Applied Soft Computing Journal 97 (2020) 106718

Contents lists available at ScienceDirect

Applied Soft Computing Journal

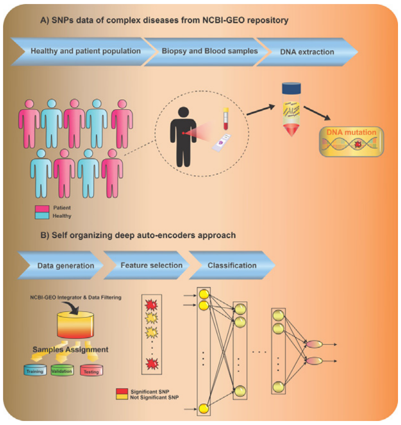
journal homepage: www.elsevier.com/locate/asoc

A Self-organizing Deep Auto-Encoder approach for Classification of Complex Diseases using SNP Genomics Data

Saeed Pirmoradi [a,1], Mohammad Teshnehlab [b,*,1], Nosratollah Zarghami [c,1], Arash Sharifi [a,1]

[a] Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
[b] Department of Control Engineering, K.N. Toosi University of Technology, Tehran, Iran
[c] Department of Biotechnology, Tabriz University of Medical Sciences, Tabriz, Iran

GRAPHICAL ABSTRACT



ARTICLE INFO

Article history:
Received 29 April 2020
Received in revised form 5 August 2020
Accepted 3 September 2020
Available online 12 September 2020

Keywords:
Single Nucleotide Polymorphism (SNP)
Deep learning
Self-organizing auto-encoder
Feature selection
Complex diseases

ABSTRACT

Recently, many Machine Learning algorithms have been utilized to identify significant Single Nucleotide Polymorphisms (SNPs) in various human diseases. However, some principal obstacles are challenging in the field of SNP detection and healthy-patient classification. The curse of dimensionality is the main challenge. On the other hand, the number of samples is decidedly smaller than the number of SNPs. In addition, the number of healthy and patient samples can be unequal. These challenges make the feature selection and classification very difficult. The main goal of the current study is the combination of the various algorithms to find out the most effective way of SNP data analysis. Therefore, an efficient method is proposed to identify significant SNPs and classify healthy and patient samples. In this regard, firstly, the Mean Encoding, as an intelligent method, is utilized to convert the nominal SNP data to numeric. Then a two-step filter method is used for feature selection, which removes the irrelevant and redundant features. Finally, the proposed deep auto-encoder is employed to classify so that it can construct its structure based on input data, automatically. To evaluate, we apply the proposed approach to five different SNP datasets, including thyroid cancer, mental retardation, breast cancer, colorectal cancer, and autism, which obtained from the Gene Expression Omnibus (GEO) dataset. The proposed method has succeeded in feature selection and classification so that it can classify healthy and patient samples based on selected features in thyroid cancer, mental retardation, breast cancer, colorectal cancer, and autism with 100%, 94.4%, 100%, 96%, and 99.1% accuracy, respectively. The results indicate that it has succeeded with high efficiency, compared with other published works.

[*] Correspondence to: K.N. Toosi University of Technology, Shariaty St., Tehran, Iran.

E-mail addresses: said.pirmoradi@gmail.com (S. Pirmoradi), teshnehlab@eetd.kntu.ac.ir (M. Teshnehlab), zarghami@tbzmed.ac.ir (N. Zarghami), arash.sharifi@gmail.com (A. Sharifi).

[1] All authors each made a significant contribution to the research.

https://doi.org/10.1016/j.asoc.2020.106718
1568-4946/© 2020 Elsevier B.V. All rights reserved.

1. Introduction

Deoxyribonucleic acid (DNA) is the building block of the human genome. There are approximately three billion base pairs in the double helix of DNA. More than 99% of them are the same among all populations, and less than 1% differ among individuals. The majority of DNA changes happen as Single Nucleotide Polymorphisms (SNPs). Studies have shown that SNPs can be very significant in association with complex diseases.

Many Machine Learning algorithms are employed to identify significant SNPs, and many models are developed to classify the healthy and patient samples based on SNP data. D. T. Evans proposed two methods to select significant SNPs, which is containing Chi-squared Sort and Difference Sort [1]. Also, he applied the Support Vector Machine (SVM) to classify healthy and patient samples. In another study, N. Batnyam et al. utilized popular feature selection algorithms to choose the significant SNPs, which involving Relief-F, Feature Selection Based on Distance Discriminant (FSDD), Feature Selection Based on R-value (RFS), and Algorithm Based on Feature Clearness (CBFS) [2]. Then the authors employed conventional classifiers like K-Nearest Neighbor (KNN), Artificial Gene Making (AGM), and SVM to classify SNP data. In addition, Feature Fusion Method (FFM) is utilized to generate new features by combining features to improve classification accuracy. A. Boutorh et al. proposed a novel method based on hybrid Association Rule Mining (ARM) and Artificial Neural Network (ANN) [3]. The authors applied ARM to select informative features, and Grammatical Evolution (GE) is used to optimize ARM. SNP data is classified by ANN, in which the Genetic Algorithm is utilized for setting ANN parameters. In recent years, R. Alzubi et al. recommended hybrid feature selection, involving Conditional Mutual Information Maximization (CMIM) as a filter method and Support Vector Machine-Recursive Feature Elimination (SVM-REF) as a wrapper method [4]. The authors employed four classifiers, including Naive Bayes, Linear Discriminant analysis, KNN, and SVM, to distinguish between healthy and patient groups. In recent studies, Uppu et al. applied a deep feedforward neural network to classify healthy and patient samples based on SNPs, which existing in simulated datasets. Also, the authors did not utilize any feature selection algorithm [5,6]. However, some principal obstacles are challenging in the field of SNPs detection. The curse of dimensionality is the main challenge because the dimension of SNP data is very high (up to one million). In high dimensional data, typically, many features are irrelevant or redundant; these properties decrease the performance of the classifier and increase the computational cost. Additionally, the number of samples (healthy or patient) are decidedly smaller than the number of SNPs that means the SNP data are sparse. Also, the amount of healthy and patient samples can be unequal, which means the SNP data are unbalanced. Data with sparse and unbalance properties are other difficulties in most studies. Besides, we need to convert nominal data to numeric data in the SNP dataset, and which encoding method for this purpose must be used. Considering all these factors, the improvement of an efficient algorithm, involving feature selection and classification, is hard and complicated.

In this context, Feature Selection (FS) algorithms play an essential role because these algorithms can identify irrelevant and redundant features so that they can reduce dimensionality by removing these features. There are many FS algorithms that each algorithm can be suitable for each particular data. Therefore, we discover which of them can be the best for each specific SNP data and will be able to select significant SNPs that cause to separate the healthy and patient samples with high accuracy.

Additionally, we need a powerful model to classify SNP data into healthy and patient groups. The classification task is carried out based on the significant SNPs, which are selected by the FS algorithm. The performance of classification illustrates that the selected SNPs have a meaningful impact on complex diseases or not. In this study, we focus on deep learning methods like deep auto-encoders. A deep learning algorithm is a specific subfield of the representation learning procedure, which detects multiple levels of representation. High-level representation (or features) illustrate more aspects of the data. The deep learning study was begun by Geoff Hinton's group in 2006 [7]. These methods are constructed by the combination of multiple nonlinear mappings (or transformations) to obtain the data representation with more abstract. A deep model is made by using stacking unsupervised representation learning models to make a deep representation. This area has been growing rapidly so that many deep learning methods have been widely discussed and reviewed in recent years. All of them have succeeded in many applications, especially in computer vision tasks, therefore it can be useful in health informatics applications. Thus, Researchers mainly have focused on important applications of deep learning in the fields of translational bioinformatics, medical imaging, pervasive sensing, medical informatics, and public health [8]. Although deep learning methods have succeeded in many applications, however, the underlying theory is not well understood. Also, there is no clear realization of which architecture accomplishes better than the others. It is challenging to characterize which structure, how many layers, and how many nodes in each layer are appropriate for a specific task [9]. It also requires specialized knowledge to determine sensible values such as the learning rate and the coefficient of the regularization. Bengio offered practical recommendations for the gradient-based training of deep architectures to determine parameters such as the learning rate, momentum, regularization coefficient, number of training iteration, and other parameters [10]. He believed that using the same size of nodes for all layers works generally better than or the same as using a decreasing or increasing size [11]. In another study, the authors declared that over-fitting is a serious problem in deep neural networks with a large number of parameters and also is slow to use. They proposed the dropout technique to address this problem [12]. In [13], the authors recommended a framework to select the hyper-parameters, involving the number of layers and nodes. It contains two basic approaches, which are manual and automatical approach. In the manual method, an expert determines hyper-parameters, which needs to understand what the hyper-parameters operate at the model. Automatical selection utilizes grid search and random search, which eliminates the necessity of experts in the model design process, but they increase time spent and computational cost. In another study, the authors applied the new penalty term to the loss function, which forces the parameters of some neurons to be zero [14]. However, the main question remains: how many layers (auto-encoders) and how many nodes (neurons) in each layer are suitable for a specific task?

