots, d, \quad (11)$$

where **resid** is the vector of residuals of the previous iteration, s is an estimate of the standard deviation of the error term given by the median absolute deviation of the residuals from their mean scaled by a constant z, $\boldsymbol{h}$ is the vector of the leverage values from least-squared fit, and *tune* is a tuning parameter. For the experiments of this paper, we used the default values for the aforementioned parameters: $z = 0.6745$ and $tune = 1.205$. The choice of the constant z makes the estimate of the standard deviation of the error term unbiased for normal distributions. Furthermore, the choice of the above default values gives coefficient estimates that are approximately 95% as statistically efficient as the

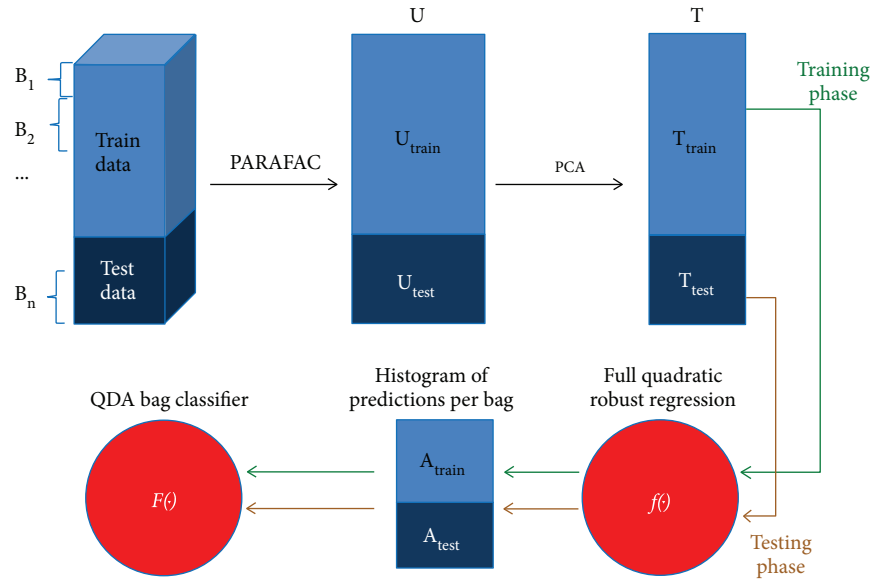


Figure 2: The architecture of TensMIL, where U is the feature matrix extracted from the raw data by PARAFAC decomposition, T is the score matrix obtained by performing PCA on U, A is the matrix containing the bag-level features, $f(\cdot)$ is the full quadratic regression model, and $F(\cdot)$ is the QDA classifier.

ordinary least squares estimates, provided that the response has a normal distribution with no outliers. By employing the above weighting function, the misclassification penalty for the instances that do not belong to the bag's class is reduced, obtaining thus a robust estimation of the hidden labels of the instances. Finally, we want to mention that we experimented with different weighting functions and different tuning parameters and we empirically concluded to use the aforementioned logistic weighting function with the default tuning settings since it yielded better results.

2.5.2. QDA-Based Bag Classification. In order to obtain the bag representation (7) and subsequent bag classification ((8) and (9)), we treat the extracted attributes in target bags (i.e., the instance-level class predictions per bag) as random variables that are defined over a space of probability distributions. We then approximate the density functions $\mathscr{H}_i(\{f(\boldsymbol{x}_{i,j}), j = 1, 2, \ldots, n_i\}), i = 1, 2, \ldots, n$, of the class label scores for each bag by histogram extraction using θ_H equally sized bins. Having estimated the histograms for all bags in the training set $\mathbf{H} = \{\mathscr{H}_i, i = 1, 2, \ldots, n\}$, we can train a bag-wise classifier that will learn to discriminate the unknown class Y. Assuming that the observations from each class $k, k = 1, 2, \ldots, C$ are drawn from a multivariate Gaussian distribution $\mathscr{H} \sim N(\mu_k, \boldsymbol{\Sigma}_k)$ and that each class has its own covariance matrix $(\boldsymbol{\Sigma}_k)$, we can use the quadratic discriminant analysis (QDA) classifier [34] to find a nonlinear quadratic decision boundary. The QDA classifier $F : \tilde{\boldsymbol{B}} \to \mathscr{Y}$ assigns an observation to the class with the maximum discriminant score $\hat{Y}_i = \text{argmax}_k\, \delta_k(h_i)$:

$$\delta_k(p) = -\frac{1}{2}(p - \mu_k)^T \boldsymbol{\Sigma}_k^{\dagger}(p - \mu_k) + \log \pi_k - \frac{1}{2} \log |\boldsymbol{\Sigma}_k|, \quad (12)$$

where δ_k is the discriminant function over the bag space, μ_k is the mean vector of all the training observations from the kth class, $\boldsymbol{\Sigma}_k$ is the covariance matrix for the kth class, and π_k is the prior probability of an observation belonging to the k th class. The parameters $(\mu_k, \boldsymbol{\Sigma}_k)$ of the discriminant functions are learnt from the training set and subsequently used in the testing phase to predict the class labels for new bags.

2.6. Implementation Details and Summary of TensMIL Architecture. In this section, we summarize the individual steps of the method, starting from the raw multidimensional data, and illustrate them in Figure 2 highlighting the differences between training and testing phase.

In the first phase, data must be arranged in a tensor of order $N \geq 3$, with the first dimension dedicated to the instances. The tensor can be constructed by placing instances of each bag $B_1, B_2, \ldots, B_n$ in a sequential order, but this is only for convenience. Training and test data can be placed in the same tensor, constructing a high-dimensional tensor as can be seen in Figure 2. In the second phase (the feature extraction phase), a PARAFAC model is computed and the train and test features are extracted from the corresponding rows of the factor matrix corresponding to the instances' dimension. In the third step, the train and test feature matrices are concatenated along the dimension corresponding to instances and PCA is performed for decorrelation and dimensionality reduction obtaining truncated train and test matrices. The percentage (θ_p) of variance explained in the PCA loading matrix is a parameter of the method and can vary for different datasets. In the fourth step, a robust quadratic regression model is trained for predicting the instances' labels. Finally, the histograms of the class predictions of each bag are then calculated and fitted to a pseudoquadratic discriminant analysis classifier.

Input: training and test instances' features $\mathbf{U}_{\mathbf{train}}$ and $\mathbf{U}_{\mathbf{test}}$, subjects' training labels $\mathbf{Y}_{\mathbf{train}}$, percentage of variance retained by PCA θ_p, the number of bins used for the histograms (θ_H)
Output: prediction model
1. Concatenate $\mathbf{U}_{\mathbf{train}}$ and $\mathbf{U}_{\mathbf{test}}$ along the first dimension into a matrix **U.**
2. Perform PCA for decorrelation and dimensionality reduction on the concatenated matrix **U** and get the scores **T**, using the m-leading singular values that preserve θ_p of data variance.
3. Split the truncated scores matrix **T** into the corresponding $\mathbf{T}_{\mathbf{train}}$ and $\mathbf{T}_{\mathbf{test}}$ (will be used in the testing phase) scores matrix.
4. Train a robust full quadratic regression model (Equation (10)) using $\mathbf{T}_{\mathbf{train}}$ and $\mathbf{y}_{\mathbf{train}}$ (the instance labels inherited by the corresponding bag labels) and get the instance labels predictions $\mathbf{Pred}_{\mathbf{train}}$ for each instance
5. Split the vector $\mathbf{Pred}_{\mathbf{train}}$ into θ_H subsets of equal sizes and store the cutting points to be used as histogram bin edges in the testing phase
6. For each of the n training bags calculate the normalized cumulative histogram and construct the $n \times \theta_H$ feature matrix $\mathbf{A}_{\mathbf{train}}$
7. Fit a QDA model F to map $\mathbf{A}_{\mathbf{train}}$ to $\mathbf{Y}_{\mathbf{train}}$ (Equation (12)).

Algorithm 1: TensMIL (training)

2.6.1. Bayesian Optimization of Hyperparameters. The parameters of the two incorporated models, $\boldsymbol{\theta}_{\mathbf{f}}$ and $\boldsymbol{\theta}_{\mathbf{F}}$, are calculated sequentially by supervised learning, whereas the number of histogram bins (θ_H) and the percentage (θ_p) of variance retained from the set of hyperparameters are optimized externally and used as input in the learning phase. We optimized the hyperparameters using Bayesian optimization [35], based on 2-fold cross-validation on the training set.

The algorithm for the training phase of TensMIL is shown in Algorithm 1.

2.7. Assessment of the Method. As evaluation metrics for the selection of the hyperparameters and overall assessment of the methodology, we used the classification accuracy (number of correctly classified samples over total number of samples), the balanced accuracy, and the area under the ROC curve (AUC). The balanced accuracy is defined as

$$\text{Bacc} = \frac{\sum_{c=1}^{C} (T_c / n_c)}{C}, \tag{13}$$

with T_c being the number of correctly classified bags of class c and n_c the number of bags in class c, for $c = 1, \ldots, C$.

The choice of metric depended on the dataset and the metric used in prior work (i.e., by selecting the same criterion, comparison with other works was possible). We performed a series of experiments by comparing different classifiers on the same datasets using 10-fold cross-validation and report the average accuracy. For each fold, we internally used a 2-fold cross-validation procedure on the training set in order to tune the hyperparameters of each method. Once the best parameters were determined, they were used to classify the test set to record the test accuracy. Therefore, all methods were assessed on independent test sets not used during training of the classification models, nor during the optimization of the hyperparameters. For fairness, we performed grid search in each fold for finding the best parameters for each of the compared methods (our own as well as other state-of-the-art methods).

3. Results and Discussion

For the evaluation of our proposed algorithm, we employed two datasets: (i) the Breast Cancer UCSB Center for Bio-Image Informatics benchmark dataset [36] consisting of histopathology color images and (ii) multichannel recordings from the FrailSafe project [37] monitoring older people. In the next sections, we describe in brief these datasets and how they are represented by multidimensional arrays.

3.1. Data Sets

3.1.1. UCSB Breast Cancer Image Classification. The UCSB breast cancer dataset [36] consists of color histopathology images of 58 subjects of size 896×768 pixels taken from 32 benign and 26 malignant breast cancer patients. The classification problem of these images was formulated as an MIL problem first by Kandemir et al. [4] who segmented the images in 7×7 patches and extracted features from each patch. In an MIL setting, image patches are considered as instances and images as bags. In order to represent the dataset as a tensor in our approach, we also segment each image in $p \times p$ patches and vectorize the pixels of each patch per channel ending up to a matrix where the rows of the matrix represent the pixels and the 3 columns represent the RGB channels. If we arrange all these matrices across the first dimension, we obtain a tensor of dimensions $I \times J \times 3$, where $I = 58 * p^2$ and J is the number of pixels per patch. If we devote the first mode of the tensor to the instances, the second mode to the pixels, and the third mode to the RGB channels, we end up with a 3-mode tensor, containing all instances as described earlier.

3.1.2. Physiological Signals from Monitoring Older People. This data set was collected as part of the FrailSafe project [37] and consists of physiological measurements acquired from older people (age > 70 years). The measurements are acquired during ordinary all day indoor or outdoor activities. The ultimate goal is to predict aging-associated decline in reserve and function (denoted as *frailty*) through the extraction of geriatric indices from multiparametric data. Standard frailty indices, such as the Fried phenotype of frailty [38], are based on the common geriatric assessment (performed

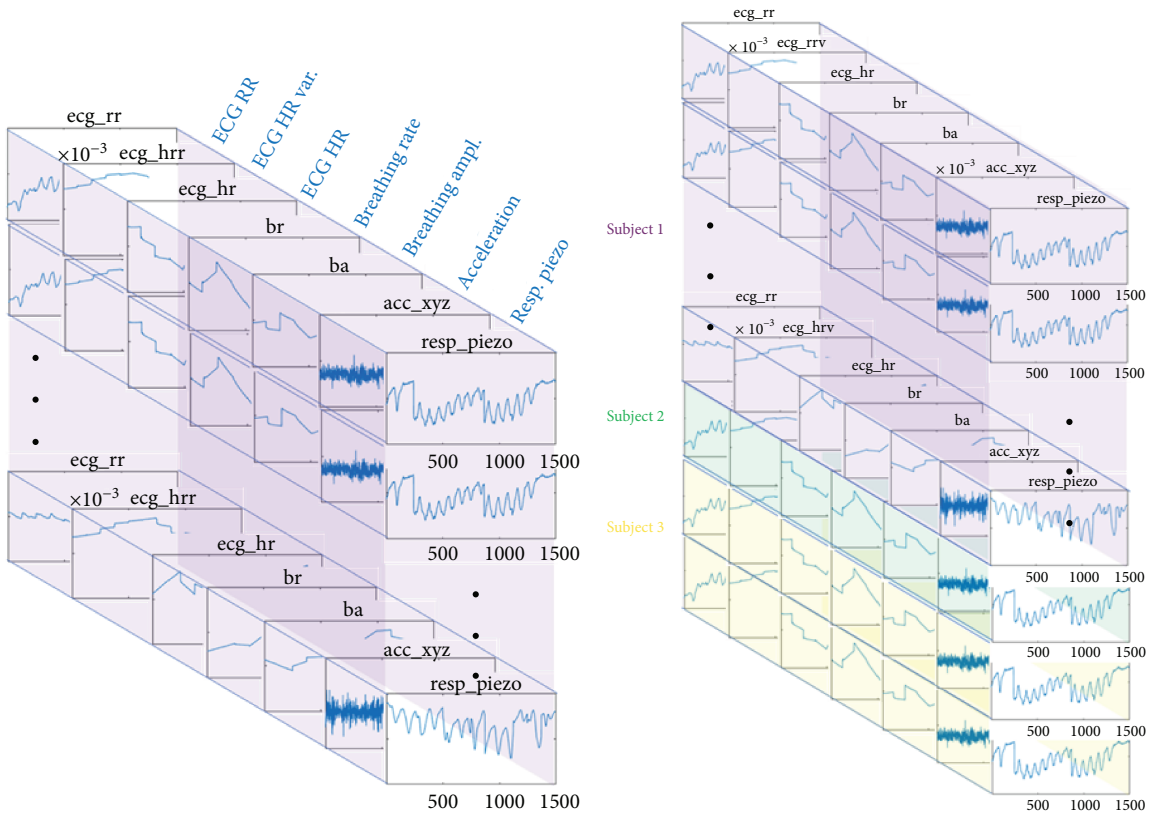


Figure 3: 3D-tensor for one subject (a) and 3D-tensor of all subjects (b).

sporadically and if considered necessary) and do not continuously monitor the health status, neither capture different medical domains. On the contrary, our goal is to extract frailty indicators from the multidimensional recordings in an effort to unobstructively monitor the health status of the older people. We assess the predictive power of physiological signals using TensMIL and the Fried score as ground truth, measured on the same time period with the acquired data. According to the Fried scale [38], three frailty stages can be distinguished: nonfrail, prefrail, and frail.

The physiological signals used in this study included time-synchronized measurements (calculated by dedicated software algorithms) from respiration, heart, posture, and physical activity. Seven channels were resampled at the same frequency (25 Hz): respiratory raw signal (by the piezoresistive sensor), magnitude of acceleration in 3 axes, breathing amplitude, breathing rate, ECG heart rate, ECG heart rate variability, and ECG RR interval. The measurements are recorded using two different devices, a fact that makes this dataset especially challenging. More details on the problem objective and the incorporated devices can be found in [1, 2].

The data representation in a tensorial form included the extraction of nonoverlapping time windows of one minute duration (i.e., 1500 time points). We consider the measurements in each time window for each subject as an instance, while the total recordings (all instances) for each subject compose one bag. In order to model the data in the form of a multidimensional array, we concatenate the multiple instances (i.e., time windows) of each subject in a 3-dimensional tensor $\mathcal{X}^{(i)}$ of dimensionality $n_i \times 1500 \times 7$, $i = 1, 2, \ldots, n$, where n_i is the number of instances available for each subject. In order to construct the whole tensor, we concatenate all tensors $\mathcal{X}^{(i)}$ along the first dimension to produce a new 3D-tensor $\mathcal{X}$ containing all instances of all bags as shown in Figure 3, resulting to a $19244 \times 1500 \times 7$ tensor. In Table 1, we summarize the available data per frailty group.

Table 1: Number of instances and percentages of bags (subjects) and instances (time windows) per class.

Class	Nr. of bags	Nr. of instances	Perc. of bags	Perc. of instances
Nonfrail	49	7127	42.24%	37.03%
Prefrail	54	8803	46.55%	45.74%
Frail	13	3314	11.21%	17.22%
Sum	116	19,244	100%	100%

3.2. Experiments

3.2.1. PARAFAC Feature Insights. Before proceeding with the results of the analysis, we provide some insights on the nature of the extracted features. As stated before, a tensor with full or missing values can be decomposed into R rank-1 components, producing a high-order dictionary that represents the

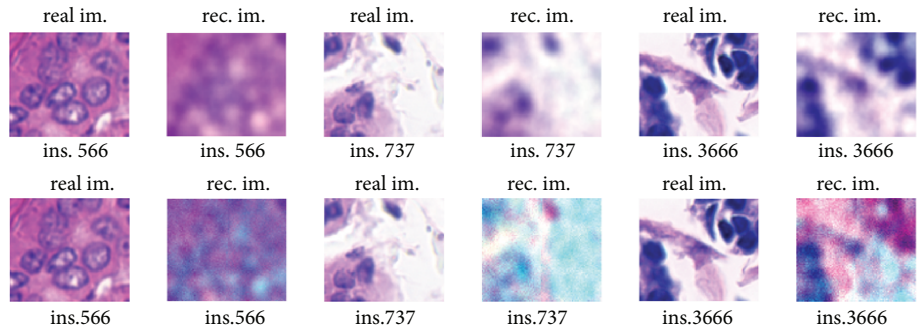


Figure 4: Random patches from BC images and their reconstruction with full values (upper row) using ALS and from 10% observed values using StrProxSGD (lower row).

latent concepts in the data. Since instances are assigned to the first dimension of the tensor, each mode-1 slice corresponds to an instance. Having computed the PARAFAC factors **U**, **V**, and **W**, we can compute, based on (1), the reconstruction of the data tensor either from full observed values or from a subset of the tensor's values (missing values).

Figure 4 depicts five random instances of the Breast Cancer dataset and their corresponding reconstructions with the ALS algorithm using full values (upper row) or with the StrProxSGD algorithm using 10% observed values (lower row). It can be observed that the reconstruction from full values results to a clearer version of the original images. As will be discussed in the next section, our experiments showed that the information preserved from the decomposition (even when using only 10% of the observed values) is sufficient to accurately classify the images in benign and malignant cases. The PARAFAC decomposition produces spatial ($\boldsymbol{V}$ from Equation (3)) and color ($\boldsymbol{W}$ from Equation (3)) components that correspond to the second and third dimension of the data tensor, which constitute the high-order dictionary. Figure 5 illustrates 40 (selected out of 120) spatial components of the dictionary. We observe also that the spatial components computed from 10% observed values are slightly noisier than the components computed from full values, a fact that showed to not significantly affect the classification accuracy.

3.2.2. Classification Assessment. The evaluation metric that we used for our experiments was different for each dataset. For the BC dataset, we report the AUC, since this metric was used for evaluation in the majority of other works. For the sake of completeness, we report also the mean test accuracy over 10 different test sets.

As reported in Table 1, the physiological signals dataset is highly unbalanced containing 11.21% of frail bags and about 42% and 47% of nonfrail and prefrail bags, respectively. For this reason, along with the test accuracy, we report also the balanced accuracy.

3.2.3. Breast Cancer Diagnosis from Histopathology Images. In this experiment, we computed the accuracy and the AUC of the proposed method against state-of-the-art MIL algorithms. We report results for each of the algorithms employing the features extracted by Kandemir et al. [4], and features extracted by the proposed method computing the PARAFAC decomposition from full values using the ALS algorithm [30] and from 10% randomly selected observed values using the StrProxSGD algorithm [31]. We should note here that the features extracted by Kandemir et al. [4] are application-specific in contrast to our extracted features that are problem-independent and can be obtained directly from any raw multidimensional data with the same procedure.

As can be observed in Table 2, when we employ the features from [4], our method is as good as JC2MIL [40] but it is outperformed by the other methods. This suggests that the feature extraction process is strongly related with the proposed MIL classification method. Indeed, when we employ the proposed features from tensor decomposition, performance improves as can be shown from the performance of TensMIL from full and 90% missing values, respectively. When using ALS features from full data, our method outperforms all other methods in terms of AUC, improving the performance by 4%–11% while in terms of accuracy TensMIL outperforms all other investigated methods and is comparable to MCILBoost. Overall, our method is comparable or outperforms other methods in terms of AUC and outperforms all other methods in terms of accuracy, except MILBOOST [41]. Concerning the case of data with missing values, our method outperforms in terms of accuracy all other investigated methods and in terms of AUC all methods except of JC2MIL to which it is comparable. Let us note here that the extraction of the hand-crafted features in [4] cannot be currently reproduced for data with missing values because the code for the feature extraction is not provided. Thus, for the missing values experiment, we compare only with the features extracted by StrProxSGD [31].

3.2.4. Physiological Signals for Frailty Prediction. In the next experiment, we evaluated the accuracy of TensMIL for frailty status prediction of older people based on motion, cardiac, and respiratory signals. In these experiments, the hyperparameters of the method were estimated by cross-validation on the training set (using the StrProxSGD algorithm for extracting features from 10% observed values) and were subsequently used for the case of full values. We performed two series of experiments. In the first experiment, we considered the three distinct frailty stages proposed by Fried (nonfrail, prefrail, and frail), whereas in the second experiment, we

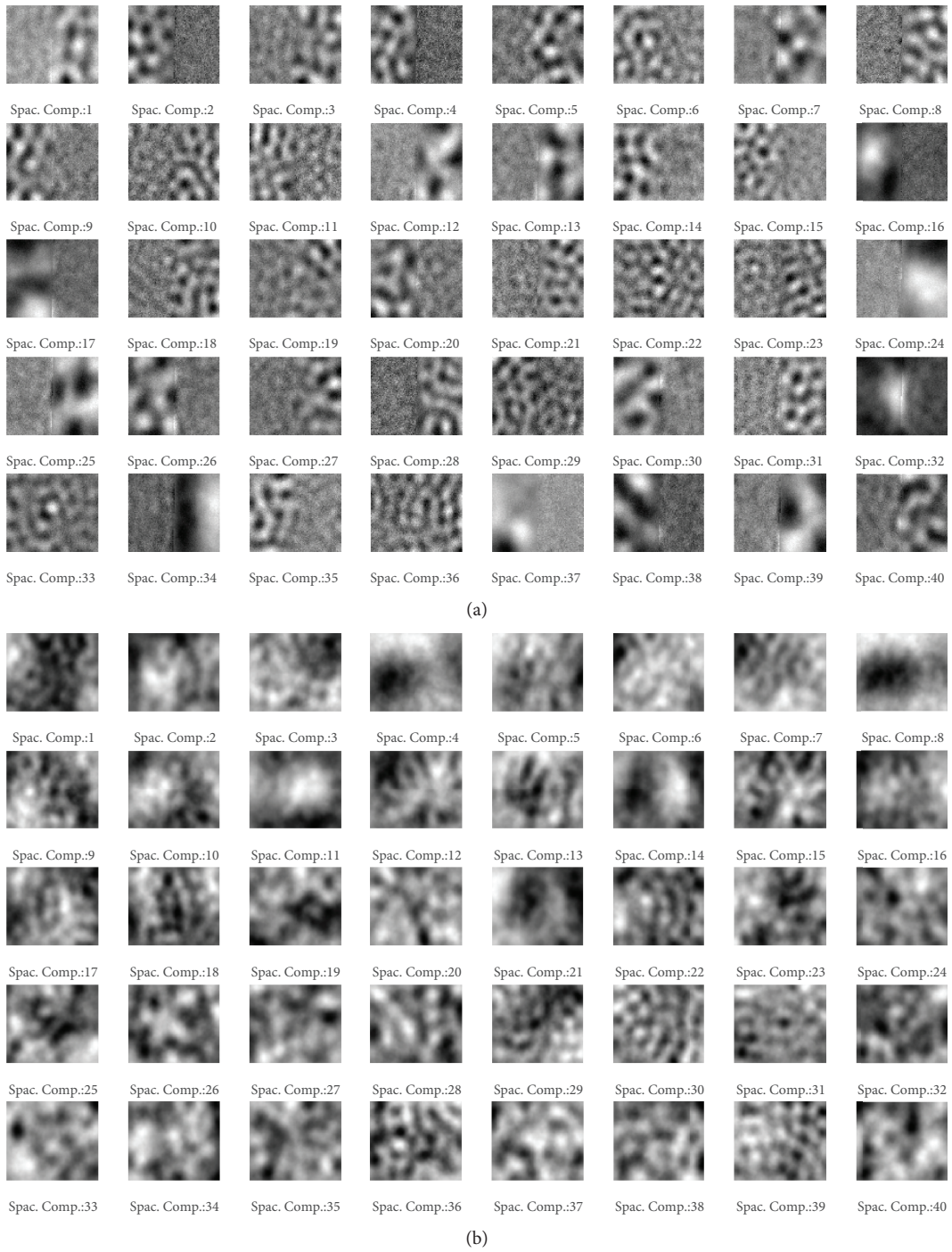


Figure 5: First 40 (out of 120) spatial components of PARAFAC model from 10% observed values using StrProxSGD (a) and from full values using ALS (b).

merged the prefrail and frail classes to create a less unbalanced dataset. Feature extraction was performed using the ALS algorithm from full data and the StrProxSGD algorithm for missing data. The results of the three class problem from full and incomplete data are shown in Table 3. When full values are considered, the accuracy of the proposed method is 45.76% (37% higher than the probability of random guess) and the balanced accuracy is 34.06% (similar to random guess). In contrast, when only 10% of the values are employed, we obtain accuracy 73.41% and balanced accuracy 67.17%, which is an improvement by a factor of 1.6 (for the accuracy) and 1.97 (for the balanced accuracy). These results strongly suggest that the data are highly noisy. Even though PARAFAC decomposition is robust against noise [43], ALS algorithm using full data could not find a good high-order dictionary for discrimination between the three classes. On the other hand, when only 10% of the data are employed, StrProxSGD could calculate a more suitable dictionary for

Table 2: Tenfold cross-validation mean test accuracy and mean AUC for the BC dataset.

BC	Kandemir [4]		ALS $R = 120$		StrProxSGD $R = 120$ (90% missing values)	
	Acc	AUC	Acc	AUC	Acc	AUC
MILES [39]	81.33 (0.15)	**0.91** (0.15)	72.67 (0.21)	0.79 (0.21)	63.33 (0.18)	0.72 (0.15)
JC2MIL [40]	74.33 (0.16)	0.84 (0.16)	72.33 (0.18)	0.78 (0.18)	77.67 (0.08)	**0.88** (0.14)
MILBoost [41]	**89.33** (0.09)	**0.94** (0.09)	81.67 (0.21)	0.87 (0.19)	68.33 (0.3)	0.77 (0.27)
MCILBoost [42]	82.33 (0.15)	**0.93** (0.12)	**85.00** (0.12)	**0.90** (0.12)	76.67 (0.22)	0.84 (0.16)
TensMIL	74.33 (0.16)	0.86 (0.16)	**84.67** (0.17)	**0.90** (0.15)	**79.33** (0.16)	0.85 (0.15)

Table 3: Test accuracy and balanced accuracy from full and 90% missing values for the 3 class problem.

Method	ALS $R = 60$		StrProxSGD $R = 60$ (90% missing values)	
	Acc	Bacc	Acc	Bacc
TensMIL	45.76 (0.13)	34.06 (0.09)	73.41 (0.01)	67.17 (0.13)

Table 4: Test accuracy from full and 90% missing values for the 2 class problem.

Methods	ALS $R = 60$	StrProxSGD $R = 60$ (90% missing values)
MILES [39]	51.59 (0.13)	67.20 (0.11)
JC2MIL [40]	**56.82** (0.07)	55.30 (0.08)
MILBoost [41]	50.83 (0.15)	54.39 (0.15)
MCILBoost [42]	45.46 (0.14)	60.91 (0.22)
TensMIL	**54.02** (0.13)	**80.83** (0.16)

the classification task. Let us note here that we do not report results from other MIL classifiers, since their performance was very poor when using the one-against-all strategy for the above multiclass problem.

Since the prefrail class lies between the frail and nonfrail class and in order to construct a more balanced dataset, we merged the prefrail with the frail group and examined the binary classification problem. As reported in Table 4, TensMIL achieved from 26.44% to 13.63% higher accuracy than the other methods by using only 10% of randomly selected values. For the case of full values, the proposed method achieves from 8.56% to 2.43% better accuracy. Only JC2MIL achieves slightly better accuracy than TensMIL.

In Table 5, we report also the mean CPU running time (across the 10-fold cross-validation sets) of TensMIL as compared to the other investigated state-of-the-art methods. The time reported corresponds to the frailty classification problem based on physiological signals, since this dataset was the largest among the two examined applications. The feature extraction component using tensor decomposition is the most time-consuming part of the method (it requires about 2.25 hours), whereas the MIL component is computationally fast. Specifically, the classification component in TensMIL requires 7 to ~52 times less training time as compared to the investigated classifiers. This fact is due to the simplicity of TensMIL since only a full quadratic regression and a QDA model have to be trained. In terms of the inference time (after feature extraction), TensMIL along with JC2MIL achieves a testing time under 1 second, which is faster than all the other investigated algorithms. We should note here that the experiments for the tensor decomposition were conducted on a Red Hat Enterprise Linux, release 6.7 (Santiago) server, comprising 162.8 GHz AMD Opteron™ 6320 processors with 62 Gb RAM, running MATLAB R2018a, while the experiments for measuring the training and test time were conducted on an Ubuntu 16.04 LTS desktop, comprising 42.0 Gz Intel® Xeon® CPU E5504 processors with 23.5 Gb RAM, running MATLAB R2017a.

Table 5: Mean CPU running time over the 10 cross-validation folds for the MIL classification component.

Methods	Training time[a]	Testing time[a]
MILES [39]	42 sec	1 sec
JC2MIL [40]	56 sec	<1 sec
MILBoost [41]	52 sec	5 sec
MCILBoost [42]	309 sec	6 sec
TensMIL	**6 sec**	**<1 sec**

[a]The experiments were conducted on an Ubuntu 16.04 LTS desktop, comprising 4 2.0 GHz Intel (R) Xeon (R) CPU E5504 processors with 23.5 Gb RAM, running MATLAB R2017a.

Finally, we compared our method with a clustering approach proposed in [1] for prediction of several clinical metrics that used statistical features from the same physiological signals, as well as other devices (GPS, game platform). Although this approach [1] showed high potential for some clinical metrics, the accuracy for the frailty index expressed by the Fried score was only 51% for the 2 class problem (nonfrail vs. prefrail and frail). TensMIL achieves 3.02% and 29.83% higher accuracy when all values or only 10% of the values are used, respectively. The clustering approach in [1] was not evaluated with missing values; however, we expect small deviations in accuracy due to the large time scale used for feature extraction and the statistical nature of the implemented features.