Here, we discuss some of the essential questions that have been driving research in the field of Artificial Intelligence (AI) application in medical science; specifically, which encoding method is suitable for SNP data that can improve feature selection and classification performance? Which FS algorithm can be the best in each specific SNP data and can select significant SNPs? What makes one representation better than the other in deep auto-encoders? How many layers or how many nodes in each layer are appropriate? We try to answer these questions and propose a new approach to identify the significant SNPs and a new classifier to classify the (case and control) samples in complex diseases. Eventually, the proposed method is applied to five different SNP datasets, and the results display that our approach has succeeded in this area with high accuracy.

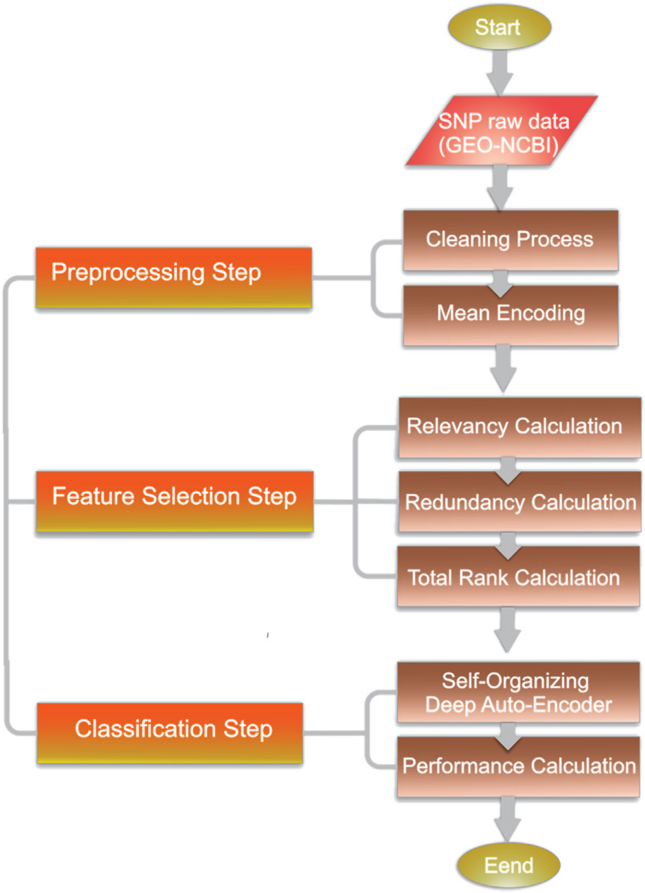


Fig. 1. The whole process flowchart.

This paper is structured as follows; Section 2 provides a framework of the whole study. The details of SNP datasets are described in Section 2.1. In Sections 2.2, 2.3, and 2.4, systems and background theories are discussed, involving the pre-processing method, the proposed feature selection algorithm, and the proposed classifier. In Section 3, we display our experimental results, where we apply the proposed approach to several SNP data and compare it with the previous works.

2. Methodology

The proposed process involves three stages: (A) A pre-processing stage, which consists of encoding nominal SNP data, and removing or replacing missing values in SNP data. (B) A feature selection stage; in this stage, significant SNPs are selected by a suitable FS algorithm. (C) A classification stage, the self-organizing auto-encoders are utilized to classify SNP data in this stage. Also, selected SNPs are evaluated according to some classification metrics such as accuracy and F1-score. The whole process is shown in Fig. 1.

2.1. SNP datasets

All of the SNP data studied in this experiment are taken from the National Center for Biotechnology Information (NCBI) Gene Expression Omnibus (GEO) website. The GEO is a public repository that archives and freely distributes Next Generation Sequencing (NGS) and other genomic data [15]. The proposed method is carried out on five complex diseases: GSE67047 [16], GSE13117 [17], GSE16619 [18], GSE34678 [19] and GSE9222 [20] that Information of these datasets are shown in Table 1.

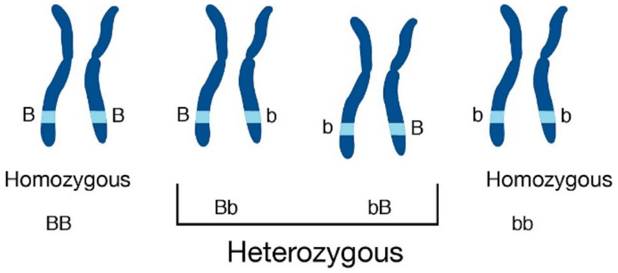


Fig. 2. Homozygous and heterozygous genotypes [4].

The SNP datasets contain five different diseases with a different number of samples (case and control), a different number of features (SNPs), and various genetic patterns. The five SNP datasets are known and frequently used in current SNP research base on Artificial Intelligence (AI). In addition, these datasets are available in Gene Expression Omnibus of the National Center for Biotechnology Information, which is one of the most reliable biology data centers in the world.

Each sample can take four possible values for a given SNP marker (or feature), that these values involve BB, Bb, bb and No Call. The BB and bb values illustrate the two homozygous genotypes, the Bb display the heterozygous genotype (as shown in Fig. 2) and when the genotype array or the calling algorithm cannot be able to determine the allele of the SNP, the No Call (NC) value is registered.

Samples are shown according to Table 2, in which all data can be obtained for a single SNP and a separate sample by taking a single row and a single column from Table 2 respectively. In addition, samples are labeled as a case or control.

2.2. Pre-processing

The pre-processing stage involves two steps. In the first step, redundant SNPs (or features) are detected and removed then in the second step, No Call or missing values are replaced by suitable amounts. In addition, the information on these steps is shown in Table 3.

2.2.1. Redundant features

The SNPs that have the same values for all case and control samples are considered as redundant features since these features or SNPs cannot separate the two groups (case and control). For example, if BB value is registered for a given SNP in all case and control, will not be helpful for feature selection and classification.

2.2.2. Missing values

Each SNP (or each row from Table 2) that involves more than 10% of No Call values are discarded; otherwise the No Call value is replaced by estimated value, which in this study is the mode of feature (most common value for a given feature in all samples) in some SNP datasets or is considered as a new feature type in some other SNP datasets.

2.2.3. Encoding method

There are different types of data in data analysis. Generally, data can be assumed as numerical data and nominal data. Most Machine Learning algorithms utilize the mathematical calculations for feature selection and classification, so nominal data should be converted to a numerical amount that can be useful. In this study, all of the SNP data are nominal data (like BB,

Table 1
Information of SNP datasets.

Dataset	No. of SNP	No. of samples	Case	Control	Information	Year	Ref.
GSE67047	1,000,000	225	96	129	Thyroid Cancer	2016	[16]
GSE13117	250,000	360	120	240	Mental Retardation	2009	[17]
GSE16619	500,000	111	69	42	Breast Cancer	2009	[18]
GSE34678	250,000	124	62	62	Colorectal Cancer	2012	[19]
GSE9222	250,000	567	335	232	Autism (ASD)	2008	[20]

Table 2
An example of the dataset with n SNPs and N samples.