4. Conclusions

In this work, we exploited the high-order structure of health data through tensor decomposition aiming at extracting application-independent features that can facilitate prediction in multiple-instance learning paradigms. The prediction

models were trained in a sequential fashion to learn local and global content, while external hyperparameters were estimated by Bayesian optimization, thus providing an end-to-end architecture. The method could successfully represent and classify data with a significant amount (90%) of missing values. It was evaluated in the UCSB breast cancer benchmark dataset, as well as for prediction of aging-associated decline. In both application scenarios, the proposed method outperformed or was comparable to existing state-of-the-art machine learning techniques. Moreover, the obtained results were superior to our previous work based on statistical features and cluster analysis. Future work includes the investigation of sparse representations and addition of nonnegativity and orthogonality constraints for the extraction of more natural and interpretable data concepts.

Data Availability

The UCSB Breast Cancer data set is publically available and can be downloaded from https://bioimage.ucsb.edu/research/bio-segmentation. The data of the physiological signals for frailty prediction are collected as part of the FrailSafe Project [27] and will be available at the repository of the project: https://frailsafe-project.eu/ (contact:vasilis@ceid.upatras.gr).

Conflicts of Interest

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

Acknowledgments

The research reported in the present paper was partially supported by the FrailSafe Project (H2020-PHC-21-2015-690140) "Sensing and predictive treatment of frailty and associated co-morbidities using advanced personalized models and advanced interventions" cofunded by the European Commission under the Horizon 2020 research and innovation program. The authors want to thank all ICT (Smartex, CERTH, Gruppo Sigla) and medical partners from the FrailSafe Project for data sharing and annotations. They especially wish to thank their colleagues K. Deltouzos and S. Kalogiannis for the help with data preprocessing.

Supplementary Materials

The preliminaries of tensors and their rank decompositions. *(Supplementary Materials)*

References

[1] S. Kalogiannis, E. I. Zacharaki, K. Deltouzos et al., "Geriatric group analysis by clustering non-linearly embedded multi-sensor data," in *2018 IEEE International Conference on Innovations in Intelligent Systems and Applications (INISTA 2018)*, Thessaloniki, Greece, 2018.

[2] A. Papagiannaki, E. I. Zacharaki, K. Deltouzos et al., "Meeting challenges of activity recognition for ageing population in real life settings," in *2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pp. 1–6, Ostrava, Czech Republic, 2018.

[3] G. Lu, L. Halig, D. Wang, X. Qin, Z. G. Chen, and B. Fei, "Spectral-spatial classification for noninvasive cancer detection using hyperspectral imaging," *Journal of Biomedical Optics*, vol. 19, no. 10, article 106004, 2014.

[4] M. Kandemir, C. Zhang, and F. A. Hamprecht, "Empowering multiple instance histopathology cancer diagnosis by cell graphs," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014. MICCAI 2014. Lecture Notes in Computer Science, vol 8674*pp. 228–235, Springer, Cham.

[5] K. Mosaliganti, F. Janoos, O. Irfanoglu et al., "Tensor classification of *N*-point correlation function features for histology tissue segmentation," *Medical Image Analysis*, vol. 13, no. 1, pp. 156–166, 2009.

[6] V. G. Kanas, E. I. Zacharaki, E. Pippa, V. Tsirka, M. Koutroumanidis, and V. Megalooikonomou, "Classification of epileptic and non-epileptic events using tensor decomposition," in *2015 IEEE 15th International Conference on Bioinformatics and Bioengineering (BIBE)*, Belgrade, Serbia, November 2015.

[7] C.-F. V. Latchoumane, F. B. Vialatte, J. Solé-Casals et al., "Multiway array decomposition analysis of EEGs in Alzheimer's disease," *Journal of Neuroscience Methods*, vol. 207, no. 1, pp. 41–50, 2012.

[8] A. Cichocki, D. Mandic, L. de Lathauwer et al., "Tensor decompositions for signal processing applications: from two-way to multiway component analysis," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 145–163, 2015.

[9] N. D. Sidiropoulos, L. de Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, 2017.

[10] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "A survey of multilinear subspace learning for tensor data," *Pattern Recognition*, vol. 44, no. 7, pp. 1540–1551, 2011.

[11] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H. J. Zhang, "Multilinear discriminant analysis for face recognition," *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 212–220, 2007.

[12] X. Zhang, X. Yuan, and L. Carin, "Nonlocal low-rank tensor factor analysis for image restoration," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, June 2018.

[13] B. Du, M. Zhang, L. Zhang, R. Hu, and D. Tao, "PLTD: patch-based low-rank tensor decomposition for hyperspectral images," *IEEE Transactions on Multimedia*, vol. 19, no. 1, pp. 67–79, 2017.

[14] Y. Wang, L. Lin, Q. Zhao, T. Yue, D. Meng, and Y. Leung, "Compressive sensing of hyperspectral images via joint tensor Tucker decomposition and weighted total variation regularization," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 12, pp. 2457–2461, 2017.

[15] A. S. Lalos, I. Nikolas, E. Vlachos, and K. Moustakas, "Compressed sensing for efficient encoding of dense 3D meshes using model-based Bayesian learning," *IEEE Transactions on Multimedia*, vol. 19, no. 1, pp. 41–53, 2017.

[16] Y. Wang, D. Meng, and M. Yuan, "Sparse recovery: from vectors to tensors," *National Science Review*, vol. 5, no. 5, pp. 756–767, 2018.

[17] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, "Scalable tensor factorizations with missing data," in *2010 SIAM International Conference on Data Mining*, pp. 701–712, Columbus, OH, USA, April-May 2010.

[18] P. J. García-Laencina, J.-L. Sancho-Gómez, and A. R. Figueiras-Vidal, "Pattern classification with missing data: a review," *Neural Computing and Applications*, vol. 19, no. 2, pp. 263–282, 2010.

[19] G. Andrew, B. Recht, J. Xu, R. Nowak, and X. Zhu, "Transduction with matrix completion: three birds with one stone," in *Advances in Neural Information Processing Systems 23 (NIPS 2010)*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds., pp. 757–765, Curran Associates, Inc., 2010.

[20] E. Hazan, R. Livni, and Y. Mansour, "Classification with low rank and missing data," in *Proceedings of the 32 nd International Conference on Machine Learning*, Lille, France, 2015.

[21] D. Porro-Muñoz, R. P. W. Duin, and I. Talavera, "Missing values in dissimilarity-based classification of multi-way data," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. CIARP 2013. Lecture Notes in Computer Science, vol 8258*, J. Ruiz-Shulcloper and G. Sanniti di Baja, Eds., Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[22] J. Amores, "Multiple instance classification: review, taxonomy and comparative study," *Artificial Intelligence*, vol. 201, Supplement C, pp. 81–105, 2013.

[23] J. Foulds and E. Frank, "A review of multi-instance learning assumptions," *The Knowledge Engineering Review*, vol. 25, no. 1, pp. 1–25, 2010.

[24] L. Dong, *A Comparison of Multi-instance Learning Algorithms*, University of Waikato. The University of Waikato, Hamilton, New Zealand, 2006.

[25] X. Xu, *Statistical Learning in Multiple Instance Problems*, University of Waikato. The University of Waikato, Hamilton, New Zealand, 2003.

[26] N. Weidmann, E. Frank, and B. Pfahringer, "A two-level learning method for generalized multi-instance problems," in *Machine Learning: ECML 2003. ECML 2003. Lecture Notes in Computer Science, vol 2837*pp. 468–479, Springer, Berlin, Heidelberg.

[27] J. Wu, S. Pan, X. Zhu, C. Zhang, and X. Wu, "Multi-instance learning with discriminative bag mapping," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 6, pp. 1065–1080, 2018.

[28] J. D. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[29] R. A. Harshman, "Foundations of the PARAFAC procedure: models and conditions for an "explanatory" multi-modal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84, 1970.

[30] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[31] T. Papastergiou and V. Megalooikonomou, "A distributed proximal gradient descent method for tensor completion," in *2017 IEEE International Conference on Big Data (Big Data)*, Boston, MA, USA, December 2017.

[32] C. Leistner, A. Saffari, and H. Bischof, *MIForests: Multiple-Instance Learning with Randomized Trees*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[33] W. Dumouchel and F. O'Brien, "Integrating a robust option into a multiple regression computing environment," in *Computing and graphics in statistics*, pp. 41–48, Springer-Verlag New York, Inc., New York, NY, USA, 1991.

[34] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer Series in Statistics, Springer-Verlag New York, 2009.

[35] M. A. Gelbart, J. Snoek, and R. P. Adams, "Bayesian optimization with unknown constraints," in *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pp. 250–259, AUAI Press, Quebec City, Quebec, Canada, 2014.

[36] E. D. Gelasca, J. Byun, B. Obara, and B. S. Manjunath, "Evaluation and benchmark for biological image segmentation," in *2008 15th IEEE International Conference on Image Processing*, San Diego, CA, USA, October 2008.

[37] "Frail safe project," Available from: https://frailsafe-project.eu/.

[38] L. P. Fried, C. M. Tangen, J. Walston et al., "Frailty in older adults evidence for a phenotype," *The Journals of Gerontology: Series A*, vol. 56, no. 3, pp. M146–M157, 2001.

[39] Y. Chen, J. Bi, and J. Z. Wang, "MILES: multiple-instance learning via embedded instance selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 1931–1947, 2006.

[40] K. Sikka, R. Giri, and M. S. BartlettX. Xie, M. W. Jones, and G. K. L. Tam, "Joint clustering and classification for multiple instance learning," in *Proceedings of the British Machine Vision Conference (BMVC)*, BMVA Press, 2015.

[41] P. Viola, J. C. Platt, and C. Zhang, "Multiple instance boosting for object detection," in *Proceedings of the 18th International Conference on Neural Information Processing Systems*, pp. 1417–1424, MIT Press, Vancouver, British Columbia, Canada, 2005.

[42] Y. Xu, J. Y. Zhu, E. I. C. Chang, M. Lai, and Z. Tu, "Weakly supervised histopathology cancer image segmentation and classification," *Medical Image Analysis*, vol. 18, no. 3, pp. 591–604, 2014.

[43] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, "Scalable tensor factorizations for incomplete data," *Chemometrics and Intelligent Laboratory Systems*, vol. 106, no. 1, pp. 41–56, 2011.

Hindawi
Complexity
Volume 2018, Article ID 4075656, 14 pages
https://doi.org/10.1155/2018/4075656

Research Article

Predicting Facial Biotypes Using Continuous Bayesian Network Classifiers

Gonzalo A. Ruz [1,2] **and Pamela Araya-Díaz** [3]

[1] *Facultad de Ingeniería y Ciencias, Universidad Adolfo Ibáñez, Av. Diagonal Las Torres 2640, Peñalolén, Santiago, Chile*
[2] *Center of Applied Ecology and Sustainability (CAPES), Santiago, Chile*
[3] *Departamento del Niño y Adolescente, Área de Ortodoncia, Facultad de Odontología, Universidad Andrés Bello, Santiago, Chile*

Correspondence should be addressed to Gonzalo A. Ruz; gonzalo.ruz@uai.cl

Received 29 June 2018; Revised 7 November 2018; Accepted 15 November 2018; Published 2 December 2018

Guest Editor: Panayiotis Vlamos

Bayesian networks are useful machine learning techniques that are able to combine quantitative modeling, through probability theory, with qualitative modeling, through graph theory for visualization. We apply Bayesian network classifiers to the facial biotype classification problem, an important stage during orthodontic treatment planning. For this, we present adaptations of classical Bayesian networks classifiers to handle continuous attributes; also, we propose an incremental tree construction procedure for tree like Bayesian network classifiers. We evaluate the performance of the proposed adaptations and compare them with other continuous Bayesian network classifiers approaches as well as support vector machines. The results under the classification performance measures, accuracy and kappa, showed the effectiveness of the continuous Bayesian network classifiers, especially for the case when a reduced number of attributes were used. Additionally, the resulting networks allowed visualizing the probability relations amongst the attributes under this classification problem, a useful tool for decision-making for orthodontists.

1. Introduction

In orthodontics, it is essential to know the changes that occur during facial growth when planning a treatment, especially in children and adolescents, because the amount and direction of growth can significantly alter the need for different treatment mechanics [1, 2]. Normally, clinicians use radiographs or photographs to compute angular, linear, or proportional measurements of the face and skull to obtain growth patterns or facial biotypes [3]. One of the most popular methods to determine the facial biotype is through the VERT index proposed by Ricketts [4]. The VERT index is computed using five different features (or attributes) that allows analyzing the facial morphology [5]. Based on the VERT index, the biotypes can be classified into Dolichofacial (long and narrow face), Brachyfacial (short and wide face), and an intermediate type called Mesofacial [3, 5]. These three biotypes are shown in Figure 1.

It has been described that some attributes used in the VERT index can alter the index in patients in whom the sagittal relationship between the jaws is altered, leading to possible diagnostic errors [3]. That is why, the possibility of automatically determining the facial biotype using attributes that are not altered by the sagittal position of the jaws would eliminate the errors observed with the use of the VERT index. Thus, in this work, we propose a machine learning approach to automatically classify a patient's biotype using alternative attributes.

In recent years, we have seen great advances in the field of machine learning in relation to predictive modeling, in particular, supervised learning algorithms for classification and regression problems, such as random forests (RF) [6], support vector machines (SVM) [7], neural networks with random weights such as feedforward neural networks with random weights (RWSLFN) [8], random variable functional link neural networks (RVFLN) [9], and extreme learning machine (ELM) [10]. All of these models are achieving extraordinary performances in several applications, including orthodontics, such as the automatic Dent-landmark detection in 3D cone-beam computed tomography dental data [11], a method

(a) (b) (c)

Figure 1: Examples of the three facial biotypes. (a) Dolichofacial, (b) Brachyfacial, and (c) Mesofacial.

that objectively evaluates orthodontic treatment need and treatment outcome from the lay perspective [12], pattern classification for finding facial growth abnormalities [13], and an automated diagnostic imaging system for orthodontic treatment in dentistry [14], just to mention a few.

While high accuracy in the predictions and good generalization power are the main goals in several applications, the use of machine learning in medical treatment planning requires additionally that these models should be simple to interpret and therefore use them as a tool for decision-making. The algorithms mentioned before, although very powerful from a quantitative point of view, are somewhat limited from a qualitative aspect, in the sense that, for example, a trained SVM classifier, does not give you explicit classification rules or a simple visual interpretation on how the attributes interact in order to obtain the classification of a new data point. This issue has been tackled by other types of machine learning techniques, where the qualitative aspect plays a key role such as inductive learning algorithms [15–18] and decision trees [19]. These techniques are known as white box models (opposite to the black box models mentioned before) since the prediction process is open to the user. An interesting machine learning model that combines probability theory (quantitative) with graph theory for visualization (qualitative) is Bayesian networks introduced by J. Pearl [20], and in particular for this work, Bayesian network classifiers [21]. A Bayesian network (BN) is a directed acyclic graph (DAG), whose nodes represent discrete attributes and the edges probabilistic relationships among them. Additionally, each node has associated a conditional probability table, indicating the conditional probability for each discrete value of the node conditioned for each value of the parent nodes in the network (graph). The structure of the graph encodes the assertion that each attribute (node) is conditionally independent of its nondescendants, given its parents in the graph (this is known as the *Markov condition*). Therefore, given that a Bayesian network satisfies the Markov condition, the joint probability distribution of all the attributes can be computed in a factorized form. Bayesian networks have been applied in the domain of dentistry, for example, a decision-making system for the treatment of dental caries [22], the assessment of tooth color changes due to orthodontic treatment [23], the evaluation of the relative role and possible causal relationships among various factors affecting the diagnosis and final treatment outcome of impacted maxillary canines [24], to establish a ranking in efficacy and the best technique for coronally advanced flap-based root coverage procedures [25], a minimally invasive technique for lateral maxillary sinus elevation and to identify the relationship between the involved factors [26], and the development of a clinical decision support system to help general practitioners assess the need for orthodontic treatment in patients with permanent dentition [27].

Learning Bayesian networks from data has two components that must be handled: (1) the structure of the networks and (2) the parameters (conditional probability tables). It has been proven that learning Bayesian networks is NP-complete [28]. Therefore, several approximate learning approaches have been devised in order to simplify the learning process [29–32].

In this paper, we consider the problem of facial biotype classification using Bayesian network classifiers with continuous attributes. The rest of the paper is organized as follows. Section 2 presents a general overview of Bayesian network classifiers; then in Section 3 we describe the dataset used in this work, the continuous attribute adaptation for common Bayesian network classifiers, a description of an incremental tree construction procedure for tree like Bayesian networks, other continuous Bayesian network classifiers approaches, and the simulation setup to test and compare the classifiers. The results and discussion appear in Section 4; then the final conclusions are given in Section 5.

2. Background

Probabilistic classification consists in computing a posterior probability given an input data point. We will use the standard notation in Bayesian networks, where random variables (attributes) are denoted by capital letters, e.g., X, and particular values with lower-case letters, e.g., x. Let us consider a training set D consisting of N data points, each one characterized by n attributes $X_1, \ldots, X_n$ and their respective

output Y or class label (with c classes). Given a new input data point $\boldsymbol{v}$, this can be classified using the Bayes rule,

$$\begin{aligned} k^{\text{predict}} &= \underset{k}{\operatorname{argmax}}\, P\left(Y = k \mid X_1 = v_1, \ldots, X_n = v_n\right) \\ &= \underset{k}{\operatorname{argmax}}\, \frac{P\left(Y = k\right) P\left(X_1 = v_1, \ldots, X_n = v_n \mid Y = k\right)}{\sum_{y'=1}^{c} P\left(Y = y'\right) P\left(X_1 = v_1, \ldots, X_n = v_n \mid Y = y'\right)} \\ &= \underset{k}{\operatorname{argmax}}\, \beta P\left(Y = k\right) P\left(X_1 = v_1, \ldots, X_n = v_n \mid Y = k\right) \end{aligned} \tag{1}$$

with β the normalizing constant. From (1), we notice that there are two probabilities that can influence the resulting prediction. The first one is $P(Y = k)$ (with $k = 1, \ldots, c$) which is known as the *a priori* probability for the class value k and represents the class k distribution in D. The computation of this probability is simple, since it consists in counting the number of training examples in D for which $Y = k$ and then dividing this value by N. The second probability, $P(X_1 = v_1, \ldots, X_n = v_n \mid Y = k)$, is called the *likelihood* and corresponds to the joint probability distribution of the attributes conditioned to the class k. There are several methods to compute the joint probability distribution, in particular, using Bayesian networks, thus, given way to *Bayesian network classifiers*. The simplest approach is to consider "naively" that the attributes are independent amongst them given the class, which yields the *naive Bayesian (NB) network classifier* [33]. The prediction is computed by

$$\begin{aligned} k^{\text{predict}} &= \underset{k}{\operatorname{argmax}}\, P\left(Y = k \mid X_1 = v_1, \ldots, X_n = v_n\right) \\ &= \underset{k}{\operatorname{argmax}}\, \beta P\left(Y = k\right) \prod_{i=1}^{n} P\left(X_i = v_i \mid Y = k\right). \end{aligned} \tag{2}$$

An example of the Bayesian network representation (with $n = 5$) of this classifier is shown in Figure 2(a).

Given the difficulty of learning Bayesian networks from data, as discussed before, learning strategies have considered restrictions on the type of the structure of the network. That is the case with the seminal work by Chow and Liu [34], which developed a learning algorithm for approximating the joint distribution by a tree structure, i.e., a network with $n-1$ edges, where one node acts as the root (no incoming edges only outgoing edges), and all the rest of the nodes have only one parent node. Let Π_i represent the parent node of the attribute X_i (for $i = 1, \ldots, n$); also let i^* be the index of the node which acts as the root; therefore, $\Pi_{i^*} = \{\emptyset\}$. Under this scheme, the training set D is partitioned according to the different class labels. Then for each partition, a tree structure is learned to model the corresponding joint probability distribution P_k (with $k = 1, \ldots, c$). The prediction is computed by

$$\begin{aligned} k^{\text{predict}} &= \underset{k}{\operatorname{argmax}}\, P\left(Y = k \mid X_1 = v_1, \ldots, X_n = v_n\right) \\ &= \underset{k}{\operatorname{argmax}}\, \beta P\left(Y = k\right) \prod_{i=1}^{n} P_k\left(X_i = v_i \mid \Pi_i\right). \end{aligned} \tag{3}$$

This model is also known as the Chow-Liu (CL) classifier. An example of the CL classifier (for $n = 5$ and $c = 2$) is shown in Figure 2(b). Notice that given that $k \in \{1, \ldots, c\}$, i.e., there are c different class labels, then the CL classifier must learn c tree structures. An alternative to this is the model called the tree augmented naive Bayes classifier or TAN [21], which learns only one tree structure for all the classes. Under this model, $\Pi_i = \{X_j, Y\}$; i.e., for each node X_i, the parent set Π_i is composed of two nodes: X_j (with $j \neq i$) and the class variable Y, with exception of i^* (the attribute root node), where $\Pi_{i^*} = \{Y\}$. The prediction using the TAN classifier can be obtained by

$$\begin{aligned} k^{\text{predict}} &= \underset{k}{\operatorname{argmax}}\, P\left(Y = k \mid X_1 = v_1, \ldots, X_n = v_n\right) \\ &= \underset{k}{\operatorname{argmax}}\, \beta P\left(Y = k\right) \prod_{i=1}^{n} P\left(X_i = v_i \mid \Pi_i\right). \end{aligned} \tag{4}$$

An example of the TAN classifier (with $n = 5$) is shown in Figure 2(c).

Of course, there are other BN classifier approaches, such as *Markov blanket* of the class variable [35], K2-attribute selection (K2-AS) algorithm [36], semi-naive Bayes model [37], k-dependence Bayesian classifier [38], Bayesian classifier inference using Bayes factor [39], etc. A complete review of discrete Bayesian network classifiers can be found in [40].

It is interesting to notice that while TAN was presented as a solution to the strong independence assumption in the naive Bayes classifier, in the tests presented in the TAN paper [21], there are cases where the naive Bayes outperformed TAN. Can it be that given that TAN forces a tree structure amongst the attributes, there may be edges in the network which should not exist but are there in order to satisfy the tree structure? With this in mind, in this paper, we propose an incremental tree construction procedure which may lead to an incomplete tree structure, known as a *forest*.

3. Methods

3.1. Dataset Description and Preprocessing. The dataset consists of 182 lateral teleradiographies from Chilean patients. For each one, cephalometric analysis was performed to compute 31 continuous attributes (see Appendix) that characterize the craniofacial morphology. This dataset has been used previously to identify craniofacial patterns through clustering analysis [41]. For this work, each lateral teleradiograph has been manually classified and validated by orthodontists into one of the three classes (Brachyfacial, Dolichofacial, and Mesofacial). A visualization of the correlation matrix of the 31 attributes is shown in Figure 3, where we can appreciate that there are several attributes which are highly (more than 0.8 in absolute value) correlated.

Highly correlated attributes are essentially attributes which capture the same information, and therefore we can reduce the number of attributes by leaving only one attribute from a highly correlated set of attributes. For example, from Figure 3 we notice that Ril0 and Mc3 are highly correlated (a correlation of 0.95); this is not surprising since both attributes indicate the sagittal position of the maxilla with respect to the skull, using different cephalometric landmarks. Therefore,

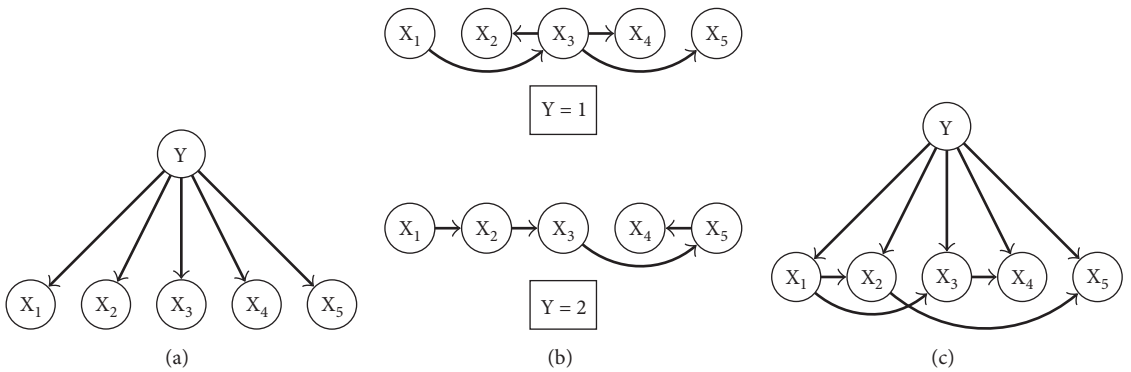


Figure 2: Examples of different Bayesian networks classifiers: (a) naive Bayes, (b) Chow-Liu tress, and (c) TAN.

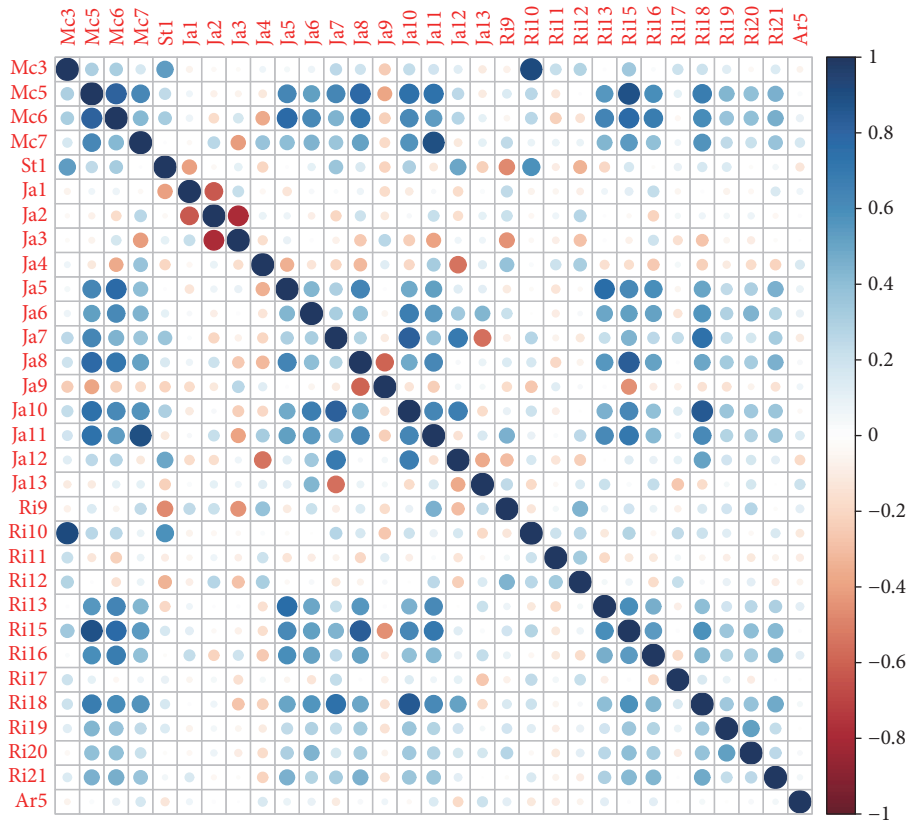


Figure 3: Correlation matrix of the 31 attributes.

we may drop Ri10 in further analyses and use only Mc3. By assuming a threshold of absolute value of 0.8 for the correlation, we excluded the following attributes: Mc5, Mc6, Ri10, Ri18, Ja8, Ja10, and Ja11. Thus, the number of attributes of the dataset is now 24. From these remaining attributes, we proceeded in visualizing their discriminatory power by performing a principal component analysis (PCA) projection of the 24-dimensional data points to a 2-dimensional space; then each point is labeled according to their class (facial biotype). The resulting visualization is shown in Figure 4.

From Figure 4, we notice that while the attributes have sufficient discriminatory power to separate the Brachyfacial class with the Dolichofacial class, the third Mesofacial class lies just between the other two, making this a difficult classification problem.

3.2. Continuous Bayesian Network Classifiers. As explained in the Introduction, we will consider Bayesian networks for this facial biotype classification problem. Given that Bayesian networks were originally formulated for discrete random variables, and our dataset has continuous variables (attributes), we need to address this issue. A typical approach is to discretize the continuous attributes and then proceed as usual. While this is a practical solution, an ideal

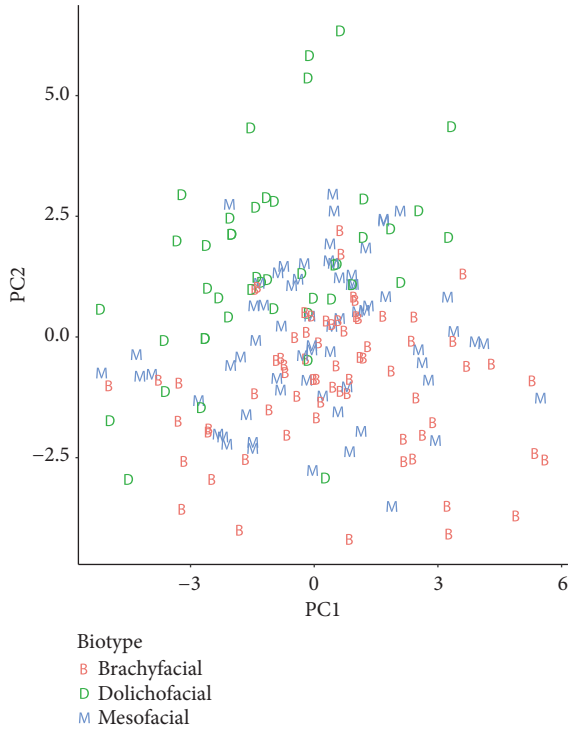


Figure 4: PCA projection of the 24-dimensional data points.

discretization is not that straightforward, and therefore, valuable information may be lost during this process. In what follows, we describe the continuous adaptation for the naive Bayes, TAN, and an incremental tree construction version of TAN, through the implementation in R, the open source software environment for statistical computing and graphics [42], that we used in our work.

3.2.1. Continuous Naive Bayes Classifier. The classification under this model is computed by (2). Here, we need to estimate the class priors $P(Y)$ and the conditional probabilities $P(X_i \mid Y)$ for $i = 1 \ldots, n$. The class priors are straightforward and can be computed by the relative frequency of each class value (Brachyfacial, Dolichofacial, and Mesofacial) in the training set. For the conditional probabilities, we partition the training set examples accordingly to their class, then for each partition we use the kernel density estimator with Gaussian kernels to compute the desired densities. The kernel density estimator function in R is called *density*. Then we use the *approx* function in R that performs linear interpolation from the estimated density to obtain the value of $P(X_i = x_i \mid Y)$ for a specific value x_i.