Samples	Sample 1	Sample 2	Sample 3	Sample 4	...	Sample N
SNP 1	BB	Bb	BB	bb	...	Bb
SNP 2	Bb	bb	Bb	BB	...	No Call
SNP 3	No Call	BB	bb	Bb	...	bb
SNP 4	Bb	Bb	No Call	BB	...	BB
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
SNP n	BB	bb	Bb	No Call	...	Bb
Labels	Case	Control	Case	Control	...	Case

Table 3
Information on the pre-processing stage.

Dataset	No. of SNP	No. of removed SNP	No. of modified SNP
GSE67047	909,622	102,600	574,874
GSE13117	262,264	67,899	182,865
GSE16619	503,590	35702	289,949
GSE34678	299,140	33,172	199,337
GSE9222	262,338	34,493	192,096

Table 4
Example of the mean encoding method.

Sample	*i*th SNP (feature)	Feature label	Feature-mean encoding	Target label
1	BB	1	3(Case)/5(Instances) = 0.6	Case
2	BB	1	3/5 = 0.6	Control
3	BB	1	3/5 = 0.6	Case
4	BB	1	3/5 = 0.6	Case
5	BB	1	3/5 = 0.6	Control
6	Bb	2	1/4 = 0.25	Case
7	Bb	2	1/4 = 0.25	Control
8	Bb	2	1/4 = 0.25	Control
9	Bb	2	1/4 = 0.25	Control
10	bb	3	2/3 = 0.66	Case
11	bb	3	2/3 = 0.66	Case
12	bb	3	2/3 = 0.66	Control

Bb, and bb); therefore, we need a suitable encoding method to transform the SNP data to the numerical amount. There are various encoding methods [21,22], and each technique can be suitable for specific data. The binary and one hot encoding are used in the majority of studies [1–6]. However, these encoding methods increase the number of features based on the number of attribute categories. Also, encoding methods such as one hot, binary, integer representation, etc., which are used in recent studies, do not describe any information about the difference or similarity of SNP genotypes. In this study, the mean encoding is utilized for encoding the SNP data. The mean encoding method is an intelligent method since it can consider the target label in the encoding process, whereas other encoding methods have no correlation with the target [23]. Also, mean encoding could prove to be a much simpler method in case of a large number of features. Mean encoding calculates the numerical amount for the unique nominal feature according to Eq. (1).

$$\mathit{mean\ encoding\ of\ feature\ i} = \frac{\mathit{Number\ of\ true\ targets\ under\ the\ feature\ i}}{\mathit{Total\ number\ of\ targets\ under\ the\ feature\ i}} \tag{1}$$

Therefore, we require applying the intelligent encoding method such as mean encoding, which applies useful information to label categories. In mean encoding, the algorithm assigns the quantity to each category (BB, Bb, and bb) based on the separability potential of each category in each specific data. In the raised example, the algorithm assigns the same quantity for BB and Bb that is equal to 0.6 (3 (case)/5 (instances)) in the *i*th feature, which illustrates to be BB and Bb is not significant in the case group by assigning the same quantities. Therefore, the *i*th feature can be irrelevant, and this property of the mean encoding can improve the performance of the feature selection step and classification step. Otherwise, mean encoding assigns the different quantities for BB and Bb, which illustrates to be BB or Bb in *i*th feature play a significant role in complex disease as shown. An example of a mean encoding method for SNP data as shown in Table 4. In the *i*th feature, we have five instances with BB quantity so that three of five are cases. Thus, the encoding value for the BB state will be 3/5 in the *i*th feature.

2.3. Feature selection method

Feature selection (FS) plays an essential role in machine learning and pattern recognition; FS can improve the accuracy of a classifier by removing irrelevant and redundant features. The FS algorithm tries to select a subset of features from input data that can efficiently explain the input data and despite decreasing the effects of noise (irrelevant and redundant features) still can provide good classification and prediction results.

FS methods are mainly divided into three types: filter, wrapper, and embedded. In recent years, hybrid feature selection methods are also developed, and utilized in many data [24]. Filter methods calculate the rank or weight for each feature so that the features with a high grade are more correlated with the target. They use various measures to calculate the grade, and these measures are mainly categorized into three types: the criteria based on correlation, based on the distance between distributions and based on information theory [25]. Filter methods do not need any classifier and called classifiers independent. Therefore, filter methods decrease time and computational cost, especially in high dimensional data, but investigate features individually and cannot consider the interaction between features. Wrapper methods try to select the optimal subset of features and utilize the classifier for evaluation, thus called classifier dependent. Wrapper methods are expensive from the point of view of time and computation but can investigate the interaction between features. Embedded methods carry out feature selection as part of the training process and reduce computation time. Many feature selection methods are discussed and reviewed in many studies [24–30], but we want to discover the suitable FS method, that be able to select features in data with high dimension and small samples.

In this study, the filter method is utilized for feature selection, since it is less expensive from the viewpoint of time and computation. Also, filter method based on correlation is used to feature selection, as the estimation of probability density function and entropy can be biased in high dimension sparse data (data with high dimension and small size), so FS methods based on information theory and based on distance between distributions perform better on dense than on sparse data [31].

The proposed filter method involves two steps: in the first step, the relevance is computed by using dispersion measures, and 1000 top features are selected, then in the second step the redundancy is calculated by using similarity measures for a selected subset of features in the first step. Eventually, the rank of each feature (for 1000 selected features) is calculated by Eq. (2), which is considered both relevance and redundancy effect for each feature. The first term of Eq. (2) denotes the relevance effect, and the second term illustrates the weighted sum of the redundancy effect.

$$Rank\left(x_i\right) = Relevance\left(x_i, Target\right) - \frac{1}{1000-1} \times \sum_{j=1, j\neq i}^{1000-1} Redundancy(x_i, x_j) \qquad (2)$$

The whole process of FS is shown in Fig. 3.

2.3.1. Dispersion measures

Various dispersion measures can be used to calculate relevancy [31]. We review some of the important measures, then select the best of them for SNP selection. One of the important dispersion criteria is the variance, and another one is the Mean Absolute Difference (MAD), which defined in Eqs. (3) & (4) respectively.

$$Var_i = \frac{1}{N}\sum_{j=1}^{N}(x_{ij} - \bar{x}_i)^2 \qquad (3)$$

$$MAD_i = \frac{1}{N}\sum_{j=1}^{N}|x_{ij} - \bar{x}_i| \qquad (4)$$

In Eqs. (3) & (4), x_i and $\bar{x}_i$ are the ith feature and its mean respectively. Another measure of dispersion includes the Arithmetic Mean (AM) and the Geometric Mean (GM), which defined in Eqs. (5) & (6) for a given positive feature x_i on N patterns.

$$AM_i = \frac{1}{N}\sum_{j=1}^{N} x_{ij} \qquad (5)$$

$$GM_i = (\prod_{j=1}^{N} x_{ij})^{\frac{1}{N}} \qquad (6)$$

The ratio of AM to GM can be used as a dispersion measure that is shown in Eq. (7). A higher value of R illustrates a higher dispersion so that the feature will be more relevant. Otherwise, R will be close to one.

$$R_i = \frac{AM_i}{GM_i} \in [1, +\infty) \qquad (7)$$

If feature i is zero amount, the $GM_i = 0$ (in Eq. (6)) and R_i will be useless. The other dispersion measure is introduced to address this problem and defined as AMGM [31] in Eq. (8), which has been applied the exponential function to each feature. In this study, AMGM has been used to calculate the relevance between feature and target, since this measure had an excellent performance in sparse data.

$$AMGM_i = \frac{\frac{1}{N}\sum_{j=1}^{N} exp(x_{ij})}{(\prod_{j=1}^{N} exp(x_{ij}))^{\frac{1}{N}}} = \frac{1}{N\exp(\bar{x}_i)}\sum_{j=1}^{N} exp(x_{ij}) \qquad (8)$$

2.3.2. Similarity measures

The redundancy is computed by using a similarity measure. Various similarity measures have been defined, that one of the well-known measures is Correlation Coefficient (CC) as shown in Eq. (9).

$$\rho\left(x_i, x_j\right) = \frac{\sum_{k=1}^{N}(x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j)}{\sqrt{Var\left(x_i\right)Var(x_j)}} \qquad (9)$$

There are also other similarity measures such as Maximal Information Compression Index (MICI) [32] and Symmetrical Uncertainty (SU) [33], as shown in Eqs. (10) and (11).

$$2\lambda\left(x_i, x_j\right) = Var\left(x_i\right) + Var\left(x_j\right) - \sqrt{(Var\left(x_i\right) + Var\left(x_j\right))^2 - 4Var\left(x_i\right)Var(x_j)(1 - \rho\left(x_i, x_j\right)^2)} \qquad (10)$$

$$SU\left(x_i, x_j\right) = \frac{H\left(x_i\right) - H(x_i|x_j)}{H\left(x_i\right) + H(x_j)} \qquad (11)$$

In this study, absolute cosine is used to calculate the redundancy between two features, since this measure has also succeeded in sparse data [31], which is defined in Eq. (12).

$$|Cos(\theta_{x_i,x_j})| = |\frac{\langle x_i, x_j\rangle}{\|x_i\| \,.\|x_j\|}| \qquad (12)$$

where $\langle .,. \rangle$ denotes the inner product and $\|.\|$ is the Euclidean norm. In Eq. (12), it will be $0 < |Cos(\theta_{x_i,x_j})| < 1$ which 0 and 1 meaning that the two features are orthogonal (maximum difference) and collinear features respectively.