3.2.2. Continuous TAN Classifier. In this case, predictions are computed by (4). To evaluate in (4) we need the resulting tree structure. TAN finds this tree by applying the maximum weighted spanning tree algorithm (Kruskal's algorithm [43] or Prim's algorithm [44]) over a fully connected undirected graph of the attributes where the weights are given by the conditional mutual information measure. For the discrete case, given two attributes X_i and X_j $(i \neq j)$ with their values x_i and x_j, respectively, and the class variable Y, this measure is computed by

$$I\left(X_i; X_j \mid Y\right) = \sum_{x_i, x_j, y} P\left(x_i, x_j, y\right) \log \frac{P\left(x_i, x_j, y\right)}{P\left(x_i \mid y\right) P\left(x_j \mid y\right)}. \tag{5}$$

This is a nonnegative quantity that measures the information that X_j provides about X_i when the value of Y is known. For continuous variables, the mutual information between two attributes is given by

$$I\left(X_i; X_j\right) = \int_{X_j} \int_{X_i} p\left(x_i, x_j\right) \log \frac{p\left(x_i, x_j\right)}{p\left(x_i\right) p\left(x_j\right)} dx_i\, dx_j. \tag{6}$$

Then, the conditional mutual information for the continuous case can be computed by

$$I\left(X_i; X_j \mid Y\right) = \sum_{y} P\left(Y\right) I\left(X_i; X_j \mid Y\right). \tag{7}$$

So, $I(X_i; X_j \mid Y)$ can be computed for each class value k (with $k = 1, \ldots, c$) by (6) using all the training examples, where $Y = k$. We estimate (6) using the *knnmi* function available in the parmigene package in R [45]. This function estimates the mutual information between two attributes using entropy estimates from k-nearest neighbors distances [46]. Once we have computed the conditional mutual information for each pair of attributes, we construct the fully connected graph with the *graph.full* function in the igraph package in R [47]. Then the tree structure is obtained from the fully connected graph by using the *minimum.spanning.tree* function (also in igraph) that uses Prim's algorithm. Since we are interested in the maximum spanning tree, we use *minimum.spanning.tree* with the negative values of the conditional mutual information as weights. The resulting tree is undirected. To obtain the directed tree, we identify which is the pair of attributes with the highest edge weight (conditional mutual information), we consider from the winning pair one of the attributes as the root, and then we set the direction of all the remaining edges to be outward from it. To finally obtain the TAN classifier, we add an edge from Y to each attribute X_i. Now we are in conditions to compute (4) for a given data point. The priors $P(Y)$ can be computed as usual through relative frequencies. Then the terms $P(X_i \mid \Pi_i)$ in the product are computed as follows. For the root attribute i^* we have that $\Pi_{i^*} = Y$; thus, we can use the kernel density approach described for the naive Bayes classifier. For the rest of the terms in the product, we will have $\Pi_i = \{X_j, Y\}$ given by the tree structure. Therefore, we need to estimate conditional probabilities such as $P(X_i \mid X_j, Y)$. Using the product rule, we have that $P(X_i, X_j) = P(X_i \mid X_j)P(X_j)$. So, if we partition the training data set accordingly to the class, we can estimate the joint

probability $P(X_i, X_j)$ and the marginal probability $P(X_j)$ for each partition, then

$$P\left(X_i \mid X_j, Y\right) = \frac{P\left(X_i, X_j \mid Y\right)}{P\left(X_j \mid Y\right)}. \tag{8}$$

We estimate the joint probability with a two-dimensional kernel approach. In particular, we use the function *kde2d* in the MASS package in R [48]. This function performs a two-dimensional kernel density estimation with an axis-aligned bivariate normal kernel, evaluated on a square. Then, to obtain specific values from this density, we use the *interp.surface* function from the fields package in R [49]. This function uses bilinear weights to interpolate values on a rectangular grid to desired values. Finally, this joint probability estimate is normalized by $P(X_j \mid Y)$ which can be computed using the same approach used for the naive Bayes classifier.

3.2.3. Continuous Incremental Tree Construction Augmented Naive Bayes Classifier. We propose an alternative learning procedure for the TAN classifier, which we call incremental tree construction augmented naive Bayes (ITCAN). One of the limitations of the TAN model is that the resulting structure will always be a tree, even if some edges have very low weights (conditional mutual information). With ITCAN, we identify partial TAN solutions where some nodes (attributes) might end up with only the incoming edge from the class. The ITCAN learning procedure with a training set is as follows:

(1) Evaluate the accuracy of a naive Bayes classifier using k-fold cross validation. Let this value be A_{nb}.

(2) Learn the TAN tree structure as described in Section 3.2.2.

(3) Create a list with the edges in a descending order with respect to their weight ($edge_h$ for $h = 1, \ldots, n-1$).

(4) Assign $model \longleftarrow$ naive Bayes model.

(5) For each h in the list:

  (a) $model \longleftarrow model + edge_h$

  (b) Evaluate the accuracy of $model$ classifier using k-fold cross validation. Let this value be A_h.

(6) $h^* \longleftarrow \operatorname{argmax}_{x \in \{nb, 1, \ldots n-1\}} A_x$.

From the above learning procedure, if $h^* = nb$, then the resulting model is the naive Bayes classifier. If $h^* = n-1$, then the resulting model is the TAN classifier. For any other value of h^*, the resulting structure will be a forest, a midway solution between naive Bayes and TAN. For the results presented later on, we use $k = 5$ in the k-fold cross validation in the ITCAN learning procedure.

There have been other approaches to search for Bayesian network models bounded by naive Bayesian networks and the TAN classifier; one example is the Forest-Augmented Bayesian Network (FAN) algorithm [50]. While the ITCAN learns once the TAN tree structure, the FAN algorithm uses another approach. It first computes the conditional mutual information between all pairs of attributes, then it constructs the fully connected graph using the negative value of the conditional mutual information as weights between the attributes. But now instead of finding the minimum weighted spanning tree (like TAN), it searches for the minimum weighted spanning forest containing exactly k edges (with $k \geq 0$ defined by the user). So to explore the complete range of structures, the user must apply FAN n-times ($k = 0, \ldots, n-1$). Another difference is when FAN transforms the undirected forest into a directed forest, it does so by choosing a root vertex for every tree in the forest. This procedure could yield different structures when compared to ITCAN which uses the edges from the unique TAN structure.

3.2.4. Other Continuous Bayesian Network Classifiers Approaches. In [51] conditional Gaussian networks (CGN) classifiers were introduced. In particular, it is of interest for this work the Gaussian NB (gNB) and the Gaussian TAN (gTAN). In the case of gNB, the probabilities in the product term in (2) are approximated by

$$P\left(X_i \mid Y = k\right) \sim N\left(\mu_{i|k}, \sigma_{i|k}\right), \tag{9}$$

where $\mu_{i|k}$ and $\sigma_{i|k}$ are the mean and the standard deviation, respectively, of attribute X_i, computed by using only the examples that have a class value $Y = k$. For gTAN, the probabilities in the product term in (4) are approximated by

$$P\left(X_i \mid \Pi_i\right) \sim N\left(m_{i|k}, v_{i|k}\right), \tag{10}$$

where $m_{i|k}$ and $v_{i|k}$ are defined by

$$m_{i|k} = \mu_{i|k} + \beta_{ij|k}\left(x_j - \mu_{j|k}\right) \tag{11}$$

$$v_{i|k} = \sigma^2_{i|k} - \frac{\sigma^2_{ij|k}}{\sigma^2_{j|k}} \tag{12}$$

where we have considered X_j as the parent attribute of X_i. $\beta_{ij|k}$ is the regression coefficient of X_i on X_j conditioned to the class value $Y = k$, defined by

$$\beta_{ij|k} = \frac{\sigma_{ij|k}}{\sigma^2_{j|k}}, \tag{13}$$

where $\sigma_{ij|k}$ is the covariance between the variables X_i and X_j conditioned to k and $\sigma^2_{j|k}$ is the variance of X_j conditioned to k.

Also important to point out, under this approach, is that the conditional mutual information is computed by

$$I\left(X_i; X_j \mid Y\right) = -\frac{1}{2}\sum_{k=1}^{c} P\left(Y = k\right) \log\left(1 - \rho_k^2\left(X_i, X_j\right)\right), \tag{14}$$

where $\rho_k(X_i, X_j) = \sigma_{ij|k} / \sqrt{\sigma^2_{i|k}\sigma^2_{j|k}}$ is the correlation coefficient between X_i and X_j conditioned to the class value $Y = k$.

Another approach to handle continuous attributes is described in [52], where kernel density estimation is adopted

(similar to the approach presented in this paper) giving way to the so-called *flexible* classifiers. The flexible naive Bayes (fNB) classifier uses a similar approach as the one described in Section 3.2.1, where the conditional probabilities are computed with Gaussian kernels. One difference is the smoothing parameter h (used by the kernel density estimator) in fNB, which is the normal rule:

$$h = \left(\frac{4}{(m+2)\,q}\right)^{1/(m+4)}, \tag{15}$$

where m is the number of continuous variables in the density function to be estimated and q is the number of cases from which the estimator is learned. In our proposal, the smoothing parameter considered (used by the *density* function in R) is a rule-of-thumb described in [53]:

$$h = 0.9Aq^{-1/5} \tag{16}$$

with

$$A = \min\left(\text{standard deviation}, \frac{\text{interquartile range}}{1.34}\right). \tag{17}$$

The flexible tree augmented naive Bayes (fTAN) computes the conditional probabilities in the product term of (4) using (8) and employing a 2-dimensional Gaussian density with identity covariance matrix, similar to the continuous TAN proposed, but fTAN uses (15) to compute the bandwidth for the kernel, whereas our proposal uses (16) with the factor 0.9 changed to 4.24. Also, fTAN estimates the conditional mutual information in the following way:

$$\widehat{I}\left(X_i; X_j \mid Y\right) = \sum_{y=1}^{c} P(Y)\,\frac{1}{n_y}\sum_{l=y:1}^{y:n_y} \log \frac{\widehat{f}\left(x_i^l, x_j^l \mid y\right)}{\widehat{f}\left(x_i^l \mid y\right)\widehat{f}\left(x_j^l \mid y\right)} \tag{18}$$

where the super-index $y : j$ refers to the jth case in the partition induced by the value y, and n_y is the number of cases verifying that $Y = y$. $\widehat{f}$ are computed using the kernels described previously. On the other hand, in our proposal, we use another approach to estimate the conditional mutual information using entropy estimates from k-nearest neighbors distances [46].

Overall, when comparing to these previous continuous formulations (CGN and flexible), we notice that our proposal, based on kernel density estimates, resembles the flexible classifiers of [52], but with alternative implementations and using current available R functions.

3.3. Simulation Setup. We will compare the classification performance of the described continuous Bayesian network classifiers; in particular, we will compare our implementations, namely, cNB, cTAN, and cITCAN, with the conditional Gaussian networks approach: gNB, gTAN, and gITCAN, as well as the flexible approach: fNB, fTAN, and fITCAN. Also, we will consider the discrete versions: dNB, dTAN, and dITCAN. For this we will use the *discretize* function from the bnlearn package in R [54]. Finally we will also consider a black box classifier such as SVM. In particular, we use the *svm* function with default setting from the e1071 package in R [55].

Table 1: Performance measures for each classifier (with 24 attributes).

Algorithm	Accuracy %	Kappa
cNB	60.4±6.7	0.39±0.11
gNB	59.5±6.3	0.38±0.09
fNB	56.0±6.4	0.33±0.10
dNB	55.2±5.9	0.32±0.09
cTAN	56.8±6.3	0.34±0.09
gTAN	58.5±6.9	0.37±0.11
fTAN	43.9±6.4	0.15±0.10
dTAN	51.9±8.2	0.27±0.13
cITCAN	60.8±6.0	0.41±0.09
gITCAN	59.4±6.2	0.38±0.09
fITCAN	46.8±6.2	0.20±0.09
dITCAN	51.8±7.1	0.26±0.10
SVM	62.7±5.9	0.42±0.09

To compute the classification performance, we randomly sample 70% of the dataset examples to generate a training set and use the remaining 30% as a test set. We train the thirteen classifiers on the same training set and then compute the accuracy (the fraction of correct predictions) and the *kappa* statistic using the test set. The kappa statistic compares the accuracy of the trained model with the accuracy of a random model. To interpret the kappa value, we use the common characterization proposed in [56]: values $\leq$ 0 as indicating poor agreement, 0 − 0.2 as slight, 0.21 − 0.4 as fair, 0.41 − 0.6 as moderate, 0.61 − 0.8 as substantial, and 0.81 − 1 as almost perfect agreement.

We run the data splitting procedure 50 times and then report the average and the standard deviation of the accuracy and the kappa value for each run. To statistically compare the performance between all the algorithms we will consider the Friedman test and a post hoc test to evaluate the pairwise performance when all the algorithms are compared to each other; in particular, we will use the Nemenyi test. Further details of the process for comparison of multiple algorithms are given in [57].

4. Results and Discussions

The classification performance results for the thirteen classifiers are shown in Table 1. On average, the best performance was obtained by SVM, while within the Bayesian network classifiers, the cITCAN obtained the best performance. Also, considering the kappa value, only SVM and cITCAN correspond to the moderate interval of classification agreement with the true classes, whereas most of the other classifiers are in the fair interval. The worst performance was obtained by fTAN (and the second worst fITCAN); this could be due to the conditional mutual information estimation, where probably not enough samples were available to conduct

Table 2: The average ranks for all the algorithms (with 24 attributes).

Algorithm	Rank
SVM	2.94
cITCAN	4.37
cNB	4.65
gITCAN	4.99
gNB	5.09
gTAN	5.94
cTAN	6.96
fNB	6.99
dNB	7.81
dTAN	8.92
dITCAN	9.21
fITCAN	11.17
fTAN	11.96

a good estimation. It is important to point out that the standard deviation for the accuracy is high, and therefore, it is necessary to perform statistical tests to effectively compare the results.

We considered the null hypothesis to be tested that all the algorithms performed the same and that the observed differences were merely random. We conducted the Friedman test in order to analyze if there are statistically significant differences for all the algorithms. All the algorithms are ranked for each dataset (run) separately, where the best performing algorithm is the one obtaining the lowest rank. Table 2 shows the average rank for each algorithm.

The Friedman statistic is given by the following:

$$\chi_F^2 = \frac{12D}{c\,(c+1)} \left[\sum_j R_j^2 - \frac{c\,(c+1)^2}{4} \right], \tag{19}$$

where R_j^2 is the j-th average rank of the algorithms. The statistic is distributed according to χ_F^2 with $c - 1$ degrees of freedom and D is the number of datasets. For the comparison of all the algorithms with the Friedman test, the χ_F^2 statistic is 300.2 and the p value is <2.2e-16, which rejects the null hypothesis that all the algorithms have the same performance.

Then, a post hoc test is performed to evaluate the pairwise performance when all the algorithms are compared to each other. The Nemenyi test with $\alpha = 0.05$ was applied, and the results are presented in Table 3. When comparing SVM with all the other classifiers, we notice that the null hypothesis cannot be rejected when compared to cNB, gNB, cITCAN, and gITCAN, respectively, since there are no statistically significant differences between them, whereas for our second best classifier, cITCAN, we notice that the null hypothesis cannot be rejected when compared to cNB, gNB, gTAN, gITCAN, and SVM, respectively.

Figure 5 shows the best cTAN model obtained throughout the 50 runs. We notice that Ja5 and Ri15 are the two attributes with the most outgoing edges (apart from the obvious class node Biotype), conditioning the probabilities of the other attributes. In particular, Ja5 is the parent node of Ri13, Ri16, Ja13, Ja6, and Ri15. This can be explained, in part, by the following: Ja5 as well as Ri13 corresponds to the length of the anterior cranial base using different landmarks. Ja13 and Ja6 correspond to variables given by the posterior cranial length. In this case, the relationship is explained given the fact that the growth of the anterior cranial base (Ja5) and posterior cranial base (Ja13 and Ja16) depends on a common factor, which is the growth of the brain; therefore, there is a linear proportionality between both structures. There is no biologically direct relationship to explain the relation between Ja5 with Ri15 and Ri16, except that, as in any biological system, there is a proportional and compensatory relationship between the structures tending to maintain the functionality and stability of the systems.

On the other hand, Ri15 is the parent node of Ri21, Mc7, Ri20, and Mc3. In this case, a greater or smaller mandibular size is directly related to a larger or smaller size of all its components, such as the width of the symphysis (Ri21) and width of the condyle (Ri20), which explains the relationship between these variables and the size of the mandibular body (Ri15). On the other hand, there is no biologically direct relation to explain the relationship between attribute Ri15 with Mc3 and Mc7. Attribute Mc3 points out the sagittal position of the maxilla, which is independent of the size of the mandibular body (Ri15), and Mc7 is a vertical relationship (lower facial height) that is not directly influenced by the mandibular size.

The best cITCAN model obtained throughout the 50 runs is shown in Figure 6. We notice that it is a forest, where only 5 edges are considered from the total 23 of the cTAN model (without counting the outgoing edges of the class variable). Here we observe that the influence of Ja5 on Ri13 and Ri15 is still required.

We explore the possibility to improve the classification performance by identifying the most relevant attributes for classification and then proceed to repeat the simulations with a reduced number of attributes. For this, the *importance* function from the randomForest package in R [58] was used. This function computes the importance of each attribute based on the Gini importance, a measure used to quantify the node impurity during the tree inference process (in decision trees or random forests). The result is shown in Figure 7.

We observe that Ja4 is the attribute with the most discriminatory power. We proceed to select the top 4 attributes, i.e., Ja4, Ja12, Mc7, and Mc3. In particular, the first three correspond to measurements that describe vertical dimensions, which is directly related to the determination of the biotype, since the primary difference between them is the relationship between the vertical dimensions of the anterior and posterior region of the craniofacial complex. It is noteworthy that attribute Mc3 is among those of higher importance, since it indicates the sagittal position of the maxilla with respect to the skull, a characteristic that is independent and not directly related to the characteristics that allow the differentiation of biotypes.

With these four attributes, we repeat the performance evaluations and the statistical tests using the same 50 runs.

Table 3: Nemenyi test for single models (with 24 attributes) in terms of accuracy (%).

	cNB	gNB	fNB	dNB	cTAN	gTAN	fTAN	dTAN	cITCAN	gITCAN	fITCAN	SVM
gNB	1.00											
fNB	0.12	0.41										
dNB	0.00	0.03	0.99									
cTAN	0.13	0.44	1.00	0.99								
gTAN	0.91	0.99	0.98	0.44	0.99							
fTAN	0.00	0.00	0.00	0.00	0.00	0.00						
dTAN	0.00	0.00	0.39	0.97	0.36	0.01	0.01					
cITCAN	1.00	0.99	0.04	0.00	0.04	0.72	0.00	0.00				
gITCAN	1.00	1.00	0.33	0.01	0.35	0.99	0.00	0.00	0.99			
fITCAN	0.00	0.00	0.00	0.00	0.00	0.00	0.99	0.16	0.00	0.00		
dITCAN	0.00	0.00	0.17	0.85	0.16	0.00	0.02	1.00	0.00	0.00	0.36	
SVM	0.59	0.22	0.00	0.00	0.00	0.01	0.00	0.00	0.83	0.29	0.00	0.00

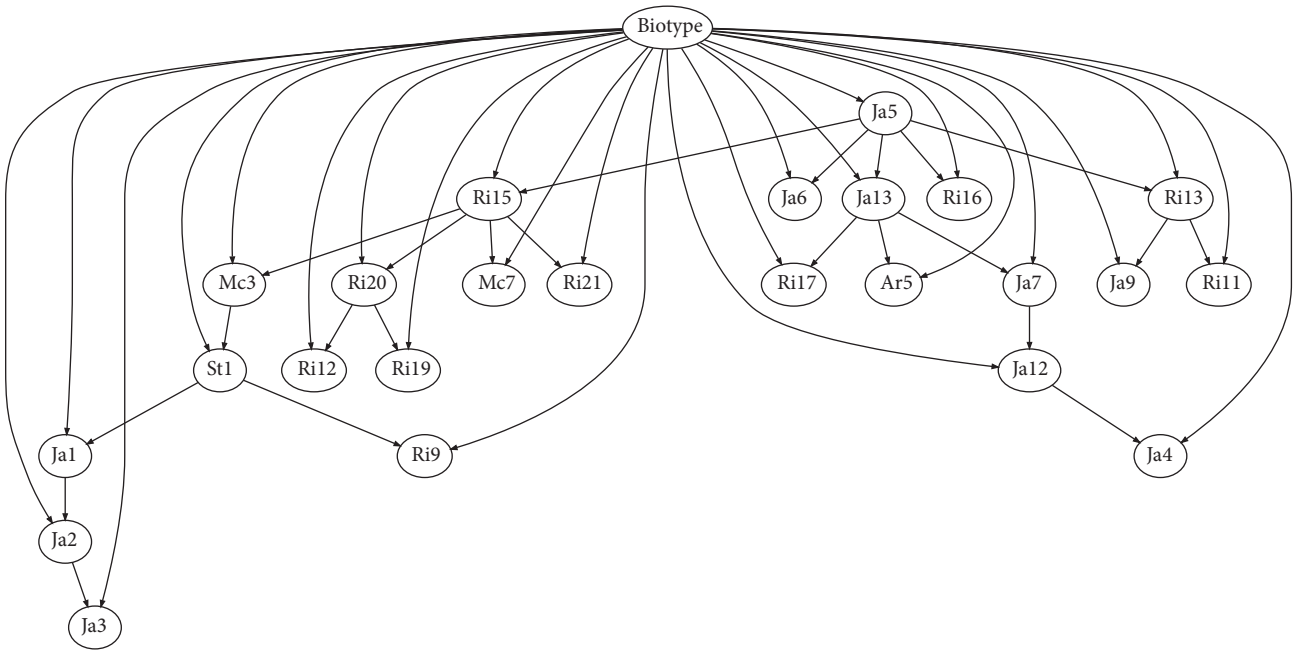


Figure 5: The cTAN classifier for the facial biotype dataset.

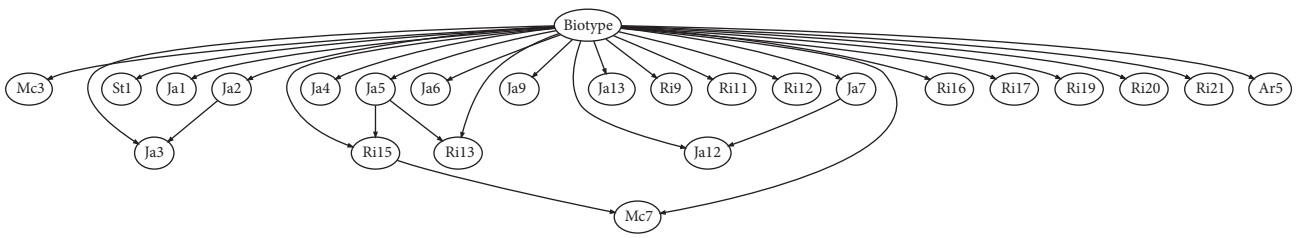


Figure 6: The cITCAN classifier for the facial biotype dataset.

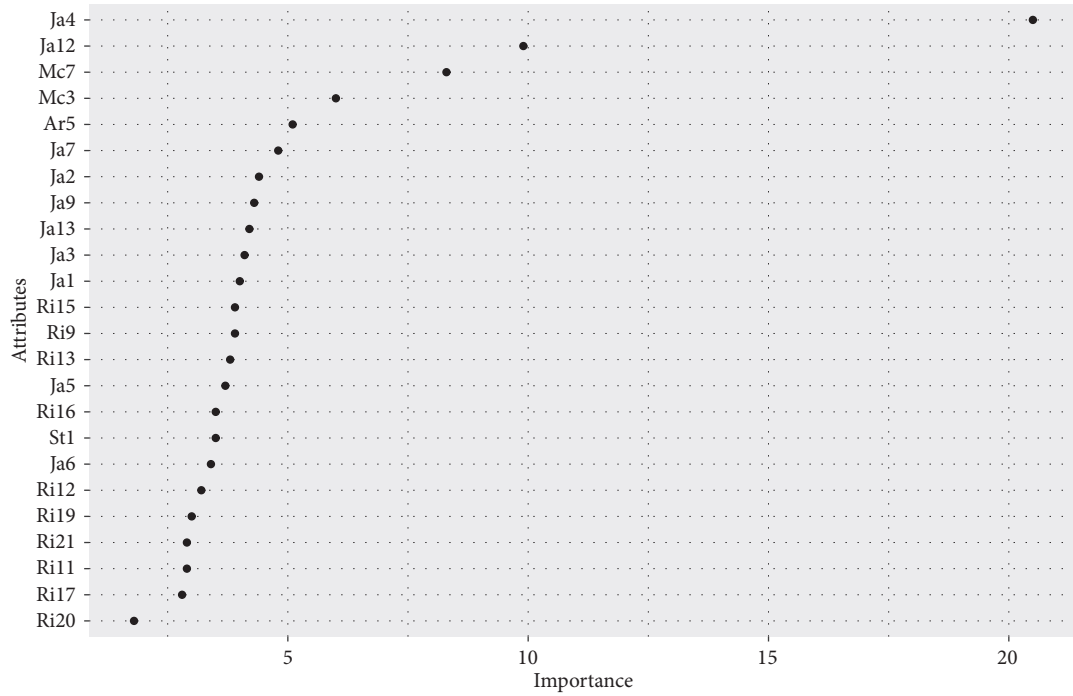


Figure 7: Attribute importance ranking based on the Gini importance measure.

Table 4: Performance measures for each classifier (with 4 attributes).

Algorithm	Accuracy %	Kappa
cNB	70.0±4.7	0.53±0.07
gNB	67.9±4.7	0.51±0.07
fNB	65.3±5.6	0.47±0.08
dNB	65.2±5.6	0.47±0.08
cTAN	68.2±5.6	0.51±0.09
gTAN	69.3±5.5	0.53±0.08
fTAN	49.6±6.9	0.22±0.10
dTAN	60.2±5.6	0.39±0.08
cITCAN	70.4±4.9	0.55±0.08
gITCAN	69.3±4.7	0.53±0.07
fITCAN	48.2±6.3	0.22±0.09
dITCAN	60.3±5.8	0.39±0.09
SVM	69.9±5.1	0.53±0.08

Table 5: The average ranks for all the algorithms (with 4 attributes).

Algorithm	Rank
cITCAN	3.96
cNB	4.14
SVM	4.22
gTAN	4.66
gITCAN	4.81
cTAN	5.47
gNB	5.61
fNB	7.01
dNB	7.08
dITCAN	9.52
dTAN	9.91
fTAN	12.15
fITCAN	12.46

The accuracy and kappa values are shown in Table 4. Overall, we see improvements in all the performance measures, in particular, the accuracy increased approximately by 10% in several classifiers. In relation to the kappa values, we notice that now cNB, gNB, fNB, dNB, cTAN, gTAN, cITCAN, gITCAN, and SVM are in the moderate interval of classification agreement with the true classes, with cITCAN obtaining the highest value. The worst accuracy and kappa value was obtained by the fITCAN classifier.

Following the same statistical tests as before, Table 5 shows the average rank for each algorithm. For the comparison of all the algorithms with the Friedman test, the χ_F^2 statistic is 372.66 and the p value is <2.2e-16, which rejects the null hypothesis that all the algorithms have the same performance.

Similar as before, a post hoc test was performed to evaluate the pairwise performance when all the algorithms are compared to each other. The Nemenyi test with $\alpha = 0.05$ was applied, and the results are presented in Table 6.

When comparing cITCAN with all the other classifiers, we notice that the null hypothesis cannot be rejected when compared to cNB, gNB, cTAN, gTAN, gITCAN, and SVM, respectively, since there are no statistically significant differences between them, whereas for our second ranked best classifier, cNB, we notice that the null hypothesis cannot be

Table 6: Nemenyi test for single models (with 4 attributes) in terms of accuracy (%).

	cNB	gNB	fNB	dNB	cTAN	gTAN	fTAN	dTAN	cITCAN	gITCAN	fITCAN	SVM
gNB	0.80											
fNB	0.01	0.85										
dNB	0.01	0.80	1.00									
cTAN	0.89	1.00	0.75	0.68								
gTAN	1.00	0.99	0.11	0.09	0.99							
fTAN	0.00	0.00	0.00	0.00	0.00	0.00						
dTAN	0.00	0.00	0.01	0.01	0.00	0.00	0.17					
cITCAN	1.00	0.65	0.01	0.00	0.77	0.99	0.00	0.00				
gITCAN	0.99	0.99	0.19	0.15	0.99	1.00	0.00	0.00	0.99			
fITCAN	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.06	0.00	0.00		
dITCAN	0.00	0.00	0.07	0.09	0.00	0.00	0.04	1.00	0.00	0.00	0.01	
SVM	1.00	0.86	0.02	0.01	0.92	1.00	0.00	0.00	1.00	0.99	0.00	0.00

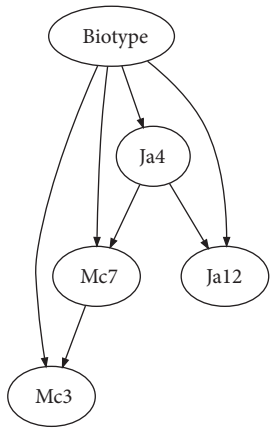


Figure 8: The cTAN classifier for the facial biotype dataset using only four attributes.

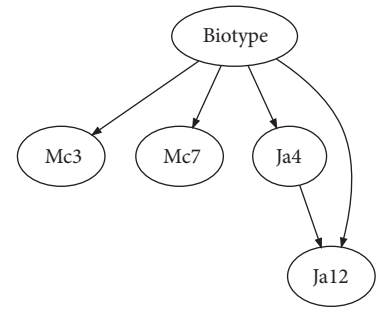


Figure 9: The cITCAN classifier for the facial biotype dataset using only four attributes.

rejected when compared to gNB, cTAN, gTAN, cITCAN, gITCAN, and SVM, respectively.

The resulting network structures for cTAN and cITCAN (for the simulations with only four attributes) are shown in Figures 8 and 9, respectively.

Overall, dropping irrelevant attributes contributed to the improvements of the classification performances of all the models.

5. Conclusion

We have presented adaptations for popular Bayesian network classifiers (naive Bayes and TAN) to handle continuous attributes. Additionally, we have proposed an incremental tree construction procedure for TAN (ITCAN) that may yield forest structures that model more effectively the posterior class distribution, thus, yielding competitive classification performances. We have applied these models to the facial biotype classification problem. Through classification performance measures and comparisons with other continuous Bayesian network classifiers approaches, we showed that these models can obtain competitive results when compared to a black-box model such as SVM. Also, the resulting network structures help to shed light on the probability relations amongst the attributes, which contributes to the understanding of their role in the classification process.

As an application in the context of medical informatics, trained Bayesian network classifiers for facial biotype classification can be used as an initial automatic screening process by orthodontists. Then, based on the posterior probability of the assigned class for each patient, define a threshold from which classifications with posterior probabilities below this threshold would require a manual validation by the orthodontist.

Appendix

Table 7 presents a list of the attributes and their description, used in this work.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Table 7: A description of the attributes used in this work.

Attribute	Description
Mc3	Linear distance from point A to nasion perpendicular
Mc5	Mandibular length (Condylion to Gnathion)
Mc6	Maxillary length (Condylion to Point A)
Mc7	Lower anterior facial height (Anterior nasal spine to menton)
St1	SNA angle (Sella-Nasion-A)
Ja1	Saddle angle (Nasion-Sella-Articulare)
Ja2	Articular angle (Sella-Articulare-Gonion)
Ja3	Upper Gonial angle (Articulare-Gonion-Nasion)
Ja4	Lower Gonial angle (Nasion-Gonion-Menton)
Ja5	Anterior cranial base length (Sella to Nasion)
Ja6	Posterior cranial base length (Sella to Articulare)
Ja7	Ramus height (Articulate to Gonion)
Ja8	Mandibular corpus length (Gonion to Gnathion)
Ja9	Cranial base and Mandibular length ratio (Sella-Nasion/Gonion-Gnathion)
Ja10	Posterior facial height (Sella to Gonion)
Ja11	Anterior facial height (Nasion to Menton)
Ja12	Jarabak's ratio (Posterior facial height/Anterior facial height)
Ja13	Posterior cranial base and ramus height ratio (Sella-Articulare/Articulare-Gonion)
Ri9	Maxillary height angle (Nasion-Center of Face-A)
Ri10	Maxillary depth angle (Porion-Orbitale and Nasion-A)
Ri11	Palatal plane angle (Porion-Orbitale/anterior nasal spine-posterior nasal spine)
Ri12	Cranial deflection (Basion-Nasion/Porion-Orbitale)
Ri13	Anterior Cranial length (Center of Cranium to Nasion)
Ri15	Mandibular corpus axis (point Xi to point protuberance menti or Pm)
Ri16	Articular cavity position: Porion to Ptv (intersection of the distal outline of pterygomaxillary fissure perpendicular to the porion-orbitale plane)
Ri17	Mandibular ramus position (Porion-Orbitale/Center of Face-point Xi)
Ri18	Posterior height (Gonion to Center of Face)
Ri19	Condylar height
Ri20	Condylar neck length
Ri21	Symphysis length
Ar5	Nasolabial angle (Columella-Subnasale-upper lip)

Acknowledgments

The authors would like to thank Conicyt-Chile under grant Fondecyt 1180706 and Basal (CONICYT)-CMM, for financially supporting this research.

References

[1] R. S. Nanda, "The contributions of craniofacial growth to clinical orthodontics," *American Journal of Orthodontics and Dentofacial Orthopedics*, vol. 117, no. 5, pp. 553–555, 2000.

[2] R. Mangla, V. Dua, M. Khanna, N. Singh, and P. Padmanabhan, "Evaluation of mandibular morphology in different facial types," *Contemporary Clinical Dentistry*, vol. 2, no. 3, p. 200, 2011.

[3] E. De Novaes Benedicto, S. A. Kairalla, G. M. S. Oliveira, L. R. M. Junior, H. D. Rosário, and L. R. Paranhos, "Determination of vertical characteristics with different cephalometric measurements," *European Journal of Dentistry*, vol. 10, no. 1, pp. 116–120, 2016.

[4] R. M. Ricketts, "Planning treatment on the basis of the facial pattern and an estimate of its growth," *The Angle Orthodontist*, vol. 27, pp. 14–37, 1957.

[5] S. G. F. Gomes, W. Custodio, F. Faot, A. A. Del Bel Cury, and R. C. M. R. Garcia, "Masticatory features, EMG activity and muscle effort of subjects with different facial patterns," *Journal of Oral Rehabilitation*, vol. 37, no. 11, pp. 813–819, 2010.

[6] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[7] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[8] W. F. Schmidt, M. A. Kraaijveld, and R. P. Duin, "Feedforward neural networks with random weights," in *Proceedings of the 11th IAPR International Conference on Pattern Recognition. Vol.II. Conference B: Pattern Recognition Methodology and Systems*, pp. 1–4, The Hague, Netherlands, 1992.

[9] Y.-H. Pao, G.-H. Park, and D. J. Sobajic, "Learning and generalization characteristics of the random vector functional-link net," *Neurocomputing*, vol. 6, no. 2, pp. 163–180, 1994.

[10] 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.

[11] E. Cheng, J. Chen, J. Yang et al., "Automatic Dent-landmark detection in 3-D CBCT dental volumes," in *Proceedings of the 33rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS 2011*, pp. 6204–6207, USA, September 2011.

[12] X. Wang, B. Cai, Y. Cao et al., "Objective method for evaluating orthodontic treatment from the lay perspective: An eye-tracking study," *American Journal of Orthodontics and Dentofacial Orthopedics*, vol. 150, no. 4, pp. 601–610, 2016.

[13] A. Lakkshmanan, A. A. Shri, and E. Aruna, "Pattern classification for finding facial growth abnormalities," in *Proceedings of the 2013 4th IEEE International Conference on Computational Intelligence and Computing Research, IEEE ICCIC 2013*, India, December 2013.

[14] S. Murata, C. Lee, C. Tanikawa, and S. Date, "Towards a fully automated diagnostic system for orthodontic treatment in dentistry," in *Proceedings of the 13th IEEE International Conference on eScience, eScience 2017*, pp. 1–8, New Zealand, October 2017.

[15] J. R. Quinland, "Learning efficient classification procedures and their applications to chess end games," in *Machine Learning an Artificial Intelligence Approach*, R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, Eds., pp. 463–482, Morgan Kaufmann, Los Altos, Calif, USA, 1983.

[16] R. S. Michalski, "A theory and methodology of inductive learning," in *Readings in Machine Learning*, J. W. Shavlik and T.

G. Dietterich, Eds., pp. 70–95, Morgan Kaufmann, San Mateo, Calif, USA, 1990.

[17] D. T. Pham and M. S. Aksoy, "RULES: A simple rule extraction system," *Expert Systems with Applications*, vol. 8, no. 1, pp. 59–65, 1995.

[18] D. T. Pham and S. S. Dimov, "An efficient algorithm for automatic knowledge acquisition," *Pattern Recognition*, vol. 30, no. 7, pp. 1137–1143, 1997.

[19] J. R. Quinlan, *C4.5: programs for machine learning*, Morgan Kaufmann, 1993.

[20] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, Boston, Mass, USA, 1988.

[21] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine Learning*, vol. 29, no. 2-3, pp. 131–163, 1997.

[22] V. K. Mago, B. Prasad, A. Bhatia, and A. Mago, "A decision making system for the treatment of dental caries," *Studies in Fuzziness and Soft Computing*, vol. 230, pp. 231–242, 2008.

[23] A.-S. Mesaros, S. Sava, D. Mitrea et al., "In vitro assessment of tooth color changes due to orthodontic treatment using knowledge discovery methods," *Journal of Adhesion Science and Technology*, vol. 29, no. 20, pp. 2256–2279, 2015.

[24] M. Nieri, A. Crescini, R. Rotundo, T. Baccetti, and P. Cortellini, "Factors affecting the clinical approach to impacted maxillary canines: A Bayesian network analysis," *American Journal of Orthodontics and Dentofacial Orthopedics*, vol. 137, no. 6, pp. 755–762, 2010.

[25] J. Buti, M. Baccini, M. Nieri, M. La Marca, and G. P. Pini-Prato, "Bayesian network meta-analysis of root coverage procedures: Ranking efficacy and identification of best treatment," *Journal of Clinical Periodontology*, vol. 40, no. 4, pp. 372–386, 2013.

[26] M. Merli, M. Moscatelli, G. Mariotti, U. Pagliaro, F. Bernardelli, and M. Nieri, "A minimally invasive technique for lateral maxillary sinus floor elevation: A Bayesian network study," *Clinical Oral Implants Research*, vol. 27, no. 3, pp. 273–281, 2016.

[27] B. Thanathornwong, "Bayesian-based decision support system for assessing the needs for orthodontic treatment," *Health Informatics Journal*, vol. 24, no. 1, pp. 22–28, 2018.

[28] D. M. Chickering, *Learning Bayesian Networks is NP-Complete*, Springer, New York, Ny, USA, 1996.

[29] G. F. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 9, no. 4, pp. 309–347, 1992.

[30] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning Bayesian networks: the combination of knowledge and statistical data," *Machine Learning*, vol. 20, no. 3, pp. 197–243, 1995.

[31] D. Heckerman, "A tutorial on learning with bayesian networks," Tech. Rep. MSR-TR-95-06, Microsoft Research, 1995.

[32] R. E. Neapolitan, *Learning Bayesian networks*, Pearson Prentice Hall, Upper Saddle River, NJ, USA, 2004.

[33] R. O. Duda and P. E. Hart, *Pattern Classification And Scene Analysis*, John Wiley & Sons, New York, Ny, USA, 1973.

[34] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 462–467, 1968.

[35] D. Margaritis and S. Thrun, "Bayesian network induction via local neighborhoods," in *Advances in Neural formation Processing Systems 12*, S. A. Solla, T. K. Leen, and K. Müller, Eds., pp. 505–511, MIT Press, 2000.

[36] G. M. Provan and M. Singh, "Learning Bayesian Networks Using Feature Selection," in *Learning from Data*, vol. 112 of *Lecture Notes in Statistics*, pp. 291–300, Springer, New York, NY, USA, 1996.

[37] M. J. Pazzani, *Constructive Induction of Cartesian Product Attributes*, Springer US, Boston, Mass, USA, 1998.

[38] M. Sahami, "Learning limited dependence bayesian classifiers," in *Proceedings of the In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 335–338, 1996.

[39] G. A. Ruz and D. T. Pham, "Building Bayesian network classifiers through a Bayesian complexity monitoring system," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 223, no. 3, pp. 743–755, 2009.

[40] C. Bielza and P. Larrañaga, "Discrete bayesian network classifiers: A survey," *ACM Computing Surveys*, vol. 47, no. 1, 2014.

[41] P. Araya-Díaz, G. A. Ruz, and H. M. Palomino, "Discovering craniofacial patterns using multivariate cephalometric data for treatment decision making in orthodontics," *International Journal of Morphology*, vol. 31, no. 3, pp. 1109–1115, 2013.

[42] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2016.

[43] J. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of the American Mathematical Society*, vol. 7, pp. 48–50, 1956.

[44] R. C. Prim, "Shortest connection networks and some generalizations," *Bell Labs Technical Journal*, vol. 36, no. 6, pp. 1389–1401, 1957.

[45] G. Sales and C. Romualdi, "Parmigene—a parallel R package for mutual information estimation and gene network reconstruction," *Bioinformatics*, vol. 27, no. 13, pp. 1876-1877, 2011.

[46] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Physical Review E: Statistical, Nonlinear, and Soft Matter Physics*, vol. 69, Article ID 066138, 2004.

[47] G. Csardi and T. Nepusz, "The igraph software package for complex network research," *InterJournal*, vol. 1695, 2006.

[48] W. N. Venables and B. D. Ripley, *Modern Applied Statistics with S*, Springer, New York, NY, USA, 2002.

[49] D. Nychka, R. Furrer, J. Paige, and S. Sain, "Fields: Tools for spatial data," R package version 9.0, 2015.

[50] P. J. Lucas, "Restricted BAYesian network structure learning," in *Advances in BAYesian networks*, vol. 146 of *Stud. Fuzziness Soft Comput.*, pp. 217–234, Springer, Berlin, 2004.

[51] A. Pérez, P. Larrañaga, and I. Inza, "Supervised classification with conditional Gaussian networks: increasing the structure complexity from naive Bayes," *International Journal of Approximate Reasoning*, vol. 43, no. 1, pp. 1–25, 2006.

[52] A. Pérez, P. Larrañaga, and I. Inza, "Bayesian classifiers based on kernel density estimation: flexible classifiers," *International Journal of Approximate Reasoning*, vol. 50, no. 2, pp. 341–362, 2009.

[53] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman & Hall, London, UK, 1986.

[54] M. Scutari, "Learning Bayesian networks with the bnlearn R Package," *Journal of Statistical Software* , vol. 35, no. 3, pp. 1–22, 2010.

[55] D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, and F. Leisch, "e1071: Misc Functions of the Department of Statistics,

Probability Theory Group (Formerly: E1071), TU Wien," R package version 1.6-8, 2017.

[56] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977.

[57] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[58] A. Liaw and M. Wiener, "Classification and regression by randomForest," *The R Journal*, vol. 2, no. 3, pp. 18–22, 2002.

Hindawi
Complexity
Volume 2018, Article ID 7536731, 8 pages
https://doi.org/10.1155/2018/7536731

Research Article

An Optimal Algorithm for Determining Risk Factors for Complex Diseases: Depressive Disorder, Osteoporosis, and Fracture in Young Patients with Breast Cancer Receiving Curative Surgery

Chieh-Yu Liu[1,2] **and Chun-Hung Chang**[3]

[1]*Biostatistical Consulting Lab, Department of Speech Language Pathology and Audiology, National Taipei University of Nursing and Health Sciences, Taipei, Taiwan*
[2]*Department of Midwifery and Women Health Care, School of Nursing, National Taipei University of Nursing and Health Sciences, Taipei, Taiwan*
[3]*Department of Psychiatry & Brain Disease Research Center, China Medical University Hospital, Taichung, Taiwan*

Correspondence should be addressed to Chieh-Yu Liu; chiehyu@ntunhs.edu.tw

Received 7 February 2018; Accepted 14 May 2018; Published 4 July 2018

Academic Editor: Panayiotis Vlamos

This study proposed a novel algorithm to investigate the risk factors for complex diseases. We employed the novel algorithm to determine the risk factors for depressive disorder, osteoporosis, and fracture in young patients with breast cancer who were receiving curative surgery. The novel algorithm has three steps. First, multiple correspondence analysis (MCA) is used to transform the raw data set into a multidimensional coordinate matrix. Second, the expectation-maximization (EM) algorithm is used for clustering the multidimensional coordinates for each category of variable. Third, v-fold cross-validation is incorporated into the coordinate matrix obtained using the MCA-EM algorithm to determine the optimal clustering of complex diseases and risk factors. A total of 4108 patients with breast cancer aged 20–39 years were enrolled. The results revealed that depressive disorder, osteoporosis, and fracture were clustered with liver cirrhosis, chronic obstructive pulmonary disease (COPD), distant metastasis, and primary metastatic and adjuvant therapies, namely, chemotherapy, radiotherapy, tamoxifen, aromatase inhibitors, and trastuzumab. Among the risk factors identified using this novel algorithm, liver cirrhosis and COPD have been rarely mentioned in the literature. In conclusion, the novel algorithm proposed in this study enables physicians and clinicians to identify risk factors for multiple diseases.

1. Introduction

Patients with cancer experience numerous side effects or complications as a result of chemotherapy, radiotherapy, surgery, or other treatments [1, 2]. Due to recent advances in new treatments, particularly for patients with breast cancer, the survival duration of patients has been significantly prolonged [3, 4]. However, breast cancer remains the most prevalent malignant neoplasm diagnosed among women worldwide and is the leading cause of death among female cancers [5–7]. In Taiwan, breast cancer is the most prevalent malignant neoplasm [8]. Aging is a high risk factor for breast cancer [9, 10], and menopause may also be considered a crucial risk factor for breast cancer [9, 11]. However, younger patients with breast cancer can receive (or tolerate) more treatments than older patients, which implies that younger patients are likely to experience more complications or side effects, such as fatigue, vomiting, and hair loss, which are common among women with breast cancer receiving chemotherapy or radiotherapy. In addition, some patients may exhibit more severe complications (or diseases), such as depressive disorder [12], osteoporosis [13], and fracture [14]. Therefore, patients with cancer may experience multiple side effects, complications, or diseases simultaneously. However, most published studies have investigated the risk factors associated with a single disease [12, 14, 15]; few have

investigated the risk factors associated with multiple diseases resulting from adjuvant treatments of breast cancer. This study developed a novel algorithm to determine the risk factors for multiple diseases, namely, depressive disorder, osteoporosis, and fracture, in young patients with breast cancer receiving curative surgery. Data in the Taiwan National Health Insurance Research Database (NHIRD) were employed by the algorithm.

2. Materials and Methods

2.1. Study Database. The study data were retrieved from a population-based claims database, the NHIRD, in Taiwan. The National Health Insurance (NHI) program in Taiwan was launched on March 1, 1995, and it provided health care coverage to more than 99% of the Taiwanese population in 2010 [16]. The NHIRD contains population-based health care information, including outpatient and inpatient clinic or hospital visits, dental service visits, and traditional Chinese medicine services. The diagnostic and medical procedures for diseases are based on the International Classification of Diseases, Ninth Revision, Clinical Modification (ICD-9-CM) and Procedure Coding System for every medical service claim.

2.2. Ethics Statement. The ethical review of the present study was approved by the Institutional Review Board of the School of Nursing, National Taipei University of Nursing and Health Sciences (CN-IRB-2011-063). In the NHIRD, the personal information of study patients is fully encrypted using double encryption procedures by the National Health Insurance Administration (NHIA). Because this study had a secondary database study design, written informed consent forms from the enrolled patients did not need to be obtained. The NHIA fully guarantees the confidentiality of the personal and health information of the study patients.

2.3. Study Population Selection Process. At the beginning, we retrieved from NHIRD from 2001 to 2007 by using the breast cancer ICD-9-CM code = 174.XX, and 73,385 patients were selected. Because we need to select newly diagnosed breast cancer patients, we need to exclude breast cancer patients who were diagnosed before January 1, 2002 ($n = 34{,}533$). Besides, we excluded death patients during 2001 to 2007 ($n = 1353$). We also excluded patients diagnosed with baseline depressive disorder (ICD-9-CM codes: 296.2X-296.3X, 300.4, and 311.X), bipolar disorder (ICD-9-CM codes: 296.0, 296.1, 296.4, 296.5, 296.6, 296.7, 296.8, 296.80, and 296.89), alcohol-use-related mental disorders (ICD-9-CM codes: V113, 9800, 2650, 2651, 3575, 4255, 3050, 291, 303, and 571.0–571.3), and osteoporosis (ICD-9-CM code: 733.XX) or with fracture history (ICD-9-CM code: 800.XX–829.XX), and male breast cancer patients ($n = 18$). There were 32,776 newly diagnosed breast cancer patients left. Because this study was aimed to investigate risk factors for complex diseases, including depressive disorder, osteoporosis, and fracture in young breast cancer patients receiving curative surgery, patients aged 20–39 years who had received a new diagnosis of breast cancer (ICD-9-CM code 174.XX) between January 1, 2001, and December 31, 2007, were further selected. All young patients with breast cancer who received curative surgery for the first time from 2001 to 2007 were finally recruited. Young patients with breast cancer who developed depressive disorder (ICD-9-CM codes: 296.2X-296.3X, 300.4, and 311.X), osteoporosis (ICD-9-CM code: 733.XX), and fracture (ICD-9-CM codes: 800.XX–829.XX) after curative surgery were enrolled. At the end of the selection process, a total number of 4108 young breast cancer patients were selected for this study. The patient recruitment scheme is presented in Figure 1.

2.4. Algorithm. The research database contains several categorical variables including binomial or multinomial variables. The first step of data analysis is to convert a data matrix containing categorical variables into a matrix containing index variables (0 or 1) through multiple correspondence analysis (MCA) [17]; the resultant matrix is called the Burt table, and each index variable indicates each level in all categorical variables. Furthermore, each index variable can be transformed into Euclidean coordinates in a higher dimensional space. Second, the resulting Euclidean coordinate matrix obtained through MCA can be considered as a high-dimensional data set or mixture distribution data set; the EM algorithm has been proven effective in determining hidden clusters among a high-dimensional or mixture distribution data set [18–20]. Third, after determining the optimal number of clusters, we used the v-fold cross-validation technique to identify the optimal clustering. The steps of the whole algorithm are detailed as follows:

Step 1 (MCA). Let $\mathbf{M}_{I\times K}$ be the raw data matrix with I as the subjects and k as the categorical variables.

(1) Convert the raw data matrix into the so-called Burt matrix:

   (i) If one categorical variable is the binary variable, let it be considered the original variable type in the Burt matrix.

   (ii) If one categorical variable has more than two levels (i.e., $\mathbf{J}_k > 2$ levels), convert this variable into a so-called indicator matrix, $\mathbf{I} \times \mathbf{J}_k$, in which each column contains binary variables coded 0 or 1.

   (iii) Place all the binary variable columns together to form the indicator matrix $\mathbf{X}_{\mathbf{I}\times\mathbf{J}}$.

   (iv) Generate the Burt matrix as = $\mathbf{X'X}$.

(2) Calculate the column and row coordinates:

   (i) The grand total of $\mathbf{M}_{I\times K}$ is $\mathbf{N}$; calculate the probability matrix as $\mathbf{P} = \mathbf{N}^{-1}\mathbf{X}$.

   (ii) Let $\mathbf{r}$ denote the vector of the row totals of $\mathbf{P}$ (i.e., $\mathbf{r} = \mathbf{P1}$, where $\mathbf{1}$ is a conformable vector comprised of 1's), and $\mathbf{c}$ denote the vector of the

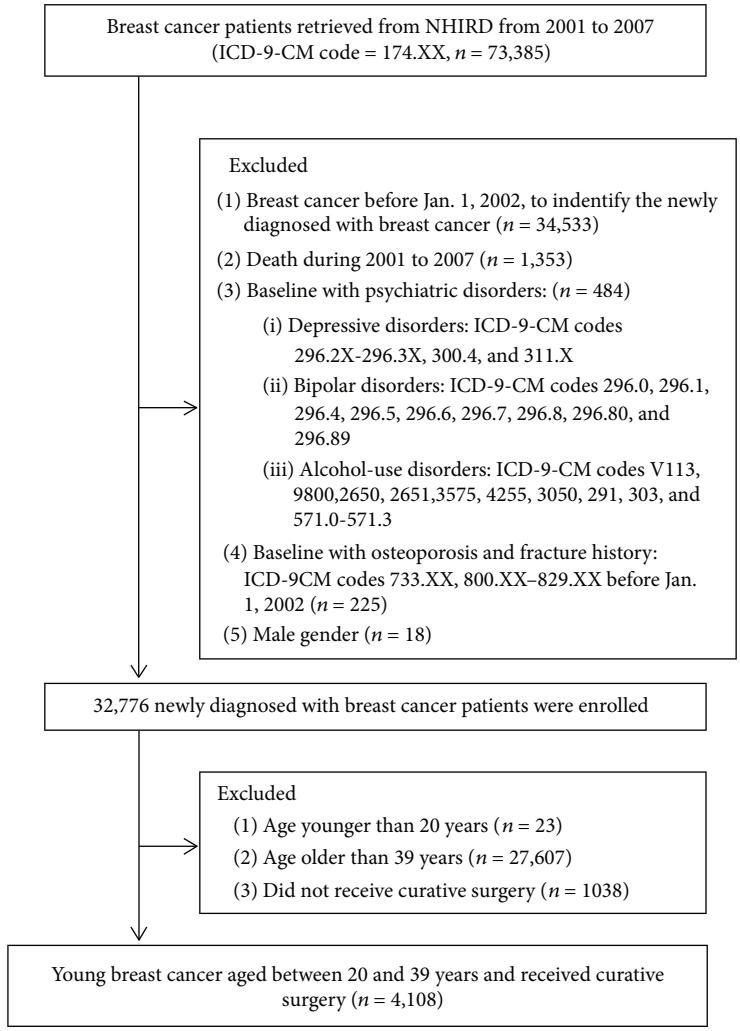


Figure 1: Recruitment scheme outlining patient selection.

column totals of **P**; $\mathbf{Dc} = \text{diag}\{\mathbf{c}\}$, and $\mathbf{Dr} = \text{diag}\{\mathbf{r}\}$.

(iii) Calculate the coordinate scores by using the singular value decomposition method:

$$D_r^{-1/2}\left(Z - rc^T\right)D_c^{-1/2} = P\Delta Q^T, \quad (1)$$

where Δ is the diagonal matrix of singular values, and $\Lambda = \Delta^2$ is the matrix containing eigenvalues. The so-called row coordinates and column coordinates are thus obtained as follows:

$$\begin{aligned}\text{row coordinate matrix} &= \mathbf{F} = D_r^{-1/2}P\Delta,\\ \text{column coordinate matrix} &= \mathbf{G} = D_c^{-1/2}Q\Delta.\end{aligned} \quad (2)$$

(3) Determine the number of dimensions by using inertia estimation:

(i) The Pearson chi-squared (χ^2) based distances from rows and columns to their respective point coordinate centers are calculated as

$$d_r = \text{diag}\left\{\mathbf{FF^T}\right\} \text{ and } d_c = \text{diag}\left\{\mathbf{GG^T}\right\}. \quad (3)$$

(ii) If we select a subset of **F** or **G**, the *inertia* for the row coordinates and column coordinates for each level is given by

$$\begin{aligned}\text{Inertia}_r &= \frac{\text{diag}\left\{\mathbf{FF'^T}\right\}}{N},\\ \text{Inertia}_c &= \frac{\text{diag}\left\{\mathbf{GG'^T}\right\}}{N},\end{aligned} \quad (4)$$

where $\mathbf{F}'$ and $\mathbf{G}'$ are the subsets of **F** and **G**.

Step 2. Expectation-maximization (EM) algorithm for clustering

(1) For a high-dimensional data set or mixture distribution data set, assuming that each variable in **F** or **G** is a random variable, the matrix can be modeled as a mixture normal-distribution probability density function as

$$\mathbf{A}_{\mu_i,\sigma_i}(x) = \frac{1}{\sigma_i\sqrt{2\pi}} e^{-(x-\mu_i)^2/2\sigma_i^2}, \tag{5}$$

where μ_i and σ_i are the mean and standard deviation for each variable in **F** or **A**.

(2) We use coordinates for each level of each categorical variable obtained using the MCA method, assuming that the number of dimensions selected using *Inertia*, which is previously obtained through MCA, is m.

(i) Combine m density functions to model a mixture distribution:

$$P_\theta(x) = \sum_{i=1}^{m} \alpha_i P_{\theta_i}(x), \qquad \theta = (\alpha_{1,\ldots,}\alpha_m, \theta_{1,\ldots,}\theta_m). \tag{6}$$

The Kth normal distribution is parametrized using

$$\theta_k = \{\mu_k, \sigma_k\}. \tag{7}$$

(ii) Given θ, the log-likelihood function of x is as follows:

$$\log P_\theta(x) = \log\left(\sum_{i=1}^{m} \alpha_i P_{\theta_i}(x)\right). \tag{8}$$

(3) For EM formulation, maximize the following function if θ' exists which can maximize the following:

$$\mathrm{Q}\left(\theta, \theta'\right) = \sum_y P_{\theta'_y}(y \mid x) \log\left(\alpha_y P_{\theta_y}(x)\right). \tag{9}$$

(i) E-step (expectation step): compute the likelihood of $y_i = k$.

$$\Lambda_{i,k} = P_{\theta'}(y_i = k \mid x_i) = \frac{\alpha'_k P_{\theta'_k}(x_i)}{P_{\theta'}(x_i)} = \frac{\alpha'_k P_{\theta'_k}(x_i)}{\sum_{j=1}^{m} \alpha_j P_{\theta'_j}(x_i)}. \tag{10}$$

(ii) M-step (maximization step): update α_k, μ_k, and σ_k for each categorical variable; for example, $k = 1, \ldots, m$:

$$\alpha_k = \frac{1}{n}\sum_{i=1}^{n} \Lambda_{i,k}, \qquad \mu_k = \frac{\sum_{i=1}^{n} x_i \Lambda_{i,k}}{\sum_{i=1}^{n} \Lambda_{i,k}}, \qquad \sigma_k^2 = \frac{\sum_{i=1}^{n} \Lambda_{i,k} \|x_i - \mu_k\|^2}{\sum_{i=1}^{n} \Lambda_{i,k}}. \tag{11}$$

Step 3. Apply v-fold cross-validation to the results of Step 2 to determine the optimal number of clusters

(i) Randomly divide the overall sample into a number of v-folds (in this study, we used $v = 10$).

(ii) All the levels of categorical variables with coordinates can be classified into $(v - 1)$-folds as the training sample and 1 testing sample, which can be applied to $i = 1, \ldots, v$th-fold.

(iii) The results for the v replications are aggregated using $-2^*\log(\textit{likelihood})$ as the measurement of clustering cost (the lower, the more favorable) and are shown in a scree plot for determining the optimal number of clusters.

The MY Structured Query Language was used for database preprocessing, which comprised extraction, linkage, and cleaning of NHIRD data in this study. After the database was processed based on the inclusion and exclusion criteria, the study data sets were obtained. All statistical analyses were performed using STATISTICA (version 10 for Windows; Statistica, Tulsa, OK, USA), and a two-tailed $p < 0.05$ was considered statistically significant.

3. Results

According to the recruitment scheme outlining patient selection (Figure 1), 4108 patients aged between 20 and 39 years who had received a diagnosis of breast cancer between January 1, 2001, and December 31, 2007, were recruited. The mean age of the patients was 34.6 years (standard deviation [SD] = 3.7 years). Of all the patients, 3.1% had depressive disorder, 0.7% had a fracture event, and 1.7% had osteoporosis. Regarding comorbidities, 1.7% patients had diabetes mellitus, 1.8% had hypertension, 0.7% had a history of heart failure, 0.5% had coronary heart disease, 0.4% had cerebrovascular disease, 2.3% had autoimmune disease, 0.9% had kidney disease, 0.4% had renal disease, 0.3% had liver cirrhosis, and 5.3% had chronic obstructive pulmonary disease (COPD). In addition, 13.4% exhibited distant metastasis and 6.9% exhibited primary metastasis. Regarding adjuvant therapies, 52.7% were receiving chemotherapy, 23.9% were receiving radiotherapy, 63.9% were receiving tamoxifen-related treatments, 7.5% were receiving aromatase inhibitor- (AI-) related treatments, and 3.2% were receiving trastuzumab treatments (Table 1).

After incorporating v-fold cross-validation into the EM clustering algorithm of the MCA coordinate matrix, four

Table 1: Demographic information of study sample ($n = 4108$).

Variable	Mean (SD)	n	%
Age (yrs)	34.6 (3.7)		
CCI	1.9 (3.0)		
Comorbidities:			
Diabetes mellitus (DM)		69	1.7
Hypertension		72	1.8
Heart failure		27	.7
Coronary heart diseases (CHD)		20	.5
Cerebrovascular diseases		18	.4
Autoimmune diseases		95	2.3
Kidney diseases		36	.9
Renal diseases		17	.4
Liver cirrhosis		12	.3
COPD		216	5.3
Distant metastatic		550	13.4
Primary metastatic		284	6.9
Adjuvant therapies:			
Chemotherapy		2165	52.7
Radiotherapy		983	23.9
Tamoxifen		2623	63.9
AIs		309	7.5
Trastuzumab		132	3.2
Outcome variables:			
Depression disorders		126	3.1
Fracture		31	.7
Osteoporosis		71	1.7

CCI: Charlson comorbidity index; COPD: chronic obstructive pulmonary disease; AIs: aromatase inhibitors.

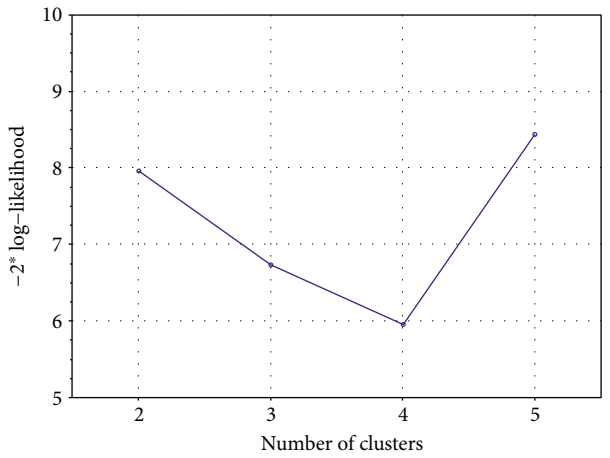


Figure 2: Graph of clustering cost sequence.