2.4. Auto-encoder

In this study, we use deep auto-encoder to classify the SNP data. However, some principal obstacles are challenging in the field of designing auto-encoder. Specifically, the underlying theory is not well understood, and there is no clear realization of which architectures carry out better than the others. It is difficult to characterize which structure and how many layers or how many nodes in each layer are appropriate for a specific task [9]. So we introduce the new deep auto-encoder that can construct its structure according to input data, and that means it can estimate the number of nodes in each layer and the number of layers. This method decreases the time spent and the computational burden; by the way, it removes the need to try various structures with various nodes and layers in the learning procedure. It also prevents overfitting with a reasonable number of nodes and layers because we do not dedicate a high capacity to solve the classification. In this approach, we assign an appropriate capacity of the deep auto-encoder according to the complexity of the problem. Finally, the self-organizing deep auto-encoder constructs its structure automatically.

2.4.1. Basic auto-encoder (AE)

The auto-encoder was proposed as a method for reducing dimensionality [34,35]. It is a particular type of artificial neural network and is composed of two parts, an encoder, and a decoder. The output of the encoder represents the reduced representation, and the decoder reconstructs the initial input from the encoder's representation. When the decoder is linear, the loss function is the mean squared error, and under complete auto-encoder (the number of hidden units is less than the input dimension as a bottleneck shape) learns to span the same subspace as PCA (Principal Component Analysis) [34]. The auto-encoder with a nonlinear encoder activation function and a nonlinear decoder activation function can learn a more powerful nonlinear generalization of PCA [36]. These subsections introduce the variants of auto-encoders.

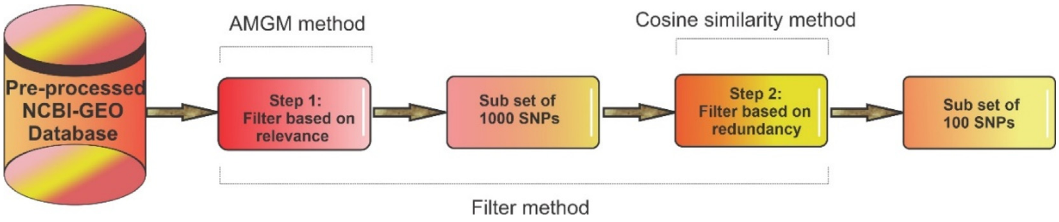

AMGM method
Cosine similarity method
Pre-processed NCBI-GEO Database
Step 1: Filter based on relevance
Sub set of 1000 SNPs
Step 2: Filter based on redundancy
Sub set of 100 SNPs
Filter method

Fig. 3. Detailed of FS stage.

We begin by introducing the classical AE model that was used in [37] to build the deep network shown in Fig. 4. The basic AE consists of two steps. The first step i.e. encoding is a function $f(x)$ that maps an input $x \in [0\ 1]^d$ to the hidden layer (or representation layer) $y \in [0\ 1]^{d'}$. It is shown in Eq. (13)

$$y = f_\theta\left(x\right) = s_f(WX + b) \tag{13}$$

s_f is a non-linear activation function. The deterministic mapping $y = f_\theta(x)$ is parameterized by $\theta = \{W, b\}$. W is a $d \times d'$ weight matrix, and b is a bias vector. The second step i.e., decoding is a function $z = g_{\theta'}(y)$ that maps back resulting latent representation to a reconstructed vector $z \in [0\ 1]^d$. It is shown in Eq. (14)

$$z = g_{\theta'}\left(y\right) = s_g(W'Y + b') \tag{14}$$

The deterministic mapping $z = g_{\theta'}(y)$ is parameterized by $\theta' = \left\{W', b'\right\}$. The weight matrix W' of the reverse mapping may be constrained by $W' = W^T$, in which case the AE is said to have tied weights.

The parameters of the AE are optimized to minimize the loss function L, as shown in Eqs. (15) and (16)

$$\theta, \theta' = \underset{\theta,\theta'}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} L(x^{(i)}, z^{(i)}) \tag{15}$$

$$\theta, \theta' = \underset{\theta,\theta'}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} L(x^{(i)}, g_{\theta'}(f_\theta(x^{(i)}))) \tag{16}$$

L can be defined as the traditional squared error $L\left(x, z\right) = \|x - z\|^2$. Another choice for loss function is cross-entropy when the used s_g is sigmoid, and the inputs are in [0 1]. The cross-entropy loss function is shown in Eq. (17).

$$L\left(x, z\right) = -\sum_{i=1}^{n} x^{(i)} \log\left(z^{(i)}\right) + (1 - x^{(i)}) \log(1 - z^{(i)}) \tag{17}$$

2.4.2. Regularized auto-encoder

Weight decay is the simplest method of regularization. This method forces the weights to be small, and it happens by optimizing the following regularized objectives instead:

$$J_{AE} + \Omega\left(\theta\right) = \left(\sum_{x \in D_x} L\left(x, g_{\theta'}\left(f_\theta\left(x\right)\right)\right)\right) + \lambda \sum_{i,j} {W_{ij}}^2 \tag{18}$$

$$J_{AE} + \Omega\left(\theta\right) = \left(\sum_{x \in D_x} L\left(x, g_{\theta'}\left(f_\theta\left(x\right)\right)\right)\right) + \lambda \sum_{i,j} |W_{ij}| \tag{19}$$

Eqs. (18) and (19) are called loss function with L2-norm and L1-norm regularization, respectively. λ Hyper-parameter controls the strength of the regularization.

2.4.3. Self-organizing auto-encoder (SOAE)

In this study, the SOAE is proposed as a new type of AE that can determine its structure according to input data. The proposed algorithm for designing AE can estimate the number of layers and their nodes; thus, the necessity of testing several numbers of structures with a different number of layers and nodes in each layer is eliminated. Therefore, this property of the proposed algorithm decreases the time spending and the computational cost. In addition, the over-fitting problem does not happen in SOAE since it does not dedicate a high capacity of a model for classification and suitable capacity of the model is utilized according to the complexity of classification. Whereas other used AEs in many applications that use high capacity of model and try to control it by using regularization methods. In recent years, Hinton et al. introduced the dropout technique to address the overfitting problem so that they remove some nodes in encoding layers based on a random function. Thus, the thinned outputs are generated by the dropout technique, which is used as input to the next layer. Our proposed algorithm is similar to the dropout technique but is different in node elimination. We apply an intelligent method to select unimportant nodes in the encoding layer, which is based on the feature selection algorithm, unlike Hinton's group that select nodes in the encoding layer based on a random function. Therefore, we can address the overfitting problem with a more reasonable method, and experimental result, involving training results and testing results as shown in Table 7, illustrates that overfitting does not happen in all of the SNP datasets. In addition, we believe that the auto-encoder structure is reasonable because the proposed algorithm performs the node selection and elimination in the encoding layer based on the right mathematics ideas and sensible selection of nodes. We utilized the proposed algorithm in our recent study, in which the architecture of Restricted Boltzmann Machine (RBM) is constructed automatically by the suggested method [38]. In this regard, we apply the same idea to Auto-Encoder (AE) with some changes to determine the number of nodes and layers automatically.