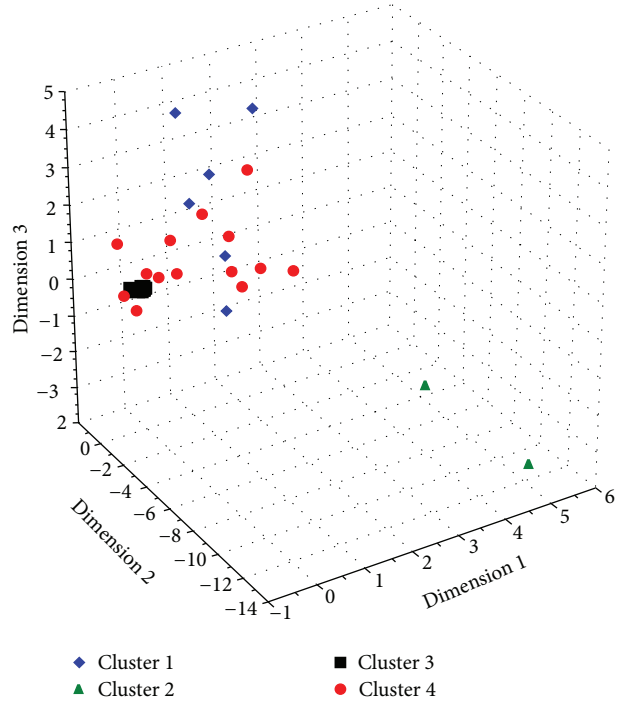
clusters were revealed to have the smallest −2*log-likelihood value; therefore, a four-cluster classification was defined as the optimal clustering approach (Figure 2).

Plots of each level of all variables in this study are presented in Figures 3(a) and 3(b). Figure 3(a) presents the four clusters obtained for all the points (levels of categorical variables in the study) in four colors without labels in a three-dimensional scatter plot, whereas Figure 3(b) has each point with labels. The three outcome variables—depressive disorder, fracture, and osteoporosis—were clustered with chemotherapy, radiotherapy, tamoxifen, AIs, trastuzumab, primary metastasis, distant metastasis, and comorbidities including liver cirrhosis and COPD (Figure 3(b)). The clustering results are also tabulated in the right column of Table 2, together with a comparison with two published studies.

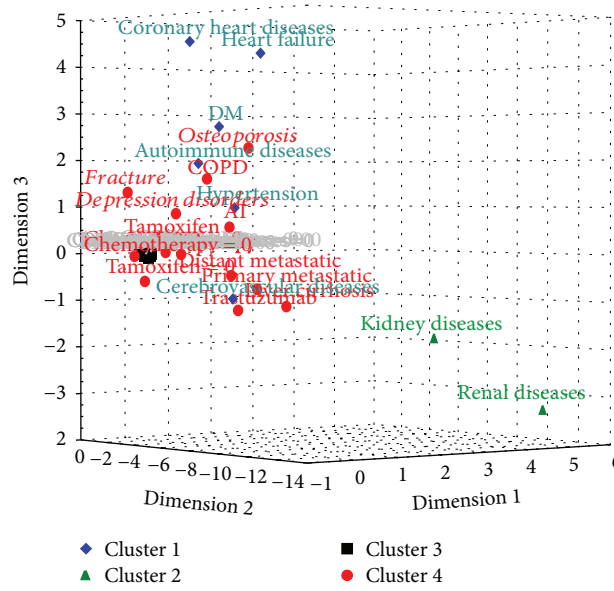
4. Discussion

The present study proposed a novel algorithm and used the NHIRD, a population-based database, to investigate risk factors for multiple diseases, namely, depressive disorder, osteoporosis, and fracture, among young patients with breast cancer who were receiving curative surgery. The study results revealed that depressive disorder, osteoporosis, and fracture were clustered with chemotherapy, radiotherapy, tamoxifen, AIs, and trastuzumab treatments; this finding is in agreement with the findings of other studies [12, 14]. In addition, depressive disorder, osteoporosis, and fracture were clustered with distant metastasis and primary metastasis. Depressive disorder was associated with distant and primary metastases; these associations were proposed in published studies [21, 22]. However, depressive disorder, osteoporosis, and fracture were also clustered with liver cirrhosis and COPD, which have rarely been investigated. Liver cirrhosis has been linked to increased levels of estrogen, which may be a risk factor for breast cancer [23]. In addition, a study from 1992 demonstrated that breast cancer may be associated with antigen CA-153 in liver cirrhosis [24]; however, more recently published articles have rarely mentioned this association. Studies on the association of COPD with depressive disorder, osteoporosis, and fracture among young patients with breast cancer have not been conducted in recent years; therefore, further investigation of the association of COPD with depressive disorder, osteoporosis, and fracture among young patients with breast cancer may be required.

The present study adopted a different approach to identifying risk factors for multiple diseases: we used "clustering" instead of typical statistical analysis methods (e.g., logistic regression and Cox regression). However, most studies have addressed univariate outcome variables (e.g., overall survival for death outcome, disease- (or progression-) free survival for recurrence of some disease, and binary outcome variable with/without some disease). Studies that have employed univariate statistical analysis methods can provide information on risk factors for some specific disease onset. However, patients with cancer usually experience multiple and complex symptoms or disease onsets, and studies or analysis methods addressing multiple concurrent diseases are still very limited. In this study, we proposed a novel algorithm that can analyze multiple outcome variables, and the findings provide clinical implications for clinicians that are not solely based on the results of univariate analyses.



(a) 3-dimensional scatter plot without variable labels



(b) 3-dimensional scatter plot with variable labels

Figure 3

The present study had some limitations. First, the NHIRD is a medical claims database, which does not include health behavior variables such as smoking behavior, alcohol consumption, lifestyle, and exercise, which may be associated with the risk of depressive disorder, osteoporosis, and fracture. Second, the NHIRD does not provide information on cancer staging, genetic mutations, or some environmental factors, which may be potential confounders associated with the risk of depressive disorder, osteoporosis, and fracture. Third, the patients enrolled in this study were mainly of Chinese ethnicity; thus, the study results derived from the proposed novel algorithm may not be generalizable to other ethnic populations.

In conclusion, the present study proposed a novel algorithm that can manage or cluster multiple disease outcomes with potential risk factors by using a large-scale population-based database. This novel algorithm can be straightforwardly applied to other diseases to help clinicians identify more potential risk factors if they plan to consider the potential risk factors associated with multiple disease outcomes.

Table 2: Present results compared with those of two other studies.

Variable	Chang et al. [12][a]	Chang et al. [14][b]	This study
Outcome variables:			
Depression disorders	✓		•
Fracture		✓	•
Osteoporosis			•
Comorbidities:			
Diabetes mellitus (DM)			□
Hypertension			□
Heart failure			□
Coronary heart diseases (CHD)			□
Cerebrovascular diseases			□
Autoimmune diseases			□
Kidney diseases			△
Renal diseases			△
Liver cirrhosis			•
COPD			•
Distant metastatic			•
Primary metastatic			•
Adjuvant therapies:			
Chemotherapy	*	*	•
Radiotherapy	*	*	•
Tamoxifen	*	*	•
AIs	*	*	•
Trastuzumab	*	*	•

[a][12]. [b][14]. *Statistically significant in studies; •, △, and □: implied variables were in the same cluster.

Abbreviations

MCA: Multiple correspondence analysis
EM: Expectation-maximization
SD: Standard deviation
NHIRD: National Health Insurance Research Database
NHIA: National Health Insurance Administration
COPD: Chronic obstructive pulmonary disease
AI: Aromatase inhibitor
MOHW: Ministry of Health and Welfare.

Data Availability

Due to the Personal Information Protection Act of Taiwan, the National Health Insurance Research Database (NHIRD) has been prohibited from releasing insuree's medical claim data via applications since June 28, 2016 (https://nhird.nhri.org.tw/apply_00.html). The Ministry of Health and Welfare (MOHW) of Taiwan decided that all NHIRD data analyses must be processed in the Health Data Science Center, which was established by the MOHW. Researchers who are interested in NHIRD research can seek access to the NHIRD by formal application (application website: https://dep.mohw.gov.tw/DOS/np-2497-113.html). All the authors of this paper understand and appreciate the need for data transparency in research and are ready to make the data available to those who have a research interest in accessing NHIRD data. Please direct your queries to NHIRD administrators Tze-Hui Wu (stcarolwu@mohw.gov.tw) or Zong-Ying Lin (st-zylin@mohw.gov.tw).

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

Chieh-Yu Liu was the principal investigator in charge of this study, conceptualizing the study design, applying the study database, coding the algorithm, conducting data analysis, drafting the initial manuscript, and critically reviewing and approving the final, submitted version of the manuscript. Chun-Hung Chang was the research assistant of this study, providing technological support for database maintenance and administrative support. All the authors read and approved the final manuscript.

Acknowledgments

The authors thank the Health Data Science Center of the Ministry of Health and Welfare of Taiwan for providing access to the National Health Insurance Research Database for data analysis in this study. This manuscript was edited by Wallace Academic Editing.

References

[1] C. McKeon, "Reducing the side effects of chemotherapy," *Australian Nursing Journal*, vol. 19, no. 8, p. 41, 2012.

[2] J. Xie, L. Xu, X. Xu, and Y. Huang, "Complications of peripherally inserted central catheters in advanced cancer patients undergoing combined radiotherapy and chemotherapy," *Journal of Clinical Nursing*, vol. 26, no. 23-24, pp. 4726–4733, 2017.

[3] R. K. Mittapalli, X. Liu, C. E. Adkins et al., "Paclitaxel–hyaluronic nanoconjugates prolong overall survival in a preclinical brain metastases of breast cancer model," *Molecular Cancer Therapeutics*, vol. 12, no. 11, pp. 2389–2399, 2013.

[4] S. Tilstra and M. McNeil, "New developments in breast Cancer screening and treatment," *Journal of Women's Health*, vol. 26, no. 1, pp. 5–8, 2017.

[5] T. Li, C. Mello-Thoms, and P. C. Brennan, "Descriptive epidemiology of breast cancer in China: incidence, mortality, survival and prevalence," *Breast Cancer Research and Treatment*, vol. 159, no. 3, pp. 395–406, 2016.

[6] A. Di Sibio, G. Abriata, D. Forman, and M. S. Sierra, "Female breast cancer in Central and South America," *Cancer Epidemiology*, vol. 44, pp. S110–S120, 2016.

[7] I. A. Aksenova, M. A. Moore, and A. S. Domozhirova, "Trends in breast cancer epidemiology in Chelyabinsk region, 2006-2015," *Asian Pacific Journal of Cancer Prevention*, vol. 18, no. 4, pp. 1163–1168, 2017.

[8] F. C. Liu, H. T. Lin, C. F. Kuo, L. C. See, M. J. Chiou, and H. P. Yu, "Epidemiology and survival outcome of breast cancer

in a nationwide study," *Oncotarget*, vol. 8, no. 10, pp. 16939–16950, 2017.

[9] N. F. Gomes-Rochette, L. S. Souza, B. O. Tommasi et al., "Association of PvuII and XbaI polymorphisms on estrogen receptor alpha (ESR1) gene to changes into serum lipid profile of post-menopausal women: effects of aging, body mass index and breast cancer incidence," *PLoS One*, vol. 12, no. 2, article e0169266, 2017.

[10] M. F. Barginear, H. Muss, G. Kimmick et al., "Breast cancer and aging: results of the U13 conference breast cancer panel," *Breast Cancer Research and Treatment*, vol. 146, no. 1, pp. 1–6, 2014.

[11] R. K. C Yuen, D. Merico, M. Bookman et al., "Whole genome sequencing resource identifies 18 new candidate genes for autism spectrum disorder," *Nature Neuroscience*, vol. 20, no. 4, pp. 602–611, 2017.

[12] C. H. Chang, S. J. Chen, and C. Y. Liu, "Adjuvant treatments of breast cancer increase the risk of depressive disorders: a population-based study," *Journal of Affective Disorders*, vol. 182, pp. 44–49, 2015.

[13] F. A. Trémollieres, I. Ceausu, H. Depypere et al., "Osteoporosis management in patients with breast cancer: EMAS position statement," *Maturitas*, vol. 95, pp. 65–71, 2017.

[14] C. H. Chang, S. J. Chen, and C. Y. Liu, "Fracture risk and adjuvant therapies in young breast cancer patients: a population-based study," *PLoS One*, vol. 10, no. 6, article e0130725, 2015.

[15] C. H. Chang, S. J. Chen, and C. Y. Liu, "Risk of developing depressive disorders following hepatocellular carcinoma: a nationwide population-based study," *PLoS One*, vol. 10, no. 8, article e0135417, 2015.

[16] M. J. Yeh and H. H. Chang, "National Health Insurance in Taiwan," *Health Affairs*, vol. 34, no. 6, p. 1067, 2015.

[17] D. Ayele, T. Zewotir, and H. Mwambi, "Multiple correspondence analysis as a tool for analysis of large health surveys in African settings," *African Health Sciences*, vol. 14, no. 4, pp. 1036–1045, 2014.

[18] S. N. Kadir, D. F. M. Goodman, and K. D. Harris, "High-dimensional cluster analysis with the masked EM algorithm," *Neural Computation*, vol. 26, no. 11, pp. 2379–2394, 2014.

[19] G. J. McLachlan and P. N. Jones, "Fitting mixture models to grouped and truncated data via the EM algorithm," *Biometrics*, vol. 44, no. 2, pp. 571–578, 1988.

[20] G. Govaert and M. Nadif, "An EM algorithm for the block mixture model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 4, pp. 643–647, 2005.

[21] X. Guo, J. Xu, Y. E, Z. Yu, and T. Sun, "Correlation between hormone receptor status and depressive symptoms in patients with metastatic breast cancer," *Oncotarget*, vol. 8, no. 31, pp. 50774–50781, 2017.

[22] A. S. Keuroghlian, L. D. Butler, E. Neri, and D. Spiegel, "Hypnotizability, posttraumatic stress, and depressive symptoms in metastatic breast cancer," *The International Journal of Clinical and Experimental Hypnosis*, vol. 58, no. 1, pp. 39–52, 2010.

[23] H. T. Sorensen, S. Friis, J. H. Olsen et al., "Risk of breast cancer in men with liver cirrhosis," *The American Journal of Gastroenterology*, vol. 93, no. 2, pp. 231–233, 1998.

[24] J. Collazos, J. Genolla, and A. Ruibal, "Breast cancer-associated antigen CA 15.3 in liver cirrhosis," *Acta Oncologica*, vol. 31, no. 7, pp. 741–744, 1992.

Hindawi
Complexity
Volume 2018, Article ID 4963139, 13 pages
https://doi.org/10.1155/2018/4963139

Research Article

A Scalable Genetic Programming Approach to Integrate miRNA-Target Predictions: Comparing Different Parallel Implementations of M3GP

Stefano Beretta,[1,2] **Mauro Castelli**,[3] **Luis Muñoz**,[4] **Leonardo Trujillo**,[4] **Yuliana Martínez**,[4] **Aleš Popovič**,[3,5] **Luciano Milanesi**,[2] **and Ivan Merelli**[2]

[1]*DISCo, Università degli Studi di Milano-Bicocca, Milan, Italy*
[2]*Istituto di Tecnologie Biomediche, Consiglio Nazionale delle Ricerche, Segrate, Italy*
[3]*NOVA Information Management School (NOVA IMS), Universidade Nova de Lisboa, Campus de Campolide, 1070-312 Lisboa, Portugal*
[4]*Tree-Lab, Posgrado en Ciencias de la Ingeniería, Instituto Tecnológico de Tijuana, Tijuana, BC, Mexico*
[5]*Faculty of Economics, University of Ljubljana, Kardeljeva Ploščad 17, SI-1000 Ljubljana, Slovenia*

Correspondence should be addressed to Mauro Castelli; mcastelli@novaims.unl.pt

Received 28 February 2018; Accepted 29 May 2018; Published 4 July 2018

Academic Editor: Dimitrios Vlachakis

There are many molecular biology approaches to the analysis of microRNA (miRNA) and target interactions, but the experiments are complex and expensive. For this reason, in silico computational approaches able to model these molecular interactions are highly desirable. Although several computational methods have been developed for predicting the interactions between miRNA and target genes, there are substantial differences in the results achieved since most algorithms provide a large number of false positives. Accordingly, machine learning approaches are widely used to integrate predictions obtained from different tools. In this work, we adopt a method called multidimensional multiclass GP with multidimensional populations (M3GP), which relies on a genetic programming approach, to integrate and classify results from different miRNA-target prediction tools. The results are compared with those obtained with other classifiers, showing competitive accuracy. Since we aim to provide genome-wide predictions with M3GP and, considering the high number of miRNA-target interactions to test (also in different species), a parallel implementation of this algorithm is recommended. In this paper, we discuss the theoretical aspects of this algorithm and propose three different parallel implementations. We show that M3GP is highly parallelizable, it can be used to achieve genome-wide predictions, and its adoption provides great advantages when handling big datasets.

1. Introduction

MicroRNAs (miRNAs) are approximately 22-nucleotide-long, single-stranded RNA molecules encoded in the genomes of plants, animals, and viruses and are capable of interfering with intracellular messenger RNAs (mRNA) [1]. miRNAs are key regulators of gene expression at the posttranscriptional level, but the precise mechanisms underlying their interactions with the respective gene targets are still poorly understood. The effect of the hybridization between a miRNA and its target mRNA is that the expression of the protein coded by the gene is silenced, either by stopping the translation process or by marking the mRNA for degradation.

Since miRNAs are involved in the onset of many different diseases, the study of their interactions with the genome is very important. For instance, several recent reports suggest that miRNA aberrations may be an important factor in the development of cancer [2, 3]. Another study demonstrated that more than 50% of miRNA targeted genes are located in

cancer-associated genomic regions or in fragile sites [4], indicating that miRNAs may play an important role in the pathogenesis of many human diseases.

It is also a challenging problem to identify miRNAs on an experimental basis due to their limited expression, which arises from the dynamic behavior of the regulation process and the tissue specificity of their control mechanism. Moreover, taking into account the different mechanisms by which miRNAs exert their role, only the absence of the protein in miRNA-transfected cells represents definitive proof of the miRNA-target interaction. This technique is very complex and costly to achieve, thereby imposing serious constraints on the number of experiments that can be performed.

Therefore, computational predictions represent a very important approach for screening possible targets to be experimentally tested. More precisely, the interactions between a miRNA and its mRNA target sites can be considered from thermodynamic, probabilistic, and evolutionary (or sequence-based) points of view. Several computational tools for predicting miRNA-target sites have been developed in recent years using one or more of the aforementioned aspects [5].

Many works have been done to compare their performance (see [6–8] for details), taking into account the most famous tools: PITA [9], miRSystem [10], miRmap [11], DIANA-microT-CDS [12], CoMir [13], mirWalk [14], and PicTar [15].

Among the most well-known prediction tools for miRNA-target recognition, three of the most adopted ones are miRanda [16, 17], TargetScan [18, 19], and RNAhybrid [20]. miRanda completes three sequential steps: (i) sequence matching to find the maximal local complementarity between a mature miRNA and the putative target site, (ii) free energy calculation to estimate the strength of a potential RNA duplex, and (iii) filtering of predicted targets on the basis of evolutionary conservation. TargetScan is based on two hypotheses: (i) highly conserved miRNAs are more involved in regulation and (ii) membership in large miRNA families leads to a higher number of existing targets. After the matching step (allowing wobble pairs and stopping at the first mismatch encountered), a thermodynamic evaluation of the RNA duplex is performed. Finally, RNAhybrid predicts the target genes based on free energy calculation. It collects the most favorable energetic structures, normalizes them, and then uses estimated p values to determine the significance of each predicted binding site.

We decided to focus on these three tools since, as described before, they employ different approaches to predict the interactions, such as sequence matching, thermodynamic evaluation of the RNA duplex, or free energy calculation. This aspect is fundamental since one of the main goals of this work is to combine results obtained with different approaches. Finally, we would like to point out that some of the other available tools base their prediction on the output of one of the three tools we considered, or employ a similar technique.

Although common guidelines are adopted by the aforementioned methods, differences in the definition of the physical models (and uncertainties concerning the real biological mechanisms) and differences in the formalization of the corresponding algorithms (and implementations) result in quite different miRNA-target predictions. Moreover, by comparing computational results with experimental validations, we can see that these tools produce a large number of false-positive predictions.

The lack of a clear consensus on the predictions achieved by these tools has led to the development of methods to perform meta-analyses of the results by integrating lists of miRNA-target genes predicted by several algorithms. Among such integrated tools, the most used ones are miRGator [21] and ExprTarget [22]. These tools exploit functional analyses and genome annotations to better characterize the identified targets. miRGator also provides miRNA expression profiles by importing expression experiments from the Gene Expression Omnibus databank [23]. Analogously, expression profiles are reported in mESAdb [24] and mirEX [25]. MAGIA [26] returns predictions as unions or intersections of results produced by TargetScan, miRanda, and RNAhybrid. Moreover, it integrates mRNA expression values with miRNA expression scores in order to elucidate inverse correlations, thus hypothesizing about new miRNA-target associations. myMIR [27] implements a pipeline for computing ranked miRNA-target lists, integrating predictions from different tools. This approach also provides functional annotations for characterizing genes targeted by each miRNA, highlighting overrepresented ontological terms.

In a previous work [28], we described the application of a classification technique based on genetic programming, called M3GP, to integrate results from three different miRNA-target prediction tools. More precisely, we considered this as a classification problem and started from a set of positive and negative examples used to train the adopted method. Although the M3GP method has been developed to improve the performance on multiclass problems, it has also been shown to achieve good results with binary problems. The idea behind this method is to improve the standard genetic programming technique in which a population of candidate solutions is evolved by applying genetic operators until an ending criterion is reached. In particular, the idea of M3GP is to define a function that transforms the input data by mapping them into another feature space. The objective of M3GP is to evolve transformations of the input data in such a way that it becomes easier to perform the classification task in the new feature space of the problem.

To assess the performance of the M3GP method on this classification problem, we compared the obtained results with those achieved by other methods which were applied to the same data, showing that M3GP always achieves good accuracy. Another key point relates to the type of models evolved by M3GP relative to other GP techniques. In M3GP, the models are composed of a set of independent subtrees allowing for the individuals to be evaluated in parallel to each other, an approach that is successfully applied in this work.

Indeed, considering the high number of species to analyze (this mechanism can be studied in several species for which the genome is already available), the number of miRNAs identified (in humans, the total is about 2,000), and the number of genes to study (in humans about

20,000), a good strategy to parallelize M3GP is essential. In this work, we discuss some important aspects of this algorithm, proposing three different parallel implementations. In summary, we present the following contributions. First, we show that M3GP can perform competitively in predicting interactions between miRNA and targets. In this task, the advantage of our method is that it does not overfit the training data and presents a minimal amount of variance over multiple runs. In other words, M3GP appears to be quite robust in this domain. Second, from an implementation perspective, we show that the method is highly parallelizable, particularly given the manner in which the output is constructed. The results show that when the search contains large populations with large multidimensional individuals, the best strategy is to parallelize the evaluation of each individual transformation.

The paper is organized as follows: Section 2 provides a description of the data used in the experimental phase where the performance of the classification algorithm in integrating the predictions of miRNA-target sites was evaluated. Section 3 presents the M3GP algorithm and describes its properties. Section 4 discusses the obtained results, also taking other state-of-the-art machine learning techniques into account. Section 5 presents the parallel implementations of the M3GP system, with a detailed discussion of the performance achieved. Section 6 concludes the paper and suggests possible avenues for future research.

2. Background

In this section, we describe the datasets used in this work and how we obtained the data on which the proposed method was tested. We started by downloading from the TargetScan website (http://www.targetscan.org/) the sequences of the miRNA families and of the untranslated regions (UTRs, the genomic loci targeted by miRNA) from 23-way alignment (We have used Release 6.1 of November 2011 to avoid problems in the names of the miRNA sequences with respect to those present in the other datasets we considered for the analysis.) More precisely, the TargetScan database contains the miRNA sequences obtained from miRBase (with the additional information about the miRNA families) and the 3UTR sequences obtained from the UCSC. We decided to use the data from TargetScan to be able to match the identifiers (of both miRNA and UTR sequences) with those used in the study we employed in our experimental analysis. Subsequently, we filtered the information relative to the *Homo sapiens* species and obtained a total of 30,887 UTR and 1,558 miRNA sequences, which were used as the starting point of our analysis.

To identify the miRNA-target interactions, we used three target prediction tools: miRanda [16, 17], TargetScan [18, 19], and RNAhybrid [20]. The results produced by these tools were combined in a matrix; by looking at the positions on the UTR of each predicted interaction and, for each of them, we considered the score corresponding to the obtained prediction. Moreover, to handle the missing predictions of some tools, we replaced the missing values with Not-a-Number (NaN). To deal with such values during the experiments performed, we assigned penalizing scores to them. About the missing data, we conducted some experiments to assess how the choice of the penalization influences the final classification results. Results of this preliminary analysis showed that the value of the penalization score only marginally affects the final results. The overall matrix was composed of 48,121,946 elements (30,887 UTRs $\times$ 1,558 miRNAs) and 3 columns corresponding to the scores of the considered prediction tools (in addition to the other columns containing information about the miRNA-target interaction).

As discussed in [29], one problem when using machine learning methods to address the prediction of miRNA-target interactions is the lack of negative examples or miRNA nontarget pairs. In fact, since having a good training set is crucial when applying these techniques for solving classification problems, the current approaches tend to randomly generate sequences to be used as negative examples that, by the way, could be unrealistic. For this reason, we decided to adopt the results proposed in [29] to populate the dataset used to train the proposed classifier.

In that work, the authors generated two sets of positive and negative miRNA-target examples. The former set (positive examples) was obtained by biologically verified experiments, while the latter examples (negative) were identified from a pooled dataset of predicted miRNA-target pairs. More precisely, the authors selected a set of computationally predicted (by one or more algorithms) targets of miRNA, measuring the tissue specificity for both of them. Then, significantly overexpressed miRNA-mRNA pairs were selected as potential negative examples, and a further expression profiling test was performed to discard those that did not pass this filter. Finally, the thermodynamic stability and seed-site conservation were measured to infer the final negative examples. The downloaded dataset from [29] is composed of 288 (Out of the 289 positive examples available in the dataset, we were not able to find a match for one of them, which was discarded.) positive examples and 286 negative examples of miRNA-target interactions.

To further increase the set of positive examples in the dataset, we downloaded the data from miRTarBase [30], which is a database of experimentally validated miRNA-target interactions. In this way, it was possible to classify the positive examples into positive and experimentally validated examples, only experimentally validated examples, and only positive examples. More specifically, we crossed miRTarBase interactions with the positive examples from [29], in order to make the dataset more robust. We thus labelled the positive examples with three different labels (although the final classification problem is binary), which are (1) for the positive examples from [29] that were also present in the miRTarBase, (2) for the (positive) interactions only found in the miRTarBase, and (3) for those only present in the positive examples from [29], while we labelled with 1 the negative examples (from [29]). In this way, in the binary classification problem encountered in this work, we selected two balanced sets of elements: the first one among the negative examples was labelled with 1, while the second one among those labelled with 1, 2, and 3 since all of them

represent "positive examples" (from [29], miRTarBase, or both). To obtain the final matrix, we combined the results of the prediction tools with these classification labels and, for those predicted interactions that are neither negative nor positive (in any of the three sets), we added the label 0. Finally, we refined these results by eliminating redundancies in the two classes of examples which are caused by notational problems (i.e., the same mRNA identified multiple times). Table 1 reports the number of elements in the matrix for each classification type.

Note that the *unknown examples* are elements of the matrix for which we do not have a priori knowledge regarding whether the interaction is true or not. For this reason, these examples could be used to identify new interaction candidates to be biologically validated; that is, those classified as positive by the proposed method could represent novel miRNA-mRNA interactions.

Table 1: Description of the types of samples in our dataset.

Classification	Number of elements
Negative examples	286
Positive and exp. validated examples	179
Only exp. validated examples	286
Only positive examples	6376
Unknown examples	48114996
Total	48121946

3. Method

In this work, we address a binary classification problem in which the two classes refer to the examples described in Section 2. More precisely, the first class refers to negative examples, while the second class refers to positive ones. To simplify the classification process, we take an equal number of elements from both classes (for the positive ones, we combined the three classes of positive examples of Table 1), defining a balanced classification task that is easier to handle for most machine learning classifiers. Figure 1 presents different views of the distribution of all samples used from each class, showing three 2D views, one for each pair of features (prediction tools). As it is possible to notice from these scatter plots, the problem basically shows three clusters of data, but in each of them the classes are multimodal and overlap within the feature space. This view clearly shows that the problem of distinguishing the classes is quite difficult.

In addition to M3GP, we executed several other standard classifiers from the machine learning literature to solve this problem (in order to evaluate the proposed method's performance with respect to the state-of-the-art techniques), as follows:

(1) Euclidean distance classifier (ED)

(2) Mahalanobis distance classifier (MD)

(3) Naive Bayes Classifier (NB);

(4) Support vector machine (SVM) (with Gaussian radial basis function kernel and a default scaling factor of 1)

(5) K-nearest neighbor (KNN) (using $K = 5$ neighbors)

(6) Treebagger classifier (TREE) (using 10 trees)

(7) Multidimensional multiclass genetic programming classifier with multidimensional populations (M3GP) [31, 32]

ED and MD are linear classifiers that assume a Gaussian unimodal distribution for each class, an assumption that is clearly violated in this problem as seen in Figure 1. KNN and NB are well-known and widely used classifiers, mainly for their simplicity, ease of implementation, and competitive performance. However, the NB classifier assumption that each feature dimension can be treated independently is often violated in practice (although this is not the case in the problem studied here). SVM and TREE are state-of-the-art methods that perform strongly in many domains. In particular, the TREE classifier can handle multimodal classes and unbalanced datasets.

3.1. Multidimensional Multiclass GP with Multidimensional Populations: M3GP. In this section, we introduce the method we previously developed and apply it to the problem of integrating the results of different prediction tools. More precisely, M3GP uses a GP-based search to achieve competitive performance on multiclass problems but also achieves strong results in binary tasks [32].

M3GP is based on the multiclass GP with multidimensional populations (M2GP) [31] that searches for a transformation, which is applied to the input data so that the transformed data of each class can be grouped into unique clusters and thus simplify the classification task. In M2GP, the number of dimensions in which the clustering process is performed is completely independent of the number of classes, such that high-dimensional datasets can be easily classified by a low-dimensional clustering, while low-dimensional datasets may be better classified by a high-dimensional clustering.

In order to achieve this, M2GP uses a representation of the solutions that allows performing, for each data point x, the mapping $k(\mathbf{x})$: $\mathbb{R}^p \rightarrow \mathbb{R}^d$, from an input feature space of p dimensions to a new one. The representation is basically the same used for regular tree-based GP, except that the root node of the tree exists only to define the number of dimensions d of the new space. Each branch stemming directly from the root performs the mapping in one of the new d dimensions. In M2GP, candidate solutions are evaluated as follows:

(1) All the p-dimensional samples of the training set are mapped to the new d-dimensional space (each branch of the tree is one of the d dimensions).

(2) In this new space, for each of the M classes in the data, the covariance matrix and the cluster centroid are calculated from the samples belonging to that class.