The self-organizing deep auto-encoder constructs its structure in two steps, estimating the number of nodes and layers. In the first step, AE transforms input features into new feature space in the encoding layer, in which the dimension of encoding space in more than the input space; so that the number of nodes in the encoding layer is determined according to the number of inputs, as shown in Eq. (20). n_{initial}, d_{input} and α are initial number of nodes in the encoding layer, input dimension, and a user-defined coefficient, respectively. Also, α is considered more than one ($1 < \alpha$) thus $n_{\text{initial}} > d_{input}$.

$$n_{\text{initial}} = \alpha \times d_{input} \tag{20}$$

We train AE with n_{initial} nodes in the encoding layer, using training data. Then Relief-F algorithm is applied to evaluate the generated features in the encoding layer so that it calculates the weight for each feature. The features that their weights are more than the threshold (user-defined parameter) are retained, and others are removed, as shown in Fig. 5. Therefore, we can estimate the number of nodes using the proposed algorithm in each AE. Finally, this algorithm is implemented for all of the AEs that are needed for designing a deep model. Table 5 shows the whole process in this step. In addition, this approach acts as a regularization method, because the algorithm removed the nodes

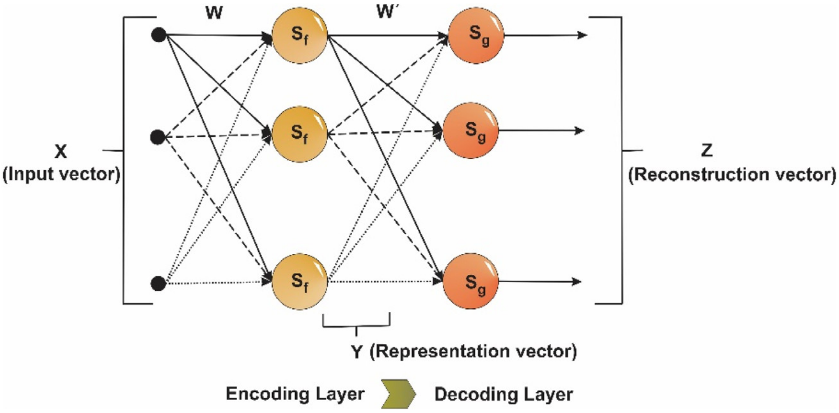


Fig. 4. The structure of auto-encoder including, encoder and decoder.

Table 5
Pseudo-code of estimation of the reasonable number of nodes.

```
Algorithm: estimation of the reasonable number of nodes

Input: all training instances, involving a vector of attributes and class value for each instance.
Output: auto-encoder with n number of nodes.
1. Set n: = primary number of nodes;
2. Train auto-encoder with n nodes using training data;
3. weights ← Relief-F algorithm (new features in the encoding layer)
4. Set Threshold: = mean of weights;
5. Keep nodes whose weights are more than the threshold and remove other nodes in the encoding layer;
6. Set n: = number of nodes that their weights are more than the threshold;
7. Auto-encoder with n nodes is ready to use;
```

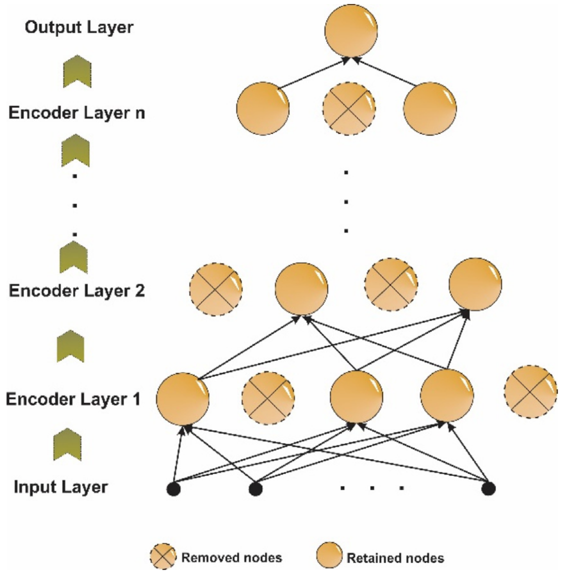


Fig. 5. The self-organizing deep auto-encoder.

whose weights are less than the threshold; thus, AE learns to copy input merely approximate.

In the second step, we add the AE layer-by-layer, in which the architecture of each AE is determined by step one. Next, we add the fine-tuning layer to each structure and compare the first structure with one AE with a second structure with two AEs. If the performance of the first structure is better than the second structure, we stop adding AEs and choose the first structure; otherwise, we continue to add more AEs and repeat step two. In addition, we utilize the training data to train the fine-tuning model and the validation data to calculate the performance of each structure. Table 6 shows the whole process in this step.

3. Experimental results

Briefly, the following steps were done for all SNP data, and then the simulation results were reported in this section.
1- The SNP data was preprocessed so that features in which the number of missing values more than the determined threshold, user-defined parameters such as 10%, was removed and also the other missing values were replaced by suitable values.
2- The SNP data were divided into three parts: namely training, validation, and test, involving 70%, 10%, and 20% of data, respectively.
3- The preprocessed SNP data were converted to numeric data using the Mean Encoding method.
4- We apply FS algorithm to SNP data (training data), in the first step, the FS algorithm calculates the relevance between each feature and target, then 1000 top features were selected, in the second step the redundancy was determined in selected features. The rank of each feature was obtained based on Eq. (2). Eventually, 100 top features (or SNPs) were chosen.
5- We apply the powerful classifier to evaluate selected features. In this step, the proposed self-organizing deep auto-encoder was used to classify SNP data based on selected SNPs. The self-organizing deep auto-encoder can determine its structure automatically so that we did not require to do a random search or grid search to construct its structure. So it has low time spending and low computational cost than the traditional method.

In all of the experiments, the sigmoid and linear functions were utilized for activation function in the encoding layer and the decoding layer, respectively. The sigmoid or soft-max function also was employed in the fine-tuning layer according to the classification, as shown in Table 7. The learning rate η was set to a variable value based on the cyclical learning rates method [39]. In the proposed algorithm, the user can control the depth and width of the self-organizing deep auto-encoder by using the suitable values for α (primary number of nodes), number of neighbors (in the Relief-F algorithm), threshold (feature selection procedure in

Table 6
Pseudo-code of estimation of the reasonable number of layers.

```
Algorithm: estimation of the reasonable number of layers

Input: all training and validation instances, involving a vector of attributes and class value for each instance.
Output: deep auto-encoder with $i$ number of layers
Set m: = maximum number of auto-encoder;
For i: =1 to m do begin
    Add $i$th auto-encoder;
    Estimation of the reasonable number of nodes;
    Add fine-tuning layer;
    Calculate MSE of the model using validation data;
    Remove fine-tuning layer;
    if $i > 1$ and MSE of structure with $i$ auto-encoder > MSE of structure with $(i-1)$ auto-encoder
        Remove $i$th auto-encoder;
        Break;
    Endif;
End;
```

Table 7
The structure of self-organizing auto-encoder and fine-tuning layer.

Dataset	Number of input	Number of layers	Number of nodes	Activation function in encoding layers	Activation function in decoding layers	Activation function in fine-tuning layer	Number of output
Thyroid Cancer	100	2	[56-2]	Sigmoid	Linear	Soft-max	2
Mental Retardation	100	2	[47-2]	Sigmoid	Linear	Soft-max	2
Breast Cancer	100	2	[56-2]	Sigmoid	Linear	Soft-max	2
Colorectal Cancer	100	2	[50-2]	Sigmoid	Linear	Soft-max	2
Autism	100	2	[41-2]	Sigmoid	Linear	Soft-max	2

Table 8
The proposed algorithm parameters for SNP datasets.

Dataset	α	Number of neighbors	Threshold	Maximum number of AEs
Thyroid Cancer	1.1	10	Mean of weights	3
Mental Retardation	1.1	10	Mean of weights	3
Breast Cancer	1.1	10	Mean of weights	3
Colorectal Cancer	1.1	10	Mean of weights	3
Autism	1.1	10	Mean of weights	3

the encoding layer), and maximum number of AEs, as shown in Table 8. Additionally, we did not apply any regularization method, because the proposed algorithm acts as a regulator method by removing the nodes, which their weights are less than the threshold. Also, we presented the architecture of the classifier model in Fig. 6, which is built by the proposed method.

6- The result of classification on SNP data (test data) indicates how much the proposed method can distinguish healthy and patient groups based on the selected SNPs. The prediction power of the self-organizing deep auto-encoder was evaluated using two measures, the accuracy, and the F-measure, which illustrates the ability of our proposed method to predict cases, as shown in Eqs. (21) & (22) respectively.

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \times 100 \tag{21}$$

$$F - measure = \frac{Precision \times Recall}{Precision + Recall} \tag{22}$$

where

$$Recall = \frac{TP}{TP + FN} \tag{23}$$

$$Precision = \frac{TP}{TP + FP} \tag{24}$$

TP, FN, FP, and TN are the number of a true positive, false negative, false positive, and false negative respectively. The proposed method was applied to all SNP data for five times and the best result was reported.

The accuracy and error rate of training data, validation data, and test data are shown in Table 9 for all complex diseases, which classified according to selected SNPs. The confusion matrix of test data is illustrated in Figs. 7–11 for all SNP data.

In addition, the accuracy, F-measure, specificity, and sensitivity of test data are shown in Table 10.

The results of the proposed method (Relevance and Redundancy Analysis + SOAE) were compared with the results of other known FS methods that were widely used in this field. The known FS methods involve the Minimum Redundancy Maximum Relevance (mRMR) algorithm [40], Relief-F [39], Fast Correlation-Based Feature Selection (FCBC) [41] and CMIM [42] that 100 top SNPs were selected according to these methods. Also, four popular classifiers contain Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Naïve Bayes (NB) and Linear Discriminant Analysis (LDA) were utilized to evaluate the selected SNPs, then the accuracy and F-measure of above combination methods for SNP datasets are illustrated in Tables 11–15.

The comparison between our proposed method and other methods (as shown in Tables 11–15) demonstrates that the proposed FS method and proposed self-organizing auto-encoder have succeeded in SNPs selection and classification with the best accuracy. For example, the obtained accuracy in Thyroid Cancer, Mental Retardation, Breast Cancer, Colorectal Cancer, and Autism datasets when our approach was used are up to 13%, 12%, 11%, 13%, 17% respectively better than the best results in other methods (are marked bold in tables).

Also, the comparison between the proposed method and different frameworks that were previously applied in the Thyroid Cancer (TC), Mental Retardation (MR), Breast Cancer (BC), Colorectal Cancer (CC) and Autism (ASD) datasets are indicated in Tables 16–20. In TC, MR, BC, CC, and ASD the obtained accuracy of the proposed method is up to 10%, 9%, 4%, 5% and 9% better than other published works respectively.

Conclusion

The human genome sequencing has obtained an excellent success in medical science, and illustrated the importance and effectiveness of genotype in complex diseases. In the current study, we tried to build a framework that has the potential to analyze SNP data. In this regard, we proposed a new method for the significant SNPs selection and classification of them in

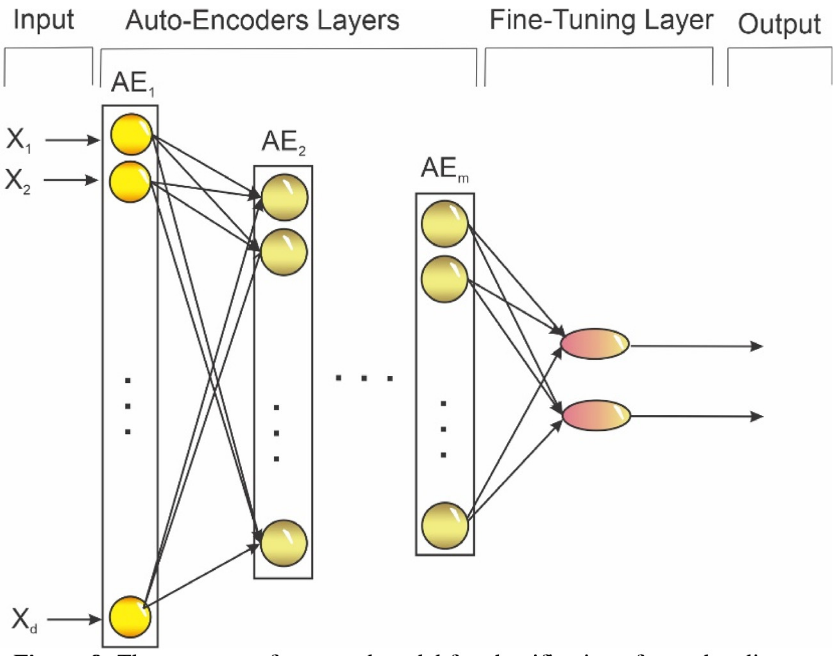


Fig. 6. The structure of proposed model for classification of complex diseases.

Table 9
The accuracy and error rate of training data, validation data and test data.

‘Dataset	Training (ACC) %	Training (Error) %	Validation (ACC) %	Validation (Error) %	Test (ACC) %	Test (Error) %
Thyroid Cancer	100	0	95.7	4.3	**100**	0
Mental Retardation	97.6	2.4	94.47	5.6	**94.4**	5.6
Breast Cancer	100	0	81.8	18.2	**100**	0
Colorectal Cancer	96.6	3.4	100	0	**96**	4
Autism	99.7	0.3	98.2	1.8	**99.1**	0.9

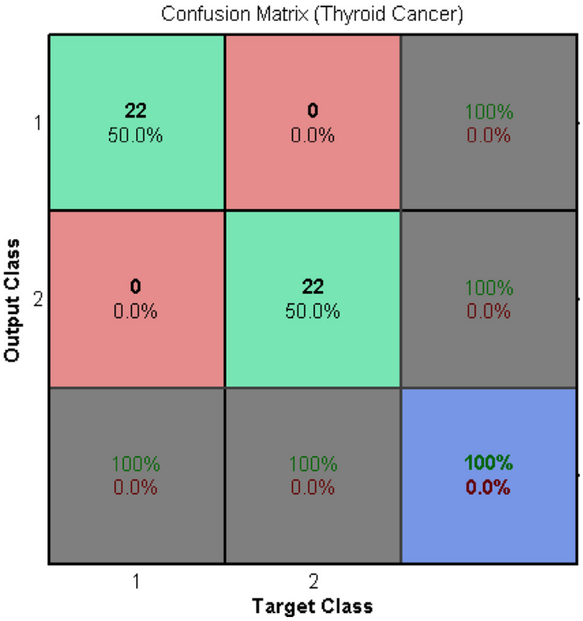


Fig. 7. The confusion matrix of thyroid cancer.

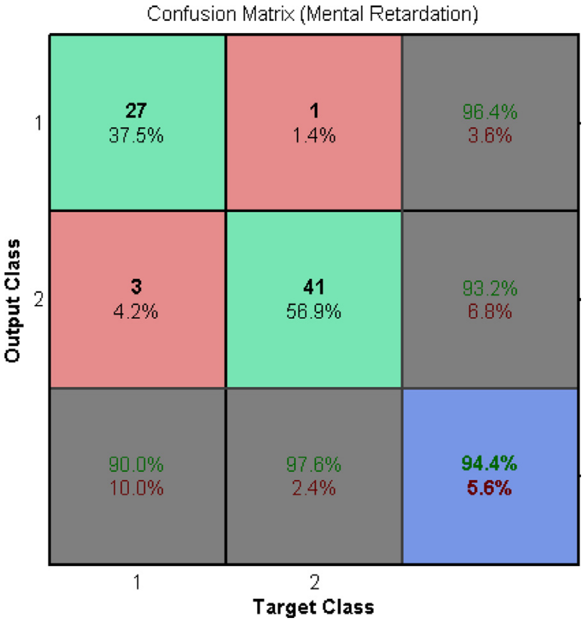


Fig. 8. The confusion matrix of mental Retardation.