(3) the *Mahalanobis* distance between each sample and each of the M centroids is calculated. Each sample

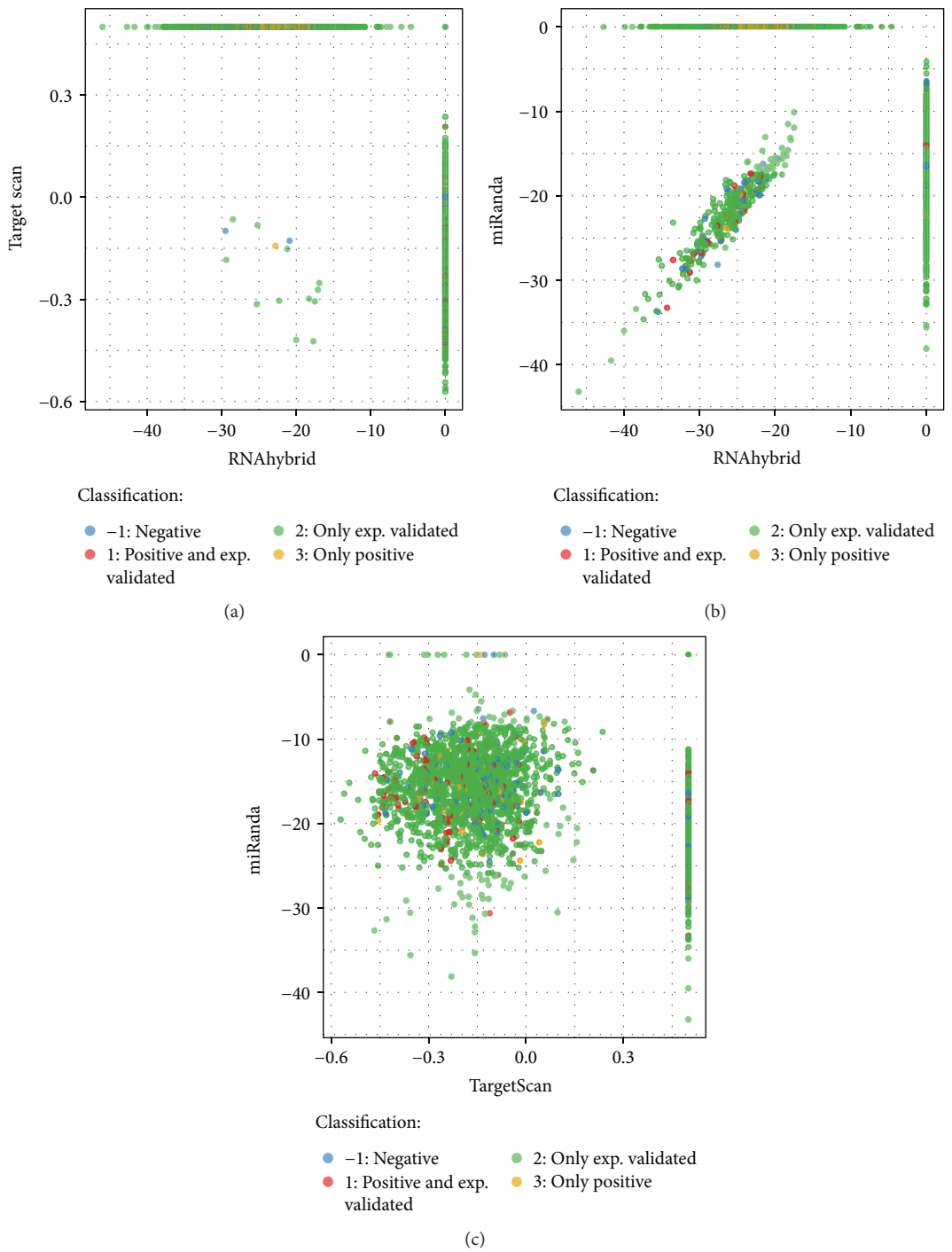


Figure 1: Scatter plots showing the distribution of the data: negative examples are represented by blue dots, while three different colors are used for positive examples. More precisely, red are those samples that are among the positive examples in the dataset from [29] and that are also experimentally validated, green for those only experimentally validated in miRTarBase, and yellow for the ones that are only in the positive examples of the dataset from [29].

is then assigned to the class having the closest centroid. Finally, the fitness function is given by the classification accuracy.

The original M2GP uses a greedy approach to determine how many dimensions the evolved solutions should have. It may happen that, by fixing the number of dimensions at the beginning of the run, the algorithm will be unable to find the best solutions during the search process with respect to those that may be found by using a different number of dimensions. Therefore, in the M3GP approach, a population that may contain transformation of different dimensions is evolved.

During the breeding phase, whenever the chosen genetic operator is a mutation, one of the three following actions is

performed with equal probability: (i) a standard subtree mutation, where a randomly created new tree replaces a randomly chosen branch (excluding the root node) of the parent tree; (ii) a randomly created new tree is added as a new branch of the root node, effectively adding one dimension to the parent tree; and (iii) a complete branch of the root node is randomly removed, effectively removing one dimension from the parent tree.

On the other hand, whenever the chosen genetic operator is a crossover, one of the two following actions is performed with equal probability: (i) a standard subtree crossover, where a random node (excluding the root node) is chosen in each of the parents, and the corresponding branches are swapped, and (ii) the swapping of dimensions, where a random complete branch of the root node is chosen in each parent, and swapped between each other, effectively swapping dimensions between the parents. The latter event is just a particular case of the first one, where the crossing nodes are guaranteed to be directly connected to the root node.

M3GP also adds a pruning procedure that removes a random dimension and reevaluates the tree. If the fitness improves, the pruned tree replaces the original one and the procedure goes through the pruning of another dimension. Otherwise, the pruned tree is discarded and the original tree goes through the pruning of a new dimension. The procedure stops after each of the original dimensions was pruned and tested for improvement.

4. Results

As stated above, the problem is posed as a balanced binary classification task where the data for both classes were taken from the sets of positive and negative examples described in Section 2, by eliminating the duplicated entries. More precisely, 212 data samples were selected to represent class 1 (of negative examples labelled with 1) and 212 data samples were selected to represent class 2 (of positive examples, from those labelled with 1, 2, and 3). Subsequently, each of the selected methods for the performance comparison (i.e., ED, MD, NBC, SVM, KNN, TREE, and M3GP) was executed. The whole dataset was split into two partitions, a training set composed of 70% of the data and a test set with the remaining 30%. For each classifier, 30 independent runs were performed using a different random partition of the data in each run.

For M3GP, we used the parameter values specified in Table 2, following the indication of [32]. These values have been shown to produce accurate results, and since a preliminary tuning phase (performed with a grid search) showed no (statistically significant) change in terms of performance with respect to other configurations tested, we relied on these values. This is related to the fact that it has become increasingly clear that GP is very robust to parameter values once a good configuration is determined, as suggested by a recent and in-depth study [33].

Figure 2 provides a detailed view of the results by showing a comparison of the boxplots. In detail, Figure 2(a) shows the error obtained by the classifiers on the training data, while Figure 2(b) shows the error on the test set.

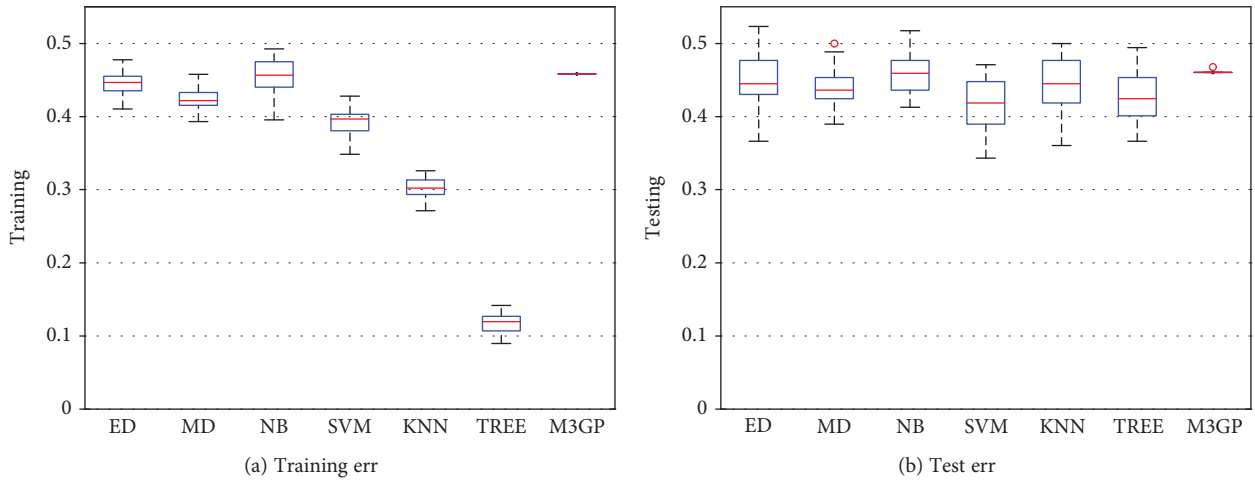
Table 2: Parameters used by M3GP to obtain the results reported in Figure 2.

Parameter	Value
Runs	30
Population size	500 individuals
Generations	100
Initialization	6-depth full initialization with 1 initial dimension
Operator probabilities	Crossover $p_c = 0.5$, mutation $p_\mu = 0.5$
Function set	$(+, -, \times, \div$ protected as in [34])
Terminal set	Ephemeral random constants [0,1]
Bloat control	17-depth limit
Selection	Lexicographic tournament of size 5
Elitism	Keep best individual

Again, as anticipated in Section 3, these results confirm that all the considered classifiers achieve a very similar performance on unseen data, even if the training performance is better for some of them. In fact, we can state that SVM, KNN, and TREE overfit the training instances and, thus, their training performance is a poor indicator of performance on unseen instances. To statistically validate the results obtained, statistical comparisons are carried out using a $1 \times N$ formulation where a single control method (M3GP) is compared with N algorithms. We use the Friedman test and the Bonferroni-Dunn correction of the p values for each comparison. In all tests, the null hypothesis (that the medians are equal) is rejected at the $\alpha = 0.05$ significance level. According to the statistical validation, SVM and TREE are the techniques that are able to outperform M3GP on unseen instances. Specifically, the p values returned by the Friedman test are 0.0001 and 0.0016, respectively. All of the remaining techniques perform similarly than M3GP.

While SVM and TREE produce a better performance with respect to M3GP, it is important to highlight an advantage provided by M3GP with regard to the former methods. Specifically, M3GP is able to produce a human-understandable model, combining the problem features in a tree structure that can be interpreted by a domain expert, hence allowing for a better understanding of the relations between the features of the problem. This is a typical property of the large majority of GP-based systems. On the other hand, SVM does not have this property, providing a model that expresses the equation of the hyperplane that divides the two classes of the problem. TREE provides an ensemble of trees, and this makes it inherently different from the other techniques considered. For this technique, it is even difficult to identify what the final model is. For all these reasons, we believe that M3GP is a useful technique for addressing classification problems with two or more classes.

Although the results achieved by the tested methods may look similar, which was expected due to the difficulty of the problem (see Figure 1 for details of the distribution of the input data), we can highlight some aspects of the adopted approach. First of all, M3GP does not seem to be affected by overfitting or a lack of generalization after learning, with



(a) Training err (b) Test err

Figure 2: Boxplot reporting the errors of the tested classifiers in the both training (a) and test (b) sets.

an almost equal training and testing performance. Second, the method seems to be quite robust to the training data used, exhibiting the least amount of variance of all the methods. This result is particularly surprising since M3GP is a stochastic search procedure, like most other evolutionary algorithms. Indeed, a common feature of such methods is that their performance tends to vary from run to run, especially when different data partitions are used. This does not happen to M3GP in this domain, making the method an interesting and promising choice for the considered problem given that there can be a lot of noise in the data collection process.

4.1. M3GP Parallelization. Like most population-based search methods, and given the manner in which M3GP operates, the algorithm can be easily implemented in a parallel way by using different methods [35]. However, probably the most direct way to parallelize the algorithm is to evaluate each candidate solution in parallel since the fitness computation of each individual is independent, and this is referred to as the *global model* [36]. In the case of M3GP, this includes the data transformation stage and the results of the MD classification step.

Another approach is to change the basic GP paradigm, which is mostly used as a synchronous and local algorithm. For instance, it is possible to use the evolutionary model proposed in [37] to distribute the evolutionary process across various client machines (either locally or over the web) in a plug-and-play manner, thus increasing the available computational resources by exploiting cloud-based technologies. However, this option is left for possible future work.

Moreover, in the particular case of M3GP, there is another parallelization option. Since the individuals are composed of several independent subtrees joined by the "dummy" root node, it is quite easy to parallelize the evaluation of each individual subtree (dimension). In other words, each feature dimension in an M3GP individual can be evaluated independently.

In summary, three different parallel implementations of M3GP were developed and tested in this work (see Figure 3):

(i) Parallel population evaluation (*Pop Parallel*): this implementation parallelizes the evaluation of each solution as a whole, sending to each parallel worker independent solutions to process. This is a very common approach for evolutionary algorithms where the fitness evaluation of the population is divided to reduce the computational time.

(ii) Parallel fitness evaluation (*Fitness Parallel*): this implementation parallelizes the evaluation of each subtree branch on the main root node, sending individual independent branches (features) to parallel workers. Thus, the evaluation of M3GP is parallelized based on the number of feature dimensions for each candidate solution.

(iii) Full parallel M3GP (*Full Parallel*): in this case, the two previous approaches are combined.

5. Discussion

Before describing the experimental analysis we performed to compare the three different implementations, we discuss some important aspects regarding each. More precisely, we want to highlight advantages and differences of the different approaches, also with respect to the *Non-Parallel* approach, from a computational complexity point of view. In particular, we focus on how the evaluation process can be parallelized. In the following analysis, we consider the computational cost of a single generation and for a fixed initial configuration of the algorithm parameters (see Table 3). The reason for this choice is that we want to focus on the differences in the described parallel approaches, discarding all operations that do not vary among them in our analysis.

The first approach, which is referred to as *Pop Parallel*, consists of evaluating each individual of the current population independently of the others and then joining the results

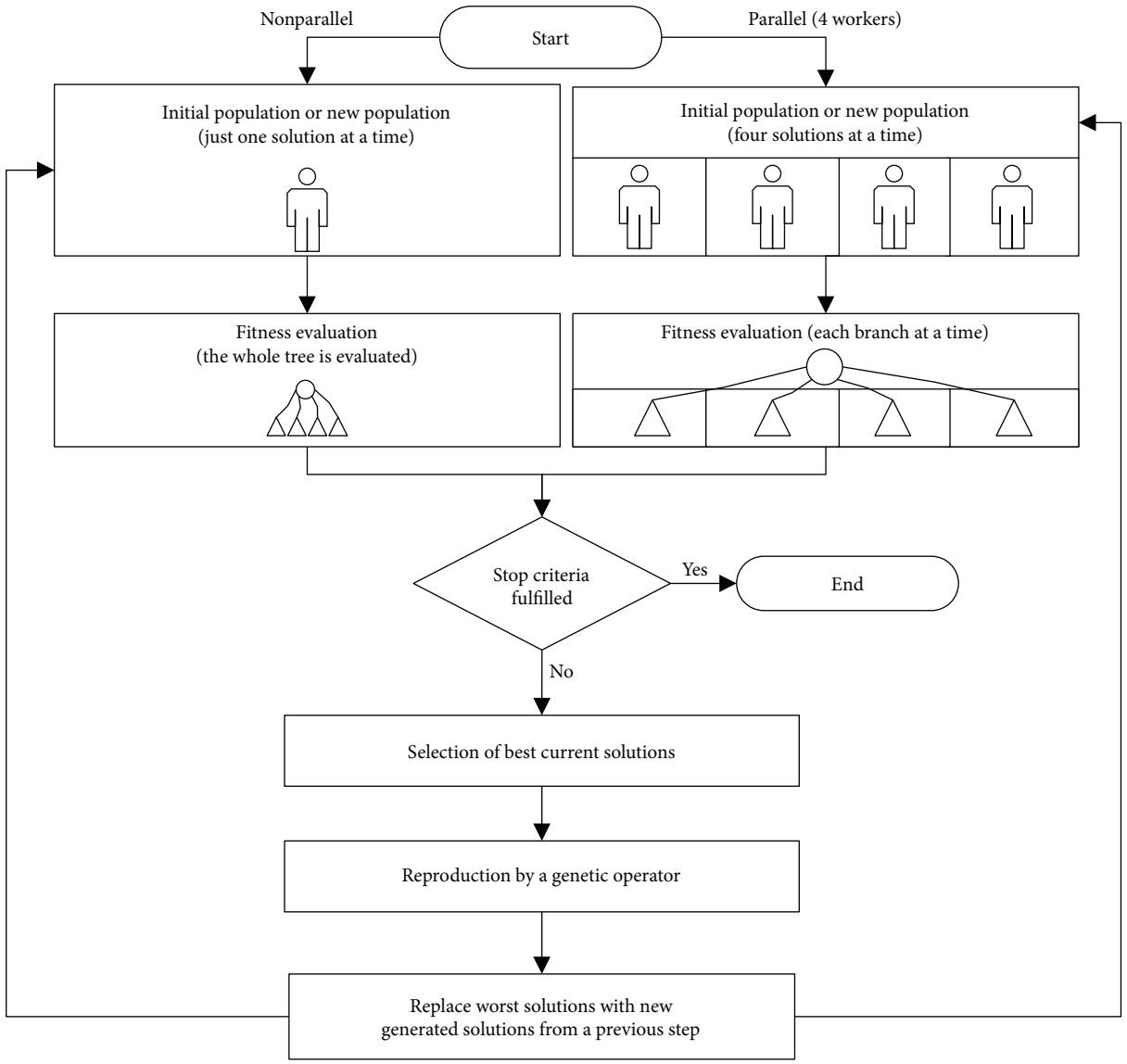


Figure 3: Parallelization options for M3GP: an example with four independent threads (or workers). After the start, a nonparallel branch and a parallel one are shown. The latter highlights possible ways to parallelize the M3GP method, that is, (i) to evolve each solution independently, (ii) to compute each branch of the fitness tree separately, or (iii) to combine the two previous approaches.

Table 3: Parameters of M3GP used in all the parallel variants.

Parameter	Value
Runs	30
Population size	10,50,100, and 150 individuals
Generations	1
Initialization	12-Depth full initialization with 1, 5, 10, 20, and 30 dimensions
Operator probabilities	Crossover $p_c = 0.5$, mutation $p_\mu = 0.5$
Function set	$(+, -, \times, \div$ protected as in [34])
Terminal set	Ephemeral random constants [0,1]
Bloat control	17-Depth limit
Selection	Lexicographic tournament of size 5
Elitism	Keep best individual

to create the new population. As anticipated before, this is probably one of the most intuitive ways to apply parallelism to M3GP. Accordingly, the overall computational time in the worst-case scenario, that is, when all individuals require the same time to be evolved, is $(P/t)C$, where P is the population size (i.e., the total number of individuals), t is the number of threads, and $C = \max\{c_i\}$, with $1 \leq i \leq P$, is the maximum time required to evaluate an individual and c_i is the computational time necessary to evolve the i^{th} individual. Moreover, we also assume that $t \leq P$ because having more threads than the number of individuals in the population will not bring any additional benefit to the parallelization since the excess threads will be unused.

Although this approach is very efficient from a theoretical point of view, some considerations must be made. In particular, one drawback is the memory consumption required to

store all the individuals processed by the different threads. The number of nodes in the trees, which will impact the amount of memory required to store each tree, can lead to high memory consumption and a possible bottleneck. Unfortunately, in GP it is common for individual models to evolve larger than they need to be, which is referred to as the bloat phenomenon.

On the other hand, the second approach we took for parallelization is the fitness computation, which is split, in the *Fitness Parallel* implementation, among the different threads. In details, the evaluation takes as an input a tree composed of d subtrees, with d the number of dimensions of the transformed feature space of each individual in M3GP. Moreover, let F be the maximum cost necessary to evaluate one of the d subtrees connected to the root, that is, $F = \max\{f_i\}$, with $1 \le i \le d$, where f_i is the cost required to compute the fitness in the ith subtree.

The overall computational time to evaluate a single individual, which is the one required by the *Non-Parallel* approach, is $dF + d$, since in the worst case each of the d subtrees requires F time to be computed, plus an additional d time to combine these results. In any case, each subtree can be computed independently so in the Fitness Parallel implementation the computation is split into t threads. In this case, the cost is $(d/t)F + (d/t)\log t$, assuming that $t \le d$. In fact, by having more threads than dimensions of the search space will only waste the excessive computational power.

In more detail, assuming the worst-case scenario in which the computation of all the subtrees requires F time, we can split the d computations on t threads, leading to a $(d/t)F$ time. Further, we can also take advantage of the parallelism to recombine these results so that this process is executed in a binary-tree structure. In this way, the d results are recursively recombined in pairs, which are distributed in t threads, so that the overall time for recombining the results is $(d/t)\log t$. Finally, as is easy to observe, when $t = 1$, we obtain the same complexity of the *Non-Parallel* approach, that is, $dF + d$.

5.1. Parallel Experiments. M3GP was implemented using the GPLAB toolbox in Matlab [], which is freely available at http://gplab.sourceforge.net. Each parallel M3GP variant was implemented using the parallel computing tools in Matlab. The experiments were carried out on a workstation with an 8-core CPU and 16 GB of RAM.

In our experiments, the goal was to evaluate the relative improvement of the parallel approach in conditions that required a high computational cost. After some preliminary experimental runs, the configuration reported in Table 3 was used to stress the search process and illustrate the benefits of the different parallel approaches. In particular, it is noted that the highest initial tree depth was set to 12 levels, making a total of 2,048 nodes for each dimension in an evolved transformation. This makes the required computational cost of evaluating each individual feature significant, and evaluating each individual is more costly when increasing the total number of dimensions. Moreover, we compared the total computational time using different population sizes and a different number of total initial dimensions in the population. For the sake of comparison, M3GP without any parallelization (referred to as *Non-Parallel*) was run with the same settings as a baseline.

A parallel environment for the experiments was set up to check for time improvements in just one generation. Letting the algorithm run for a complete search will just increase the computational cost linearly since each generation is independent of the previous one, in particular when the current solution is already close to the maximum allowed depth. For a fair comparison, the same random seeds were used for all parallel M3GP variants so that the random tree generation process did not influence the results.

5.2. Comparison of the Different Parallel Approaches. The results are summarized in the plots of Figure 4 where for each run, the time required to evaluate a population for the sequential M3GP and all the three parallel versions (i.e., *Pop Parallel*, *Fitness Parallel*, and *Full Parallel*) is shown. The x-axis in each plot corresponds to the number of dimensions of each individual, while each plot corresponds to a different population size (i.e., 10, 50, 100, and 150).

Running M3GP with just one dimension (one branch below the root node) does not provide any substantial benefit for the parallelization. In this case, with different population sizes, the fastest evaluation was the *Non-Parallel* M3GP. The reason is that in the *Fitness Parallel* version the majority of the time was spent sending data to different workers and then merging the computed results.

A tendency that emerged from the results we obtained and that is easy to observe in all the plots is that increments in the population size do not substantially affect the trend of the total computational time. In fact, it is almost linear in all cases considered, although the (absolute) time required to process bigger populations is greater than that required for processing smaller populations.

On the other hand, by increasing the number of dimensions, the overall workload entailed in evaluating each solution increases. The parallel implementations start to show significant improvements when five dimensions are used, and they continuously improve when more dimensions are added.

This is particularly true of the *Fitness Parallel* implementation, which tends to require less time than the other variants do. In all analyzed cases, only the *Fitness Parallel* variant requires less processing time than the sequential *Non-Parallel* M3GP does. It is evident that when increasing the population size and the number of dimensions, *Full Parallel* and *Pop Parallel* start to perform worse with each increment.

5.3. M3GP Speed-Up. To assess the speed-up of the different M3GP systems we proposed, the implementations were tested with up to 8 threads, and the results are shown in Figures 5 and 6. The values reported are calculated as the ratio between the run time of the *Non-Parallel* version over the parallel ones. Then, by varying the number of threads, we can measure the gain in terms of speed.

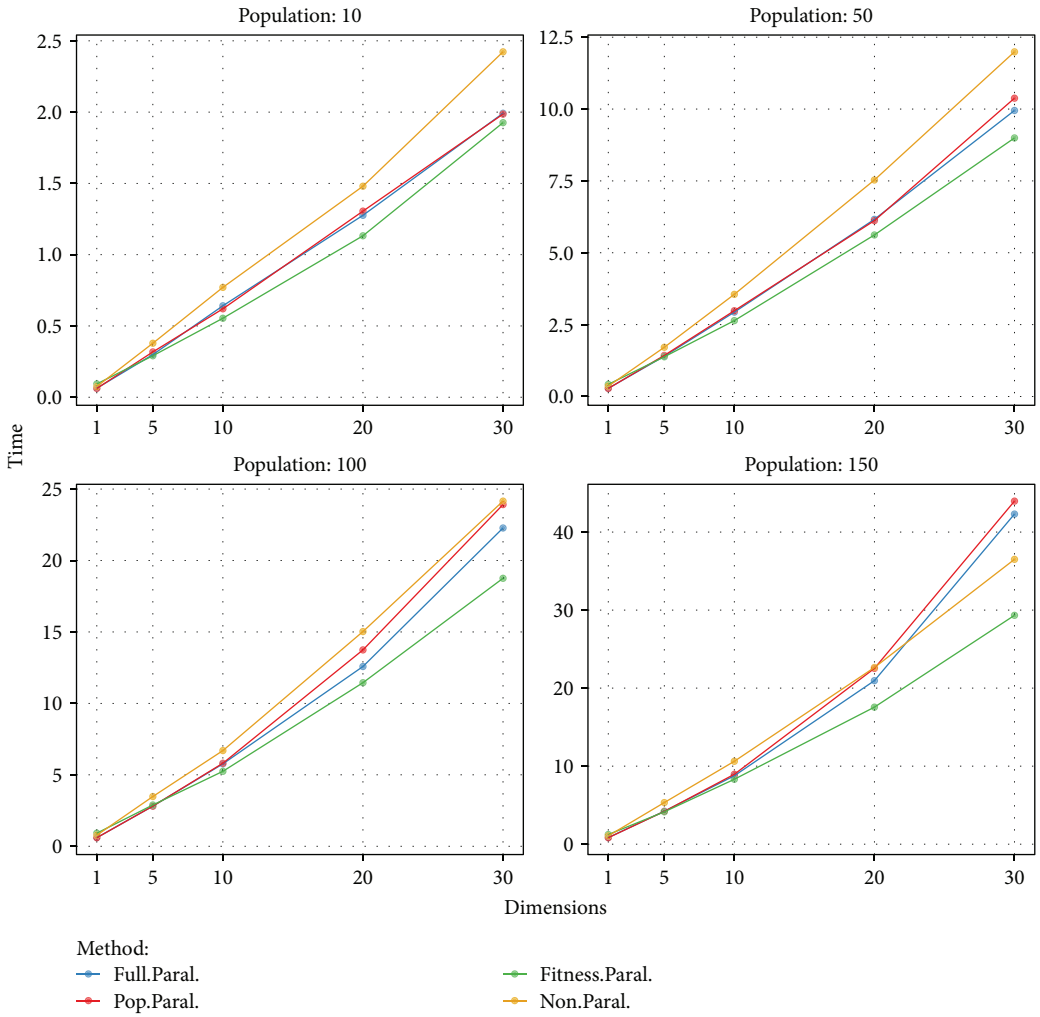


Figure 4: Parallel runs with M3GP, time (in minutes) versus dimensions.

In the plots in Figure 5, we reported the speed-up of the three parallel implementations of M3GP for individuals with 10 dimensions, tested over different values of the population size: 10, 50, 100, and 150. There is a performance gain when using 2 or more threads. In fact, when adopting only one thread, we have values lower than 1 because the parallel runs required additional operations, with respect to the *Non-Parallel* one, leading to this loss in performance. The *Full Parallel* and *Pop Parallel* perform best in this setup, with *Parallel Fitness* always performing slightly worse, except for the simplest case with the smallest population. However, it is still necessary to evaluate how this performance generalizes when evolving larger transformations with more dimensions.

Therefore, we performed similar tests by fixing the population size to 100 and varying the number of dimensions in the individual transformations. In Figure 6, we show the plots for 1, 5, 10, 20, and 30 dimensions. Not surprisingly, in the first case with only a single dimension, the *Parallel Fitness* implementation does not take advantage of the increasing number of threads since its parallelization is based on the number of dimensions. In the other cases, we can observe that by increasing the number of dimensions, this implementation improves its speed-up, achieving the best performance when the number of dimensions reaches 30.

Finally, although Matlab allows a very fast and useful way to realize and test the M3GP method (also providing an easy way to implement a parallel algorithm), we think that a more efficient implementation could lead to a significant performance improvement.

6. Conclusions

In this work, we applied a novel GP-based classification method called M3GP to the problem of integrating the results of different miRNA-target prediction tools, namely, miRanda, TargetScan, and RNAhybrid, and classifying them. Moreover, we derived a parallel implementation of this algorithm to reduce the computational cost of the search when evolving large multidimensional transformations.

The results we obtained with the M3GP method are promising and competitive when compared with the ones achieved by the other classifiers we tested (Euclidean distance, Mahalanobis distance, naive Bayes, SVM, KNN, and

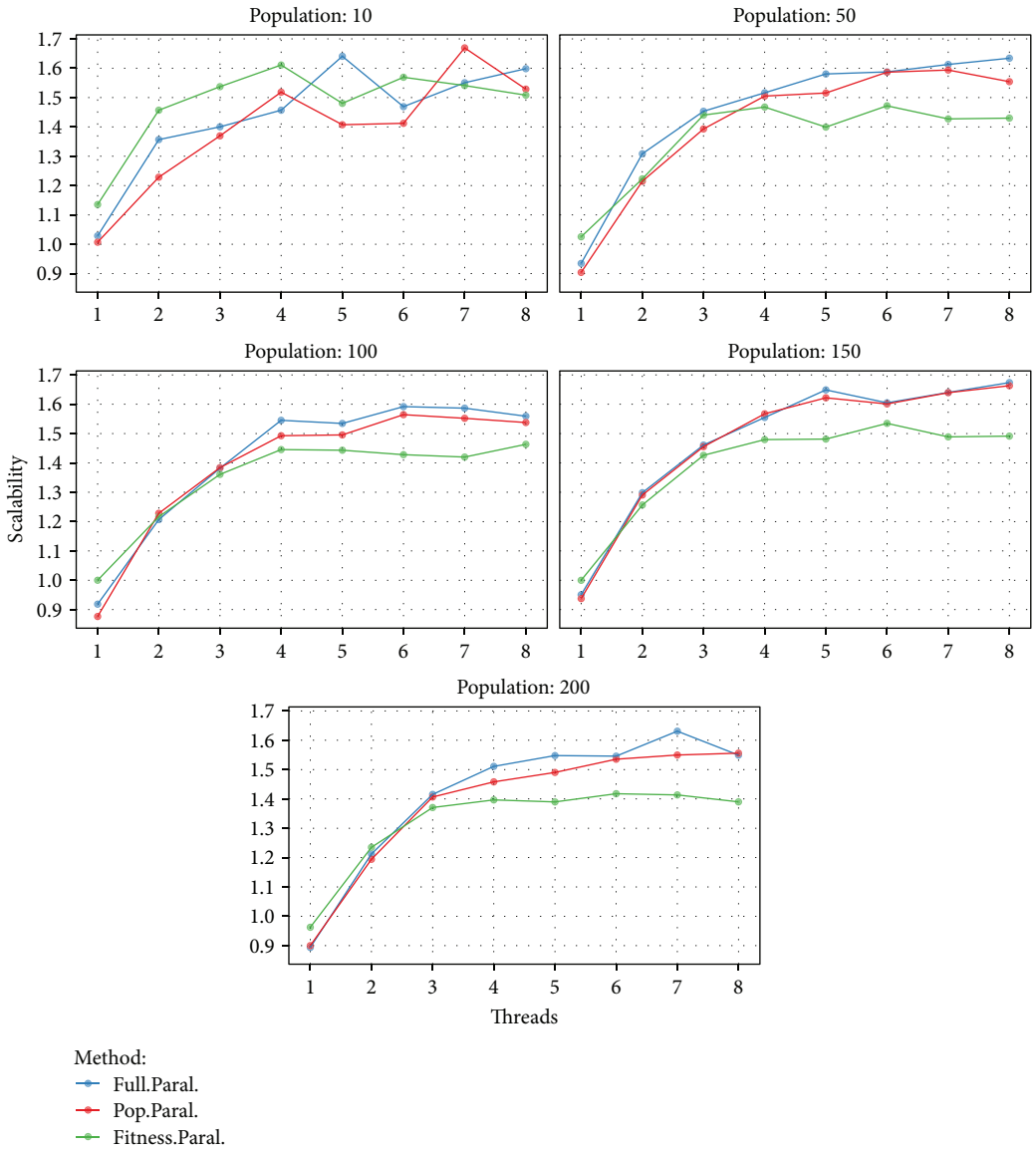


Figure 5: Speed-up plots for solutions having 10 dimensions, with population sizes of 10, 50, 100, and 150 individuals, and using from 1 to 8 threads.