complex diseases. According to our proposed method, the SNP data were preprocessed, leading to eliminate SNPs with high missing values and the other SNPs with low missing values were replaced by suitable values. The mean encoding method, as an intelligent approach due to considering the target label in the encoding process, was applied to SNP data to convert the nominal to numeric data. As followed, the two-step filter method was used to select significant SNPs, which involves relevance and redundancy calculation. In the FS method, the relevance was calculated for all SNPs, but the redundancy was computed only

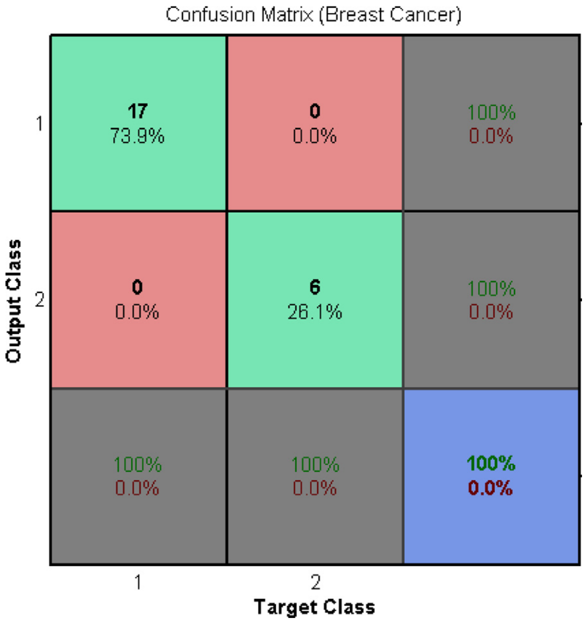


Fig. 9. The confusion matrix of test data for breast cancer.

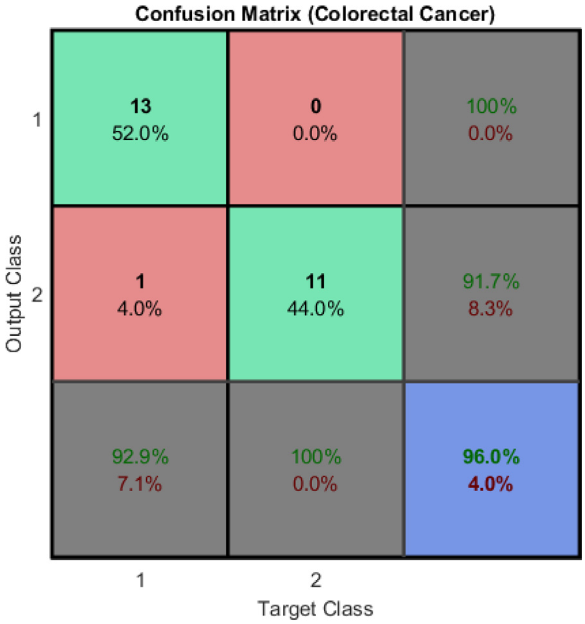


Fig. 10. The confusion matrix of colorectal cancer.

Table 10
The accuracy, F-measure, specialty and sensitivity of test data.

Dataset	Accuracy (%)	F-measure (%)	Specificity (%)	Sensitivity (%)
Thyroid Cancer	**100**	**100**	100	100
Mental Retardation	**94.4**	**93.1**	93.2	96.4
Breast Cancer	**100**	**100**	100	100
Colorectal Cancer	**96**	**96.3**	91.7	100
Autism	**99.1**	**99.2**	100	98.5

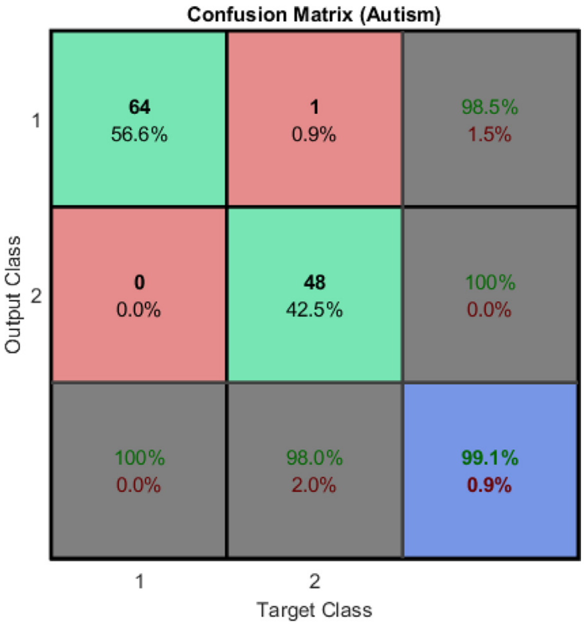


Fig. 11. The confusion matrix of autism.

Table 11
Performance of four FS methods with different classifiers to evaluate the selection SNPs for Thyroid Cancer dataset [4].

Thyroid Cancer		SVM	KNN	NB	LDA
mRMR	Accuracy	83.33	76.22	76.22	78.44
	F-measure	87.20	69.26	67.74	79.84
Relief-F	Accuracy	83.78	82.89	67.56	78.89
	F-measure	88.05	81.69	71.31	77.41
FCBF	Accuracy	85.38	84.11	75.84	79.44
	F-measure	86.59	86.20	68.76	77.84
CMIM	Accuracy	**86.44**	85.56	86.00	79.78
	F-measure	85.65	84.56	86.46	77.64

Table 12
Performance of four FS methods with different classifiers to evaluate the selection SNPs for Mental Retardation dataset [4].

Mental Retardation		SVM	KNN	NB	LDA
mRMR	Accuracy	81.67	74.72	80.00	76.67
	F-measure	76.72	69.76	81.93	73.92
Relief-F	Accuracy	65.28	66.11	65.28	63.89
	F-measure	16.49	6.90	63.16	49.80
FCBF	Accuracy	70.83	69.11	71.80	67.89
	F-measure	49.69	56.44	73.56	79.82
CMIM	Accuracy	**82.83**	69.44	81.67	70.56
	F-measure	81.88	66.31	82.99	70.45

Table 13
Performance of four FS methods with different classifiers to evaluate the selection SNPs for Breast Cancer dataset [4].

Breast Cancer		SVM	KNN	NB	LDA
mRMR	Accuracy	**89.09**	85.49	**89.09**	76.68
	F-measure	92.34	86.30	90.48	75.78
Relief-F	Accuracy	88.18	70.32	56.68	48.66
	F-measure	86.28	68.17	51.23	42.96
FCBF	Accuracy	88.28	76.11	65.28	60.16
	F-measure	92.49	76.92	60.46	49.86
CMIM	Accuracy	**89.09**	86.36	**89.09**	70.08
	F-measure	94.25	88.12	92.71	68.71

on selected SNPs with a high rank, which resulted in decreasing the time and computation cost in high dimension SNP data. Additionally, we utilized new dispersion and similarity measures, which were suitable for high dimensional sparse SNP data. Eventually, 100 top SNPs were selected and then were evaluated by using Self-Organizing Deep Auto-Encoder (SOAE) as a powerful classifier.

Table 14
Performance of four FS methods with different classifiers to evaluate the selection SNPs for Colorectal Cancer dataset [4].

Colorectal Cancer		SVM	KNN	NB	LDA
mRMR	Accuracy	79.53	59.77	63.67	57.20
	F-measure	73.17	29.29	53.89	56.24
Relief-F	Accuracy	78.00	52.40	50.00	46.90
	F-measure	69.55	7.99	46.30	51.90
FCBF	Accuracy	**82.29**	63.11	55.28	49.78
	F-measure	77.20	36.50	43.16	49.80
CMIM	Accuracy	77.10	64.50	50.83	56.43
	F-measure	70.35	38.70	41.68	55.56

Table 15
Performance of four FS methods with different classifiers to evaluate the selection SNPs for Autism dataset [4].

Autism		SVM	KNN	NB	LDA
mRMR	Accuracy	76.54	70.52	69.13	77.43
	F-measure	77.81	78.12	71.65	81.10
Relief-F	Accuracy	70.03	70.32	60.32	67.56
	F-measure	68.18	74.08	66.82	76.36
FCBF	Accuracy	74.28	76.11	65.28	73.89
	F-measure	86.49	83.92	69.16	83.80
CMIM	Accuracy	**81.50**	70.02	77.25	78.13
	F-measure	83.39	80.20	76.20	80.64

Table 16
Comparison of classification accuracy for Thyroid Cancer.