Treebagger). The main advantage in contrast with other machine learning techniques is that the performance of M3GP is more robust than that of the other methods. Specifically, in this domain M3GP shows almost no overfitting or lack of generalization relative to the training performance and is robust to the training set used since its performance variance is almost null.

This makes the M3GP method a good candidate for solving the problem at hand. We also showed that M3GP is easily parallelizable. In fact, like most evolutionary algorithms, M3GP can be parallelized during fitness evaluation, or by assigning each solution to a different thread, or by combining these two approaches. However, it was shown that the best strategy is to parallelize the evaluation of every single individual by sending each tree branch (feature dimension) of the root node to a different thread. This can be done by virtue of the way in which M3GP individuals construct their output response. This approach showed substantial improvements when compared to the sequential approach and to other parallelization options.

One of the possible future research directions will focus on exploiting the parallelization aspects by adopting a more efficient implementation of the M3GP method and also by testing its execution in a distributed environment. Moreover, to further improve the robustness of the results and the strength of the approach, we plan to integrate the results of other miRNA-target prediction tools (in addition to the ones we considered here) but also to employ other datasets, such as [38]. As a final remark, we would like to point out that for future study, one can employ additional target sequences, such as the 5-UTR of the genes and the coding part of the transcripts.

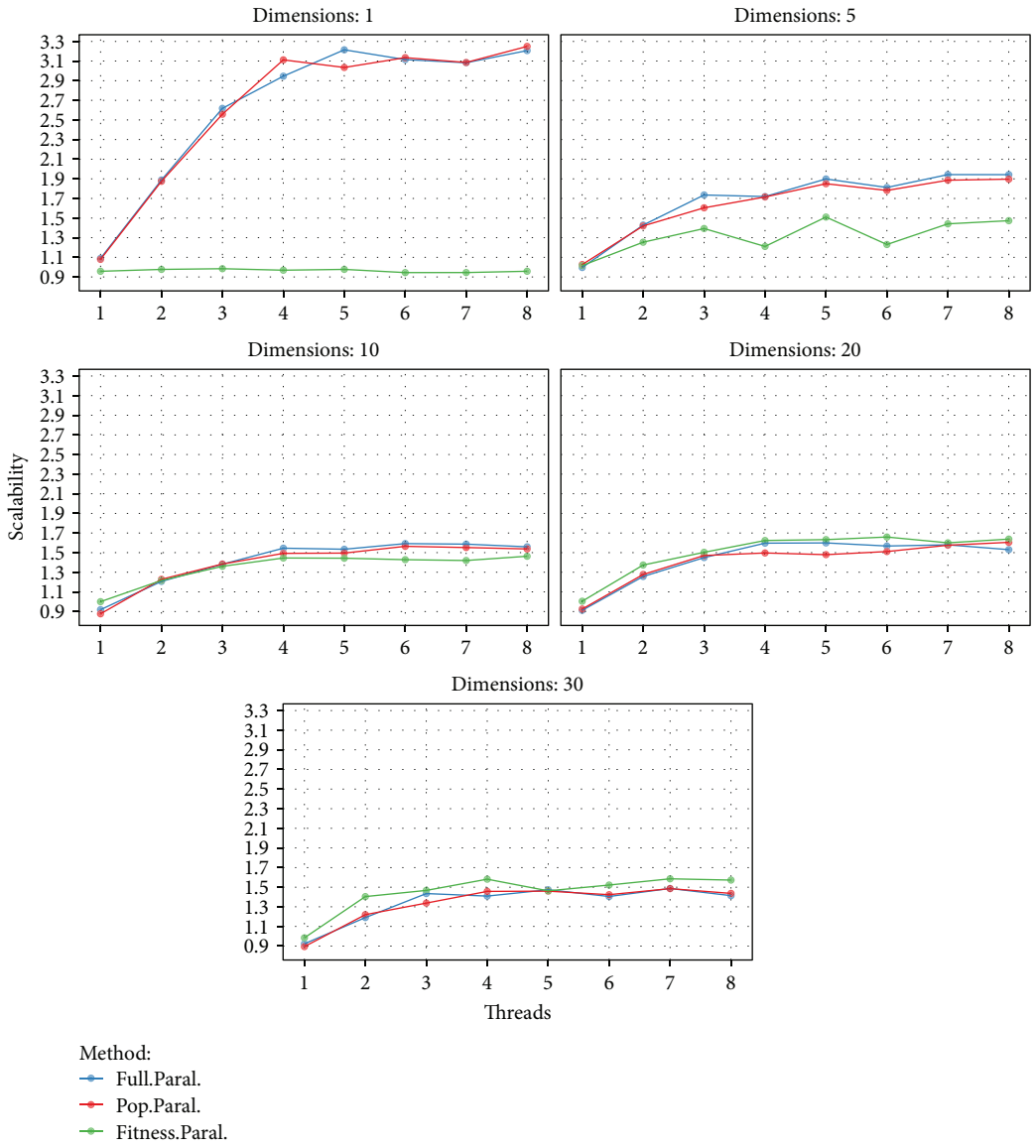


Figure 6: Speed-up plots for a population size of 100, with individuals of 1, 5, 10, 20, and 30 dimensions and using from 1 to 8 threads.

Data Availability

As described in the article, we used data available at the following website: http://www.targetscan.org/.

Conflicts of Interest

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

Acknowledgments

This research was partially funded by Consejo Nacional de Ciencia y Tecnología (Mexico) Fronteras de la Ciencia 2015-2 Project no. FC-2015-2:944, and the third author was supported by Consejo Nacional de Ciencia y Tecnología graduate scholarship 401223.

References

[1] S. Manvati, K. C. Mangalhara, J. Khan et al., "Deciphering the role of microRNA – a step by step guide," *Gene Expression Patterns*, vol. 25-26, pp. 59–65, 2017.

[2] Y. Peng and C. M. Croce, "The role of microRNAs in human cancer," *Signal Transduction and Targeted Therapy*, vol. 1, no. 1, article 15004, 2016.

[3] L. He, X. He, L. P. Lim et al., "A microRNA component of the p53 tumour suppressor network," *Nature*, vol. 447, no. 7148, pp. 1130–1134, 2007.

[4] M. Leclercq, A. B. Diallo, and M. Blanchette, "Prediction of human miRNA target genes using computationally reconstructed ancestral mammalian sequences," *Nucleic Acids Research*, vol. 45, no. 2, pp. 556–566, 2017.

[5] C. P. C. Gomes, J. H. Cho, L. Hood, O. L. Franco, R. W. Pereira, and K. Wang, "A review of computational tools in microRNA discovery," *Frontiers in Genetics*, vol. 4, p. 81, 2013.

[6] X. Fan and L. Kurgan, "Comprehensive overview and assessment of computational prediction of microRNA targets in animals," *Briefings in Bioinformatics*, vol. 16, no. 5, pp. 780–794, 2015.

[7] P. K. Srivastava, T. Moturu, P. Pandey, I. T. Baldwin, and S. P. Pandey, "A comparison of performance of plant miRNA target prediction tools and the characterization of features for genome-wide target prediction," *BMC Genomics*, vol. 15, no. 1, p. 348, 2014.

[8] M. M. Akhtar, L. Micolucci, M. S. Islam, F. Olivieri, and A. D. Procopio, "Bioinformatic tools for microRNA dissection," *Nucleic Acids Research*, vol. 44, no. 1, pp. 24–44, 2016.

[9] M. Kertesz, N. Iovino, U. Unnerstall, U. Gaul, and E. Segal, "The role of site accessibility in microRNA target recognition," *Nature Genetics*, vol. 39, no. 10, pp. 1278–1284, 2007.

[10] T.-P. Lu, C.-Y. Lee, M.-H. Tsai et al., "miRSystem: an integrated system for characterizing enriched functions and pathways of microRNA targets," *PLoS One*, vol. 7, no. 8, article e42390, 2012.

[11] C. E. Vejnar and E. M. Zdobnov, "MiRmap: comprehensive prediction of microRNA target repression strength," *Nucleic Acids Research*, vol. 40, no. 22, pp. 11673–11683, 2012.

[12] M. D. Paraskevopoulou, G. Georgakilas, N. Kostoulas et al., "DIANA-microT web server v5. 0: service integration into miRNA functional analysis workflows," *Nucleic Acids Research*, vol. 41, no. W1, pp. W169–W173, 2013.

[13] C. Coronnello and P. V. Benos, "ComiR: combinatorial microRNA target prediction tool," *Nucleic Acids Research*, vol. 41, no. W1, pp. W159–W164, 2013.

[14] H. Dweep et al.M. Alvarez and M. Nourbakhsh, "miRWalk database for miRNA–target interaction," in *RNA Mapping. Methods in Molecular Biology (Methods and Protocols), vol 1182*Humana Press, New York, NY, USA.

[15] A. Krek, D. Grün, M. N. Poy et al., "Combinatorial microRNA target predictions," *Nature Genetics*, vol. 37, no. 5, pp. 495–500, 2005.

[16] A. J. Enright, B. John, U. Gaul, T. Tuschl, C. Sander, and D. S. Marks, "MicroRNA targets in *drosophila*," *Genome Biology*, vol. 5, no. 1, article R1, 2003.

[17] B. John, A. J. Enright, A. Aravin, T. Tuschl, C. Sander, and D. S. Marks, "Human microrna targets," *PLoS Biology*, vol. 2, no. 11, article e363, 2004.

[18] B. P. Lewis, I.-h. Shih, M. W. Jones-Rhoades, D. P. Bartel, and C. B. Burge, "Prediction of mammalian microRNA targets," *Cell*, vol. 115, no. 7, pp. 787–798, 2003.

[19] R. C. Friedman, K. K.-H. Farh, C. B. Burge, and D. P. Bartel, "Most mammalian mRNAs are conserved targets of microRNAs," *Genome Research*, vol. 19, no. 1, pp. 92–105, 2009.

[20] M. Rehmsmeier, P. Steffen, M. Hochsmann, and R. Giegerich, "Fast and effective prediction of microRNA/target duplexes," *RNA*, vol. 10, no. 10, pp. 1507–1517, 2004.

[21] S. Nam, B. Kim, S. Shin, and S. Lee, "miRGator: an integrated system for functional annotation of microRNAs," *Nucleic Acids Research*, vol. 36, Supplement 1, pp. D159–D164, 2008.

[22] E. R. Gamazon, H.-K. Im, S. Duan et al., "Exprtarget: an integrative approach to predicting human microrna targets," *PLoS One*, vol. 5, no. 10, article e13534, 2010.

[23] R. Edgar, M. Domrachev, and A. E. Lash, "Gene expression omnibus: Ncbi gene expression and hybridization array data repository," *Nucleic Acids Research*, vol. 30, no. 1, pp. 207–210, 2002.

[24] K. D. Kaya, G. Karakülah, C. M. Yakıcıer, A. C. Acar, and Ö. Konu, "mESadb: microRNA expression and sequence analysis database," *Nucleic Acids Research*, vol. 39, Supplement 1, pp. D170–D180, 2011.

[25] D. Bielewicz, J. Dolata, A. Zielezinski et al., "mirEX: a platform for comparative exploration of plant pri-miRNA expression data," *Nucleic Acids Research*, vol. 40, no. D1, pp. D191–D197, 2012.

[26] G. Sales, A. Coppe, A. Bisognin, M. Biasiolo, S. Bortoluzzi, and C. Romualdi, "MAGIA, a web-based tool for miRNA and genes integrated analysis," *Nucleic Acids Research*, vol. 38, Supplement 2, pp. W352–W359, 2010.

[27] D. Corrada, F. Viti, I. Merelli, C. Battaglia, and L. Milanesi, "myMIR: a genome-wide microRNA targets identification and annotation tool," *Briefings in Bioinformatics*, vol. 12, no. 6, pp. 588–600, 2011.

[28] S. Beretta, M. Castelli, Y. Marítnez et al., "A machine learning approach for the integration of mirna-target predictions," in *2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP)*, pp. 528–534, Heraklion, Greece, Febuary 2016.

[29] S. Bandyopadhyay and R. Mitra, "TargetMiner: microRNA target prediction with systematic identification of tissue-specific negative examples," *Bioinformatics*, vol. 25, no. 20, pp. 2625–2631, 2009.

[30] S.-D. Hsu, Y.-T. Tseng, S. Shrestha et al., "miRTarBase update 2014: an information resource for experimentally validated miRNA-target interactions," *Nucleic Acids Research*, vol. 42, pp. D78–D85, 2014.

[31] V. Ingalalli, S. Silva, M. Castelli et al.K. Krawiec, M. I. Heywood, M. Castelli et al., "A multi-dimensional genetic programming approach for multi-class classification problems," in *17th European Conference on Genetic Programming, volume 8599 of LNCS*, M. Nicolau, Ed., pp. 48–60, Springer, Granada, Spain, 2014.

[32] L. Munoz, S. Silva, L. Trujillo et al.M. I. Heywood, J. McDermott, M. Castelli et al., "M3GP: multiclass classification with GP," in *18th European Conference on Genetic Programming, volume 9025 of LNCS*, pp. 78–91, Springer, Copenhagen, 2015.

[33] M. Sipper, F. W, K. Ahuja, and J. H. Moore, "Investigating the parameter space of evolutionary algorithms," *BioData Mining*, vol. 11, no. 1, p. 2, 2018.

[34] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT press, 1992.

[35] E. Alba and J. M. Troya, "A survey of parallel distributed genetic algorithms," *Complexity*, vol. 4, no. 4, pp. 31–52, 1999.

[36] P. Angeline and K. Kinnear, *Massively Parallel Genetic Programming*, MIT Press, 1996.

[37] M. García-Valdez, L. Trujillo, J.-J. Merelo, F. F. de Vega, and G. Olague, "The EvoSpace model for pool-based evolutionary algorithms," *Journal of Grid Computing*, vol. 13, no. 3, pp. 329–349, 2015.

[38] A. Ruepp, A. Kowarsch, D. Schmidl et al., "PhenomiR: a knowledgebase for microRNA expression in diseases and biological processes," *Genome Biology*, vol. 11, no. 1, article R6, 2010.

Hindawi
Complexity
Volume 2018, Article ID 3078374, 11 pages
https://doi.org/10.1155/2018/3078374

Research Article

Feature Representation Using Deep Autoencoder for Lung Nodule Image Classification

Keming Mao,[1] **Renjie Tang**,[2] **Xinqi Wang**,[1] **Weiyi Zhang**,[1] **and Haoxiang Wu**[1]

[1]*College of Software, Northeastern University, Shenyang, Liaoning Province 110004, China*
[2]*China Mobile Group Zhejiang Co., Ltd., Hanzhou, Zhejiang Province 310016, China*

Correspondence should be addressed to Keming Mao; maokm@mail.neu.edu.cn

Received 16 January 2018; Accepted 27 March 2018; Published 7 May 2018

Academic Editor: Dimitrios Vlachakis

This paper focuses on the problem of lung nodule image classification, which plays a key role in lung cancer early diagnosis. In this work, we propose a novel model for lung nodule image feature representation that incorporates both local and global characters. First, lung nodule images are divided into local patches with Superpixel. Then these patches are transformed into fixed-length local feature vectors using unsupervised deep autoencoder (DAE). The visual vocabulary is constructed based on the local features and bag of visual words (BOVW) is used to describe the global feature representation of lung nodule image. Finally, softmax algorithm is employed for lung nodule type classification, which can assemble the whole training process as an end-to-end mode. Comprehensive evaluations are conducted on the widely used public available ELCAP lung image database. Experimental results with regard to different parameter setting, data augmentation, model sparsity, classifier algorithms, and model ensemble validate the effectiveness of our proposed approach.

1. Introduction

Lung cancer is one of the most deadly diseases around the world, with about 20% among all cancers in 2016. The 5-year cure rate is only 18.2% in spite of great progress in recent diagnosis and treatment. It is noted that if the patient can be accurately diagnosed in the early stage and suitable treatment can be implemented, there will be a greater chance for their survival [1]. Therefore, it is of great significance to do research about early diagnosis of lung cancer. Computed Tomography (CT) is currently the most popular method among lung cancer screening technologies [2]. CT can generate high resolution data, which enable small/low-contrast lung nodules effectively detected compared with conventional radiography methods. According to the report of National Lung Screening, low-dose CT scan reduces lung cancer mortality by a rate of 20% [3]. Due to the fact that traditional lung cancer diagnosis only relies on professional experts, two main drawbacks will be caused: (1) subjectivity, different doctors have different diagnostic results for the same CT scan image; (2) huge workload, reading CT images consumes much time and effort. This makes the efficiency inevitably weakened. With the development of computer vision technology, some benefits are brought for medical image process and analysis. Its efficiency and stability provide auxiliary help for doctors with automatically or semiautomatically pattern.

During the last two decades, a number of researchers have been devoted to the development of medical image process and analysis with computer vision and machine learning technologies especially for lung disease diagnosis [4]. Among these studies, lung nodule image classification has attracted much attentions for it is a key step for lung cancer analysis. The lung nodule is characterized by its appearance and relation between surrounding regions. Usually, the lung nodule can be classified into 4 types [5], as shown in Figure 1. To be specific, Figures 1(a)–1(d) demonstrate nodule types W, V, J, and P, respectively, where

W is well-circumscribed nodule located centrally in the lung without any connection to other structures;

V is vascularized nodule that is also central in the lung but closely attached to the neighbouring vessels;

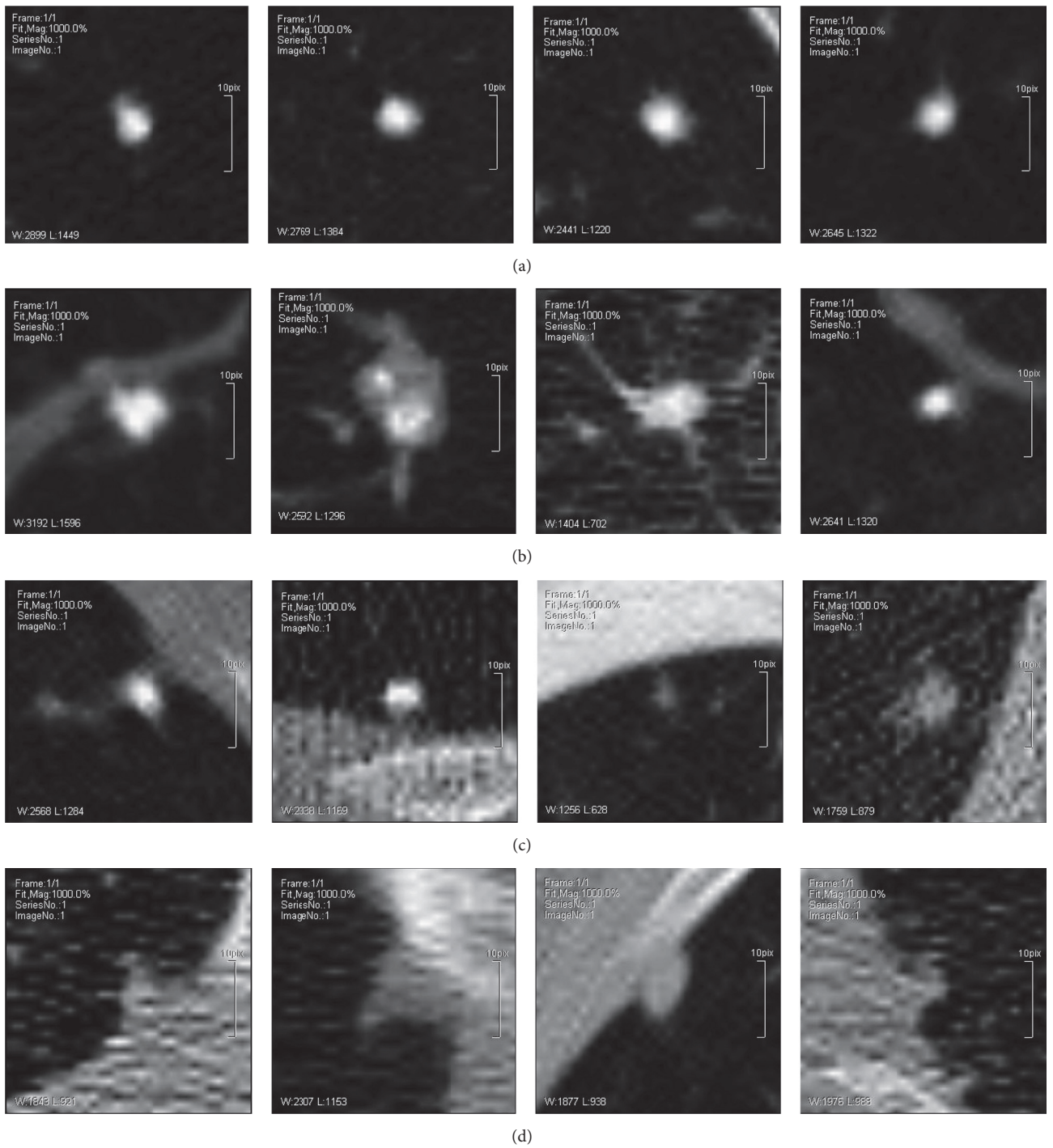


Figure 1: Demonstration of four types lung nodule image samples (cropped from images in [6]).

J is juxtapleural nodule that has a large portion connected to the pleura;

P is pleural-tail nodule that is near the pleural surface connected by a thin tail.

Lung nodule CT image classification includes two main steps. First, feature extraction and representation use segmentation, filter, and statistical method to describe feature of lung nodule based on shape and texture. Second, classifier design constructs classifier based on supervised or unsupervised machine learning method. However, these methods belong to the fields of traditional image processing and machine learning, which can only characterize the abstraction of lung nodule image in a shallow layer and make the research at low level. As a result, the complex structure of lung nodule makes the classification still a challenging problem. This paper proposes a novel model for lung nodule feature representation and classification. The model considers both

local feature and global feature. Lung nodule CT images are first divided into local patches with Superpixel, and each patch is associated with a relatively intact tissue. Then local feature is extracted from each patch with deep autoencoder. Visual vocabulary is constructed with local features. Global representation is constructed by bag of visual word (BOVW) model and classifier is trained using softmax algorithm. The main contributions of our work are as follows: (i) a novel feature representation model for lung nodule image classification is proposed. Local and global features are constructed by unsupervised deep autoencoder and BOVW model, and (ii) comprehensive evaluations are conducted, and performance analyses are reported from multiple aspects.

The structure of this paper is organized as follows. Related works are introduced in Section 2. Section 3 gives the framework. Local feature representation and global feature representation are given in Sections 4 and 5. Section 6 presents the classifier model. Experimental evaluations are shown in Section 7. Section 8 concludes this paper.

2. Related Works

Many studies have reported the classification of lung nodule in CT image. Some representative works are introduced in this section. Many researches designed feature based on texture, shape, and intensity of lung nodule image. A feature extraction method based on morphological and shape of lung nodule was designed in [7]. A subclass local constraint based method is proposed in [8]. Spectral clustering and approximate affine matrix were used to construct data subclass and each subclass was used as reference dictionary. The testing image was represented by sparse dictionary. Finally, two metrics based on approximation and distribution degree were merged. In [9], spectrum was sampled around center of lung nodule and feature was constructed by FFT. All features were used to construct the dictionary, and then BOVW mode was used to represent the feature of lung nodule. The Haralick texture feature based on spatial direction distribution was proposed in [10], and SVM was used as classifier finally. Ridge direction information was adopted in [11]. Local random comparison method was used to construct the feature vector, and then random forest was used as classifier. Reference [12] first labeled nodule as solid, part-solid, and nonsolid. Then shape based feature was extracted and kNN was used train the classifier. Reference [13] adopted smoothness and irregularity of lung nodule as feature representation. Texture, shape, statistics, and intensity were extracted as feature representation and ANN was used as classifier in [14]. An eigenvalue of Hessian matrix based feature extraction method is adopted in [15], and AdaBoost was used as classifier. Reference [16] used rotation-invariant second-order Markov-Gibbs random field to model the intensity distribution of lung nodule, and Gibbs energy was used to describe the feature vector. Finally, Bayes classifier was constructed. LDA and 3D lattice were used to construct the mapping between lung nodule image and feature representation in [17]. Reference [18] used topology histogram to represent feature vector of lung nodule, and discriminant and K-means were used as classifier. These methods represent the lung nodule image feature in relatively low level, and they lack sophisticated extraction. On the other hand, these methods need heavy participation of professional expert and they have less generality.

Some well-engineered feature extraction and representation methods widely used in computer vision domain were adopted in lung nodule image classification. Reference [22] proposed a method based on texture and context of lung nodule. Lung images are divided into nodule level and context level; then SIFT, LBP, and HOG features were extracted. Reference [19, 23] divided lung nodule as foreground and background with graph model and conditional random field. Then SIFT was used to extract feature and SVM was used as classifier. In [24], SIFT feature was first extracted. Then PCA and LDA were used for dimension reduction. Finally, complex Gabor response was used for representation. In [25], a supervised method was used for initial classification with 128-length SIFT descriptor and weighted Clique was constructed using 4-length probability vector against the 4 nodule types. The overlap that lung nodule belongs to different types was used for optimizing the final classification result. These methods adopt general designed features. They obtain higher performance compared with traditional low-level features, while such methods are considered as mid-level abstraction of lung nodule and with less flexibility.

Several methods were concerned with other aspects. An ensemble based method was applied in [26] for lung nodule classification. Lung nodule image patch was used as input, and six large scale artificial neural networks were trained for classification. Data imbalance problem was discussed in [27]. It used downsampling and SMOTE algorithms to train lung nodule classifier.

Due to its breakthrough in the field of image processing and speech recognition, deep learning has become one of the most hottest topics in machine learning research and application [20, 28–30]. High-level abstraction of image object can be described using deep learning model. Meanwhile, feature extraction and representation are more efficient and effective. In [28], curvature, hu-moment, morphology, and shape features were used to detect nodule candidate region. Then convolutional neural network (CNN) was used to extract feature for candidate region and multiple classifiers were merged for final result. Some changes were made in [29, 30]. OverFeat was used for CNN parameter initialization. In [20], a deep feature extraction with one hidden layer autoencoder was adopted, and a binary decision tree was used as classifier for lung cancer detection. This paper proposes a lung nodule image classification method combining both local and global feature representation. Our proposed work is close but has essential difference from the work of [20]. Method in [20] just applied one hidden layer autoencoder to lung nodule image. Our proposed method uses Superpixel to generate intact patches and deep autoencoder to extract local feature. Moreover, method BOVW is incorporated for lung nodule global feature representation and method in [20] has no consideration.

3. Framework

The procedure of proposed lung nodule classification method is shown in Figure 2. It contains training and testing stages.

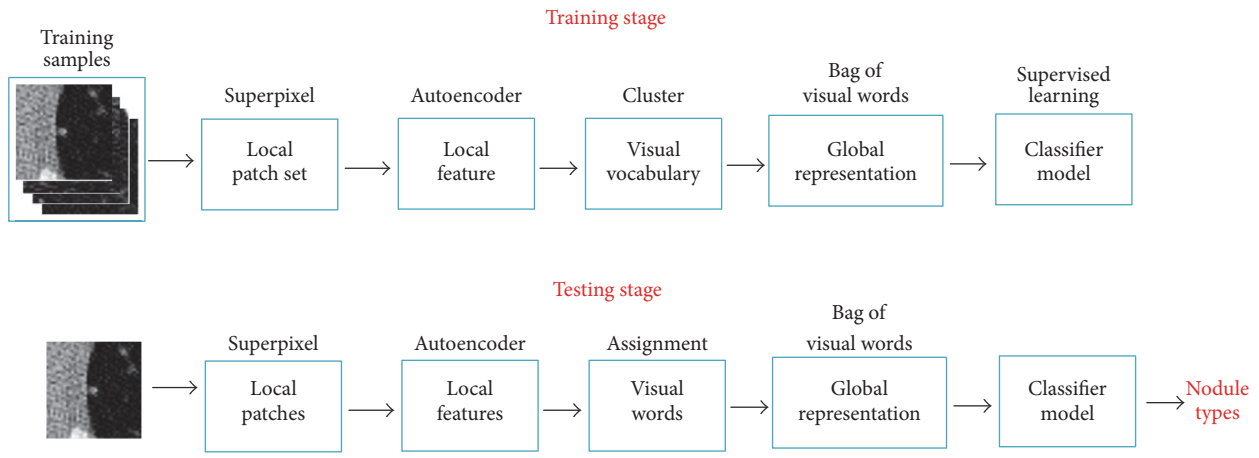


Figure 2: Framework of the proposed method.

In training stage, lung nodule image samples are used as input and the output is a trained classifier model. Collected training image samples are first divided into local patches with Superpixel. Local patches are assigned with no class label and constitute local patch set. With the local patch set, local features are extracted by unsupervised learning model, deep autoencoder. Next, visual vocabulary is constructed based on clustering all local feature vectors. A lung nodule image can therefore be described by a global feature representation with bag-of-visual-words model. Finally, classifier is trained by supervised learning with nodule type labels. In testing stage, the input is a lung nodule image with unknown type, and the output is its predicted type label. Similar to training stage, a test image is divided into multiple patches. Each local patch is transformed into local feature and assigned with a visual word. Finally, global feature representation of test image is used for classification by the trained model. Details of the proposed method will be introduced in the following sections.

4. Local Feature Representation

Local feature representation is proposed in this section. The process consists of two steps: (1) local patch generation and (2) local feature extraction and representation.

4.1. Local Patch Generation. Decomposing a lung image into small patches is useful and practical and for important tissues can be picked up and unrelated ones can be get rid of. As shown in (1), a lung nodule image x can be composed of a group of image patches p_i, where n denotes the number of local patches:

$$x = \{p_1, p_2, \ldots, p_n\}. \quad (1)$$

The location and scale of local patches are determined through generation [22, 24]. Useless part will be contained for large size patch, while small part may not cover enough intact tissue. Superpixel is a popular method that can partition the image into small similar regions with better representativeness and integrity [31]. So it is adopted in this work.

Figure 3 illustrates the process of the proposed local patch generation method. For a lung nodule image (Figure 3(a)), it is first segmented into local patches using Superpixel and a Superpixel map is obtained (Figure 3(b)). Local patches essentially indicate the uniform regions. Figure 3(c) is an individual patch sample. However, the region that Figure 3(c) gives is an irregular shape, and it is inconvenient for local feature extraction and representation. So we expand local patch with its minimum enclosing rectangle, as shown in Figure 3(d). Finally, a lung nodule image is decomposed into a set of local patches, as shown in Figure 3(e). Besides, there are some additional criterions to determine whether an image patch is qualified for local feature extraction:

(i) Let p_i be a local patch; it is removed when the area of p_i is larger than $A_{\max}$ or smaller than $A_{\min}$.

(ii) Let p_i and p_j be two local patches; if the ratio between their intersection and their union is larger than O_t, then the smaller one is removed.

$A_{\max}$, $A_{\min}$, and O_t are predefined thresholds.

4.2. Local Feature Extraction and Representation. With the rapid development of unsupervised learning in recent years, using unlabeled data to extract feature with autoencoder has become an appropriate way. Autoencoder model is essentially a multilayered neural networks. Its original version is a forward network with one hidden layer. Let x_i be the input data, a_i^j be the activation of unit i in layer j, and w_i be the matrix of weights controlling function mapping from layer i to layer $i+1$. If layer i has s_i units and layer $i+1$ has s_{i+1} units, then w_i will be a matrix with size of $s_i * s_{i+1}$. The activation can be formulated as (2), where a_1^2 is the 1st unit in the 2nd layer and x_0–x_3 are 4 input features:

$$a_1^2 = g\left(w_{10}^1 x_0 + w_{11}^1 x_1 + w_{12}^1 x_2 + w_{13}^1 x_3\right). \quad (2)$$

The main difference between ordinary forward neural network and autoencoder is that an autoencoder's output is

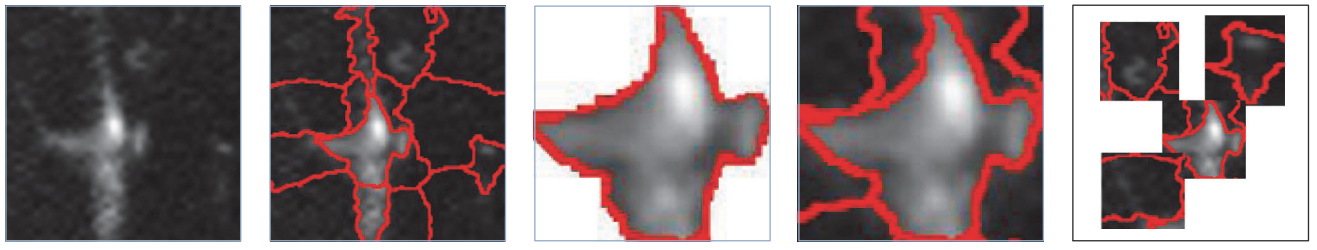


Figure 3: The process of local patch generation by Superpixel.

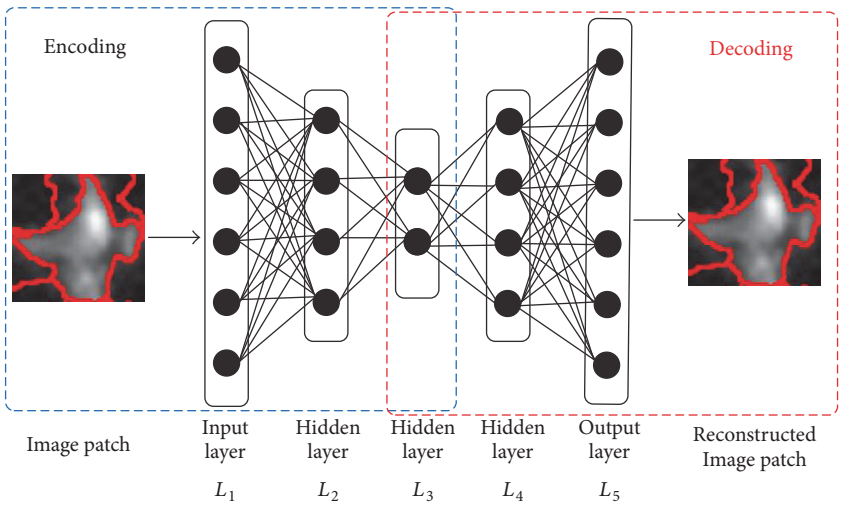


Figure 4: The process of stacked deep autoencoder model.

always the same as or similar to its input. The basic formula can be expressed as follows:

$$a = h(x) = f\left(W^E x + b\right),$$
$$x' = h'(x) = g\left(W^D a + b'\right) = g\left(W^D h(x) + b'\right). \quad (3)$$

An autoencoder can be seen as a combination of encoder and decoder. The encoder includes an input layer and a hidden layer, which converts an input image x into feature vector a. The decoder includes a hidden layer and an output layer that transform feature a to output feature x'. W^E and W^D are weight matrices of encoder and decoder, respectively. Functions $f(\cdot)$ and $g(\cdot)$ can be either sigmoid or tanh activation functions, which is used to activate the unit in each layer. When x' approximates x, it is considered that the input feature can be reconstructed from an abstract and compressed output feature vector a. The cost function can be generally defined as follows:

$$J(W,b) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\left\|x_i' - x_i\right\|^2 + \lambda\sum_{l=1}^{N_l-1}\sum_{i=1}^{M_i}\sum_{j=1}^{M_{i+1}}\left(W_{ij}^l\right)^2. \quad (4)$$

A deep autoencoder can be constructed by stacking more hidden layers. As shown in Figure 4, there are 5 layers in the model (including 3 hidden layers). L_1 to L_3 are encoding layers, and L_3 to L_5 are decoding layers. L_i is used as the input of the layer L_{i+1}, and the weights can be gained based on (3). There are 2 stacked autoencoders. The activation of 1st hidden layer is the input of the 2nd stacked autoencoder. The network can be trained in a fine-tuning stage by minimizing the equation (4). W_1 and W_4 are trained through the encoding and decoding weights of the 1st stacked autoencoder, and W_2 and W_3 are trained through the encoding and decoding weights of the 2nd stacked autoencoder. Finally, the whole network can be constructed layer by layer in a stacked way. Moreover, Figure 4 just shows an example of symmetric encoding and decoding structures, and other variational structures can also be adopted.

Therefore, each local patch of a lung nodule image p_i can be represented by a fixed-length feature vector pf_i with deep autoencoder model. Then (1) is transformed as follows:

$$x = \{p_1, p_2, \ldots, p_n\} = \{pf_1, pf_2, \ldots, pf_n\}. \quad (5)$$

5. Global Feature Representation

For BOVW model, visual vocabulary is first constructed based on clustering all local patch descriptors (local feature representation) generated by a set of training images. Then each lung nodule image can be represented globally by a histogram of visual words. Distance between histograms of

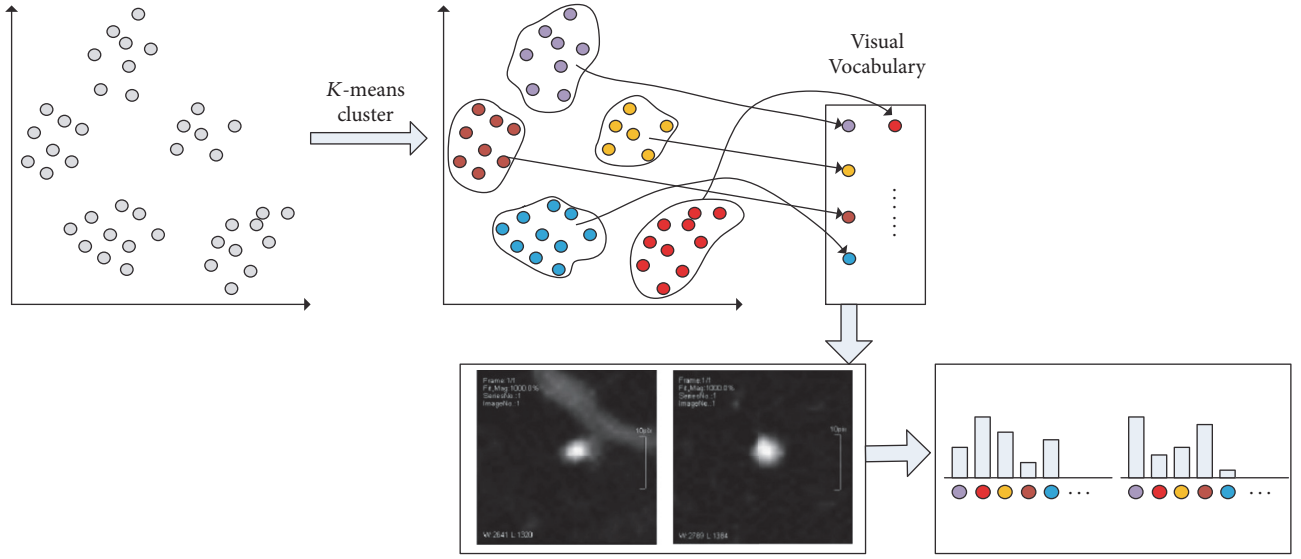


Figure 5: Procedure of BOVW representation of lung nodule image.

visual words can be treated as similarity between lung nodule image samples.

Recall that a lung nodule image is decomposed into a group of local patches and each patch is represented with a feature vector based on deep autoencoder. Assume there are D local patches generated from all lung nodule training images and each local patch is represented with d-dimensional feature vector; then all local feature vectors can be assembled into a feature space with size of $d * D$. Clustering is performed with $d * D$ features, and k-means clustering method is adopted since it has relatively low time and storage complexity, irrelevant to data process ordering. Each cluster center c_i represents a visual word i, and k cluster centers constitute the visual vocabulary. A lung nodule image sample x can be represented by the encoded local patches as a bag, which is the occurrence frequency of visual word in vocabulary. To get the histogram representation $h(x)$ of an image x, all local patch feature vectors of x are mapped onto the cluster center of the visual vocabulary, and each local feature is assigned with the label of its closest cluster center using Euclidean distance in feature space. Then a k-bins histogram $h(x)$ is obtained by counting all the label of local patches generated by image x, as shown in (6). Figure 5 exhibits the procedure of global representation of lung nodule image.

$$h(x) = [h(x)_1, h(x)_2, \ldots, h(x)_k]. \tag{6}$$

6. Classifier Model

With global representation of lung nodule image, softmax algorithm is used to train nodule type classifier. Let $\{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$ denote training data set. x_i denotes the lung nodule image sample and $y_i \in \{0, 1, 2, 3\}$ denotes nodule type label.

For an input image sample x_i, we want to compute $p(y = j \mid x_i)$ $(j \in \{0, 1, 2, 3\})$. The output, a 4-dimensional vector, is estimated to represent the probability of each type that x_i belongs to. The hypothesis function can be expressed as follows:

$$h_\theta(x_i) = \begin{bmatrix} p(y_i = 0 \mid x_i; \theta) \\ p(y_i = 1 \mid x_i; \theta) \\ p(y_i = 2 \mid x_i; \theta) \\ p(y_i = 3 \mid x_i; \theta) \end{bmatrix} = \frac{1}{\sum_{j=0}^{3} e^{\theta_j^T x_i}} \begin{bmatrix} e^{\theta_0^T x_i} \\ e^{\theta_1^T x_i} \\ e^{\theta_2^T x_i} \\ e^{\theta_3^T x_i} \end{bmatrix}, \tag{7}$$

where $\theta = \{\theta_0, \theta_1, \theta_2, \theta_3\}$ is model parameter set. This equation normalizes the result and makes the sum to 1. For training procedure, the loss function is given as follows:

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=0}^{3} 1\{y_i = j\} \log \frac{e^{\theta_j^T x_i}}{\sum_{l=0}^{3} e^{\theta_l^T x_i}}\right] + \frac{\lambda}{2}\sum_{i=1}^{k}\sum_{j=0}^{n}\theta_{ij}^2, \tag{8}$$

where $1\{\cdot\}$ is an indicative function, and stochastic gradient descent (SGD) is used for function optimization and the corresponding derivative functions are given as follows:

$$\nabla_{\theta_j} J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[x_i\left(1\{y_i = j\} - p(y_i = j \mid x_i; \theta)\right)\right] + \lambda\theta_j, \tag{9}$$

$$p(y_i = j \mid x_i; \theta) = \frac{e^{\theta_j^T x_i}}{\sum_{l=0}^{3} e^{\theta_l^T x_i}}.$$

Figure 6: Demonstration of lung CT images (downloaded from [6]).

7. Experimental Evaluations

7.1. Dataset and Program Implementation. In order to evaluate the performance of the proposed lung nodule image representation and classification method, a widely used public available lung nodule image dataset, ELCAP, is used for testing [6]. The dataset contains 379 lung CT images, which are collected from 50 distinct low-dose CT lung scans. The center position of lung nodule is marked in an extra $*$.csv file.

Figure 6 demonstrates the lung nodule CT scan images, which are sampled from different slices. Table 1 shows the format of a $*$.cvs file. Each row denotes a lung nodule. The 4th column indicates the slice number where the lung nodule exists. The 2nd and 3rd columns give the positions that the lung nodule is located in. In this section, lung nodule images are cropped from the raw CT images based on the x- and y-coordinates of nodule center given in Table 1. The raw lung CT scan image is fixed with $512 * 512$ pixels, and the cropped nodule images are too small to implement the algorithm. Therefore we further resize the cropped lung nodule image into $180 * 180$ pixels with bicubic method. The lung nodule images are labeled with one of four types according to the guidance by an expert. Programs are implemented with Matlab 2016a programming language and tested on a Pentium i7 CPU, 8 G RAM, NVIDIA GTX 960 GPU, Windows OS PC.

The experiments include the following aspects: (1) parameter setting; (2) classification rate with different parameters; (3) classification rate with data augmentation; (4) classification rate with model sparsity; (5) classification rate with different classifier algorithms; (6) comparing with other methods; (7) classification rate with model ensemble. The performance of lung nodule image classification is computed with overall classification rate, as shown in the following:

$$\text{Classification rate} = \frac{N_{\text{correct}}}{N_{\text{all}}}, \tag{10}$$

where N_{correct} is the number of correctly labeled images and N_{all} is the number of all testing images. Cross validation mode is adopted. The dataset is divided into 8 groups: 7 randomly chosen groups are used for training and the left group is used for testing. This process is repeated 7 times and the result is computed by averaging 7 independent tests.

Table 1: Format of lung nodule position.

Type	x	y	Slice
Nodule	98	218	54
Nodule	355	153	84
Nodule	139	366	130
Nodule	436	213	169
Nodule	372	163	239
Nodule	328	175	229
Nodule	54	224	169

7.2. Parameter Setting. The parameters are needed to be set in local patch generation, local feature representation, and global feature representation. For local patch generation, we need to set the number of superpixels that each lung nodule image generates. For local feature representation, the number of hidden layers and nodes that each layer contains should be set. For global feature representation, the size of visual vocabulary should be set.

As shown in Table 2, the number of patches that each lung nodule image generates is set with 15, 20, 25, and 30. The number of hidden layers in deep autoencoder is set with 1, 2, and 3. The number of nodes in deep autoencoder is set with 50, 75, 100, 125, and 150. The size of visual vocabulary is set with 200, 300, 400, and 500. The classification rate is evaluated on the combination of these parameters. For convenience, parameters are expressed with p_1, p_2, p_3, and p_4, respectively.

7.3. Classification Rate with Different Parameters. The size of local patch is set with $30 * 30$ pixels in our experiment. Table 3 gives the average performance of lung nodule image classification based on combination of parameters p_1, p_2, p_3, and p_4. It can be seen that classification model with $p_1 = 25$, $p_2 = 2$, $p_3 = (100, 50)$, and $p_4 = 400$ gets the optimal result,

Table 2: Parameter setting.

Parameter	Parameter explanation	Setting
p_1	Number of Superpixel generation	15, 20, 25, 30
p_2	Hidden layers of deep autoencoder	1, 2, 3
p_3	Nodes of deep autoencoder	50, 75, 100, 125, 150
p_4	Size of visual vocabulary	200, 300, 400, 500

Table 3: Performance with different parameters.

Parameters setting			Performance
p_1	p_2 (p_3)	p_4	
15	1 (50)	200	0.8199
15	1 (50)	400	0.832
15	1 (50)	600	0.829
15	2 (100, 50)	200	0.858
15	2 (100, 50)	400	0.86
15	2 (100, 50)	600	0.845
15	3 (150, 100, 50)	200	0.836
15	3 (150, 100, 50)	400	0.845
15	3 (150, 100, 50)	600	0.85
25	1 (50)	200	0.83
25	1 (50)	400	0.842
25	1 (50)	600	0.849
25	2 (100, 50)	200	0.887
25	2 (100, 50)	400	0.895
25	2 (100, 50)	600	0.891
25	3 (150, 100, 50)	200	0.835
25	3 (150, 100, 50)	400	0.832
25	3 (150, 100, 50)	600	0.86
40	1 (50)	200	0.824
40	1 (50)	400	0.82
40	1 (50)	600	0.813
40	2 (100, 50)	200	0.824
40	2 (100, 50)	400	0.815
40	2 (100, 50)	600	0.833
40	3 (150, 100, 50)	200	0.81
40	3 (150, 100, 50)	400	0.80
40	3 (150, 100, 50)	600	0.806

reaching 89.5%. We can also see that different parameter settings have great impact on the classification results.

7.4. Classification Rate with Data Augmentation. Overfitting is common in machine learning, and it is influenced by both model complexity and the size of training data. Data augmentation scheme is usually adopted to lessen this problem [32]. In this section, data augmentation is used to enlarge the size of training data. Random rotation, random cropping, and random perturbation (brightness, saturation, hue, and contrast) are used as basic augment techniques.

For original lung nodule image, it is sampled with possibility of 0.5 for data augmentation. The new created examples are set with same labels as original. As shown in Table 4, data augmentation can increase classification rate

Table 4: Performance with data augmentation.

Data augmentation method			Average performance
Random rotation	Random cropping	Random perturbation	
No	No	No	0.895
Yes	No	No	0.899
No	Yes	No	0.887
Yes	Yes	No	0.908
No	No	Yes	0.903
Yes	No	Yes	0.911
No	Yes	Yes	0.912
Yes	Yes	Yes	0.924

with 3%. This shows that adding more augmented data for training can improve the compatibility and generalization of the classification model.

7.5. Classification Rate with Model Sparsity. In this subsection, a sparsity constraint is imposed on the hidden layer. Sparsity is a recently proposed technique to improve the generalization of the model [33]. A sparsity regularization term is added to (4), and the new objective functions are given as follows:

$$J(W,b) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}\left\|x_i' - x_i\right\|^2 + \lambda\sum_{l=1}^{N_l-1}\sum_{i=1}^{M_i}\sum_{j=1}^{M_{i+1}}\left(W_{ij}^{l}\right)^2 + \beta\sum_{i=1}^{N_l-1}\mathrm{KL}\left(\rho \parallel \rho_i\right), \tag{11}$$

$$\mathrm{KL}\left(\rho \parallel \rho_i\right) = \rho\log\frac{\rho}{\rho_i} + (1-\rho)\log\frac{(1-\rho)}{(1-\rho_i)},$$

$$\rho_i = \frac{1}{N}\sum_{i=1}^{N}h_i(x).$$

The sparsity regularization term is regulated by Kullback-Leibler divergence $\mathrm{KL}(\rho \parallel \rho_i)$. ρ_i is the average activation of ith layer of deep autoencoder and ρ is the target activation. ρ with small value can reduce the mean activation of the model. β is a trade-off parameter. Table 5 gives the result of classification performance with different ρ (values from 0.1–0.9). It can be seen that ρ set around 0.3–0.4 leads to the superior performance.

7.6. Classification Rate with Different Classifier Algorithms. In this subsection, we evaluate the performances of 4 commonly used classifier algorithms. Softmax (which is used in this paper), SVM, kNN, and decision tree are used. The same feature representation is adopted. Table 6 shows that softmax slightly outperforms SVM, kNN, and decision tree. The

Table 5: The effect of model sparsity.

Sparsity (ρ)	Performance
0.1	0.92
0.2	0.929
0.3	0.938
0.4	0.939
0.5	0.93
0.6	0.905

Table 6: The effect of different classifier algorithms.

Classifier model	Performance
Softmax	0.939
SVM	0.931
kNN	0.927
Decision tree	0.919

Table 7: Performance comparing with other methods.

Classification method	Performance
Ref. [19]	0.877
Ref. [7]	0.88
Ref. [20]	0.82
Ref. [21]	0.895
Ref. [9]	0.891
Our proposed method	0.939

Table 8: Performance of model ensemble.

Number of models	Performance
1	0.939
5	0.952
6	0.954
7	0.955

results demonstrate that, compared with classifier algorithm, the feature representation is the key problem. Meanwhile, it is easy to combine the softmax algorithm and the proposed feature representation method into an end-to-end structure, which can make model training more convenient.

7.7. Comparing with Other Methods. In order to evaluate the classification rate of different methods, 5 related algorithms are used for testing. Reference [19] studies the same problem as ours. Reference [20] adopts the primitive autoencoder method. References [7, 21] use non-deep-learning methods for classification. Reference [9] employs the BOVW model. The compared methods are reimplemented and are tested with diverse parameters. Table 7 gives the testing result. Among all testing methods, the proposed one demonstrates the best performance. Comparing with non-deep-learning method, our method can construct better feature representation, while, comparing with primitive autoencoder method, the Superpixel and DAE used in our method can catch more detailed information.

7.8. Classification Rate with Model Ensemble. Model ensemble can improve the classification performance by aggregating multiple individual classifiers [34]. We evaluate model ensemble based on Majority Rule in this subsection. In Majority Rule, the class label is assigned with the one that most classifier votes. The function to evaluate the class label e for image I is given as follows:

$$p(e) = \sum_{i=1}^{N} S\left(C_i(I) = e\right),$$
$$e = \arg\max_i p\left(e_i\right), \tag{12}$$

where I is a testing image, e is a class label, N denotes number of selected models, and C_i means ith classifier. $S(C_i(I) = e) = 1$, if C_i classifies I as e. The label with maximal value of $p(\cdot)$ is determined as the final result. If multiple labels have the same votes, the arithmetic average of class probabilities predicted by individual model is used as classification result.

With different parameters combination, models with top performances are retained for ensemble. Table 8 gives the testing result. The 1st row denotes the single model. The 2nd to 4th rows denote model ensemble with 5, 6, and 7 individual models, respectively. The result demonstrates that model ensemble can complement individual ones and the performance is improved with about 1.5%.

8. Conclusion and Future Works

In this paper, a novel feature representation method is proposed for lung nodule image classification. Superpixel is first used to divide lung nodule image into local patches. Then local feature is extracted and represented from local patches with deep autoencoder. Bag-of-visual-words model is used as global feature representation with visual vocabulary constructed by local feature representation. Finally, an end-to-end training is implemented with a softmax classifier. The proposed method is evaluated from many aspects, including parameter setting, data augmentation, model sparsity, comparison among different algorithms, and model ensemble. We draw a conclusion that the proposed method achieves superior performance. The merits of our method are the combination of local and global feature representation, and better model generalization can be gained by incorporating unsupervised deep learning model.

Our future works will focus on two aspects: (1) study new classification framework and method according to up-to-date convolutional neural network and (2) analysis of our method in large data set for making further improvement and optimization.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Liaoning Doctoral Research Foundation of China (no. 20170520238), National Natural Science Foundation of China (no. 61772125 and no. 61402097). The authors gratefully acknowledge the support of NVIDIA Corporation with the donation of GPU used for this research.

References

[1] National Cancer Institute, "SEER Stat Fact Sheets: Lung and Bronchus Cancer. NCI online 2016," http://seer.cancer.gov/statfacts/html/lungb.html.

[2] American Cancer Society, https://www.cancer.org.

[3] Y. Shieh and M. Bohnenkamp, "Low-Dose CT Scan for Lung Cancer Screening: Clinical and Coding Considerations," *CHEST*, vol. 152, no. 1, pp. 204–209, 2017.

[4] A. K. Dhara, S. Mukhopadhyay, A. Dutta, M. Garg, and N. Khandelwal, "Content-Based Image Retrieval System for Pulmonary Nodules: Assisting Radiologists in Self-Learning and Diagnosis of Lung Cancer," *Journal of Digital Imaging*, vol. 30, no. 1, pp. 63–77, 2017.

[5] S. Diciotti, G. Picozzi, M. Falchini, M. Mascalchi, N. Villari, and G. Valli, "3-D segmentation algorithm of small lung nodules in spiral CT images," *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, no. 1, pp. 7–19, 2008.

[6] ELCAP public lung image database, http://www.via.cornell.edu/databases/lungdb.html.

[7] J. N. Stember, "The Normal Mode Analysis Shape Detection Method for Automated Shape Determination of Lung Nodules," *Journal of Digital Imaging*, vol. 28, no. 2, pp. 224–230, 2015.

[8] Y. Song, W. Cai, H. Huang, Y. Zhou, Y. Wang, and D. D. Feng, "Locality-constrained subcluster representation ensemble for lung image classification," *Medical Image Analysis*, vol. 22, no. 1, pp. 102–113, 2015.

[9] F. Ciompi, C. Jacobs, E. T. Scholten et al., "Bag-of-frequencies: a descriptor of pulmonary nodules in computed tomography images," *IEEE Transactions on Medical Imaging*, vol. 34, no. 4, pp. 962–973, 2015.

[10] F. Han, G. Zhang, H. Wang et al., "A texture feature analysis for diagnosis of pulmonary nodules using LIDC-IDRI database," in *Proceedings of the 2013 IEEE International Conference on Medical Imaging Physics and Engineering, ICMIPE 2013*, pp. 14–18, IEEE, Shenyang, China, October 2013.

[11] J. Bai, X. Huang, S. Liu, Q. Song, and R. Bhagalia, "Learning orientation invariant contextual features for nodule detection in lung CT scans," in *Proceedings of the 12th IEEE International Symposium on Biomedical Imaging, ISBI 2015*, pp. 1135–1138, IEEE, New York, NY, USA, April 2015.

[12] C. Jacobs, E. M. van Rikxoort, T. Twellmann et al., "Automatic detection of subsolid pulmonary nodules in thoracic computed tomography images," *Medical Image Analysis*, vol. 18, no. 2, pp. 374–384, 2013.

[13] T. W. Way, B. Sahiner, H.-P. Chan et al., "Computer-aided diagnosis of pulmonary nodules on CT scans: improvement of classification performance with nodule surface features," *Medical Physics*, vol. 36, no. 7, pp. 3086–3098, 2009.

[14] S. Akram, M. Y. Javed, and A. Hussain, "Automated thresholding of lung CT scan for Artificial Neural Network based classification of nodules," in *Proceedings of the 14th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2015*, pp. 335–340, IEEE, Las Vegas, NV, USA, July 2015.

[15] A. Robert, G. Jonathan, A. Fereidoun et al., "Automated classification of lung bronchovascular anatomy in CT using AdaBoost," *Medical Image Analysis*, vol. 11, no. 3, pp. 315–324, 2007.

[16] A. El-Baz, G. Gimel'farb, R. Falk, and M. El-Ghar, "Appearance analysis for diagnosing malignant lung nodules," in *Proceedings of the 7th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, ISBI 2010*, pp. 193–196, IEEE, Rotterdam, Netherlands, April 2010.

[17] H. Takizawa, S. Yamamoto, and T. Shiina, "Accuracy improvement of pulmonary nodule detection based on spatial statistical analysis of thoracic CT scans," *IEICE Transaction on Information and Systems*, vol. 90, no. 8, pp. 1168–1174, 2007.

[18] K. Yoshiki, N. Noboru, O. Hironobu et al., "Hybrid Classification Approach of Malignant and Benign Pulmonary Nodules Based on Topological and Histogram Features," *Medical Image Computing & Computer Assisted Intervention*, pp. 297–306, 2000.

[19] F. Zhang, Y. Song, W. Cai et al., "A ranking-based lung nodule image classification method using unlabeled image knowledge," in *Proceedings of the IEEE 11th International Symposium on Biomedical Imaging (ISBI '14)*, pp. 1356–1359, Beijing, China, May 2014.

[20] D. Kumar, A. Wong, and D. A. Clausi, "Lung nodule classification using deep features in CT images," in *Proceedings of the 12th Conference on Computer and Robot Vision (CRV '15)*, pp. 133–138, IEEE, Halifax, Canada, June 2015.

[21] A. K. Dhara, S. Mukhopadhyay, A. Dutta, M. Garg, and N. Khandelwal, "A Combination of Shape and Texture Features for Classification of Pulmonary Nodules in Lung CT Images," *Journal of Digital Imaging*, vol. 29, no. 4, pp. 466–475, 2016.

[22] F. Zhang, Y. Song, W. Cai et al., "Lung nodule classification with multilevel patch-based context analysis," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 4, pp. 1155–1166, 2014.

[23] Y. Song, W. Cai, Y. Wang, and D. D. Feng, "Location classification of lung nodules with optimized graph construction," in *Proceedings of the 9th IEEE International Symposium on Biomedical Imaging (ISBI '12)*, pp. 1439–1442, IEEE, Barcelona, Spain, May 2012.

[24] A. Farag, S. Elhabian, J. Graham, A. Farag, and R. Falk, "Toward precise pulmonary nodule descriptors for nodule type classification," *Medical Image Computing and Computer-Assisted Intervention*, vol. 13, no. 3, pp. 626–633, 2010.

[25] F. Zhang, W. Cai, Y. Song, M.-Z. Lee, S. Shan, and D. Dagan, "Overlapping node discovery for improving classification of lung nodules," in *Proceedings of the 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC '13)*, pp. 5461–5464, Osaka, Japan, July 2013.

[26] K. Suzuki, F. Li, S. Sone, and K. Doi, "Computer-aided diagnostic scheme for distinction between benign and malignant nodules in thoracic low-dose CT by use of massive training artificial neural network," *IEEE Transactions on Medical Imaging*, vol. 24, no. 9, pp. 1138–1150, 2005.

[27] Y. Sui, Y. Wei, and D. Zhao, "Computer-aided lung nodule recognition by SVM classifier based on combination of random undersampling and SMOTE," *Computational and Mathematical Methods in Medicine*, Article ID 368674, pp. 1–13, 2015.

[28] 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 Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1160–1169, 2016.

[29] B. Van Ginneken, A. A. A. Setio, C. Jacobs, and F. Ciompi, "Off-the-shelf convolutional neural network features for pulmonary nodule detection in computed tomography scans," in *Proceedings of the 12th IEEE International Symposium on Biomedical Imaging (ISBI '15)*, pp. 286–289, Brooklyn, NY, USA, April 2015.

[30] W. Shen, M. Zhou, F. Yang, C. Yang, and J. Tian, "Multi-scale convolutional neural networks for lung nodule classification," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 9123, pp. 588–599, 2015.

[31] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2281, 2012.

[32] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," https://arxiv.org/abs/1207.0580.

[33] X. Zhang and R. Wu, "Fast depth image denoising and enhancement using a deep convolutional network," in *Proceedings of the 41st IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016*, pp. 2499–2503, Shanghai, China, March 2016.

[34] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 1–39, 2010.