Thyroid Cancer	Number of SNPs	Accuracy	Ref.
CMIM + SVM-REF	100	90.37	[4]
Proposed method	100	**100**	

Table 17
Comparison of classification accuracy for Mental Retardation.

Mental Retardation	Number of SNPs	Accuracy	Ref.
Chi-Squared Sort + SVM	100	59	[1]
Difference Sort + SVM	100	69	[1]
Relief-F + SVM	100	76.7	[2]
RFS + SVM	100	73.58	[2]
FSDD + SVM	100	78.56	[2]
CBFS + SVM	100	86.61	[2]
CMIM + SVM-REF	100	85.00	[4]
Proposed method	100	**94.4**	

Table 18
Comparison of classification accuracy for Breast Cancer.

Breast Cancer	Number of SNPs	Accuracy	Ref.
Relief-F + SVM	100	58.00	[2]
RFS + SVM	100	51.2	[2]
FSDD + SVM	100	56.44	[2]
CBFS + SVM	100	93.46	[2]
CMIM + SVM-REF	100	96.39	[4]
Proposed method	100	**100**	

Table 19
Comparison of classification accuracy for Colorectal Cancer.

Colorectal Cancer	Number of SNPs	Accuracy	Ref.
CMIM + SVM-REF	100	90.74	[4]
Proposed method	100	**96**	

Table 20
Comparison of classification accuracy for Autism.

Autism	Number of SNPs	Accuracy	Ref.
Chi-Squared Sort + SVM	100	69	[1]
Difference Sort + SVM	100	71	[1]
CMIM + SVM-REF	100	89.50	[4]
Proposed method	100	**99.1**	

The significant feature of SOAE is that it can determine its structure (number of nodes and layers) so that it causes to eliminate the testing of various structures with a variable number of nodes and layers in the learning procedure. This advantage has decreased the time spent and the computational burden in the learning and operation phases. In all of the experiments, SOAE was designed automatically according to the SNP data. Furthermore, we assigned an appropriate capacity of the deep auto-encoder according to the complexity of the problem; therefore, any regularization method was not used in the proposed model.

In conclusion, the proposed method was applied to five different SNP datasets, in which the accuracy and F-measure were utilized to evaluate the performance of this method. The results displayed that the proposed FS approach has succeeded to identify the significant SNPs in complex diseases; on the other hand, the SOAE was able to classify the healthy and patient samples with high accuracy according to the significant SNPs.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The Cancer Control Research Center, Cancer Control Foundation, Iran University of Medical Sciences, Tehran, Iran supported this work as project No: CCF-97065. In addition, thanks are due to any ones, which their suggestions have helped to improve the quality of this paper. All authors have read and approved the submitted manuscript.

References

[1] D.T. Evans, A SNP Microarray Analysis Pipeline using Machine Learning Techniques, Ohio University, 2010.
[2] N. Batnyam, A. Gantulga, S. Oh, An efficient classification for single nucleotide polymorphism (SNP) dataset, in: Computer and Information Science, Springer, 2013, pp. 171–185.
[3] A. Boutorh, A. Guessoum, Complex diseases SNP selection and classification by hybrid Association Rule Mining and Artificial Neural Network—based Evolutionary Algorithms, Eng. Appl. Artif. Intell. 51 (2016) 58–70.
[4] R. Alzubi, N. Ramzan, H. Alzoubi, A. Amira, A hybrid Feature Selection Method for complex diseases SNPs, IEEE Access 6 (2018) 1292–1301.
[5] S. Uppu, et al., A deep learning appraoch to detect SNP interactions, J. Softw. 11 (10) (2016) (accepted), Will be published in.
[6] S. Uppu, A. Krishna, P. Gopalan, Towards deep learning in genome-wide association interaction studies, in: Pacific Asia Conference on Information System, Taiwan, 2016.
[7] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.
[8] D. Ravì, et al., Deep learning for health informatics, IEEE J. Biomed. Health Inf. 21 (1) (2017) 4–21.
[9] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, M.S. Lew, Deep learning for visual understanding: A review, Neurocomputing 187 (2016) 27–48, http://dx.doi.org/10.1016/j.neucom.2015.09.116.
[10] Y. Bengio, Practical recommendations for gradient-based training of deep architectures, in: Neural Networks: Tricks of the Trade, Springer, 2012, pp. 437–478.
[11] H. Larochelle, Y. Bengio, J. Louradour, P. Lamblin, Exploring strategies for training deep neural networks, J. Mach. Learn. Res. 10 (Jan) (2009) 1–40.
[12] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.
[13] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep Learning, MIT press Cambridge, 2016.
[14] J.M. Alvarez, M. Salzmann, Learning the number of neurons in deep networks, in: Advances in Neural Information Processing Systems, 2016, pp. 2270–2278.

[15] T. Barrett, R.J.M.i.b Edgar, Gene Expression Omnibus: Microarray Data Storage, Submission, Retrieval, and Analysis, Vol. 411, 2006, pp. 352–369, [Online]. Available: https://www.ncbi.nlm.nih.gov/geo/.

[16] B. Luzón-Toro, et al., Identification of epistatic interactions through genome-wide association studies in sporadic medullary and juvenile papillary thyroid carcinomas, 8 (1) (2015) 83.

[17] D.J. McMullan, et al., Molecular karyotyping of patients with unexplained mental retardation by SNP arrays: a multicenter study, Hum. Mutat. 30 (7) (2009) 1082–1092.

[18] M. Kadota, et al., Identification of novel gene amplifications in breast cancer and coexistence of gene amplification with an activating mutation of PIK3CA, Cancer Res. 69 (18) (2009) 7357–7365.

[19] F. Jasmine, et al., A genome-wide study of cytogenetic changes in colorectal cancer using SNP microarrays: opportunities for future personalized treatment, 7 (2) (2012) e31968.

[20] C.R. Marshall, et al., Structural variation of chromosomes in autism spectrum disorder, Am. J. Hum. Genet. 82 (2) (2008) 477–488.

[21] Z. Gniazdowski, M. Grabowski, Numerical coding of nominal data, 2016, arXiv preprint arXiv:1601.01966.

[22] K. Potdar, T.S. Pardawala, C.D. Pai, A comparative study of categorical variable encoding techniques for neural network classifiers, Int. J. Comput. Appl. 175 (4) (2017) 7–9.

[23] D. Micci-Barreca, A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems %, J. SIGKDD Explor. Newsl. 3 (1) (2001) 27–32, http://dx.doi.org/10.1145/507533.507538.

[24] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Comput. Electr. Eng. 40 (1) (2014) 16–28.

[25] W. Duch, Filter methods, in: Feature Extraction, Springer, 2006, pp. 89–117.

[26] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (Mar) (2003) 1157–1182.

[27] L. Ladha, T. Deepa, Feature selection methods and algorithms, Int. J. Comput. Sci. Eng. 1 (3) (2011) 1787–1797.

[28] C. Lazar, et al., A survey on filter techniques for feature selection in gene expression microarray analysis, IEEE/ACM Trans. Comput. Biol. Bioinform. (TCBB) 9 (4) (2012) 1106–1119.

[29] Y. Saeys, I. Inza, P. Larrañaga, A review of feature selection techniques in bioinformatics, Bioinformatics 23 (19) (2007) 2507–2517.

[30] N. Sánchez-Maroño, A. Alonso-Betanzos, M. Tombilla-Sanromán, Filter Methods for Feature Selection – A Comparative Study, in: H. Yin, P. Tino, E. Corchado, W. Byrne, X. Yao (Eds.), Intelligent Data Engineering and Automated Learning - IDEAL 2007: 8th International Conference, Birmingham, UK, December 16-19, 2007. Proceedings, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 178–187.

[31] A.J. Ferreira, M.A.T. Figueiredo, Efficient feature selection filters for high-dimensional data, Pattern Recognit. Lett. 33 (13) (2012) 1794–1804, http://dx.doi.org/10.1016/j.patrec.2012.05.019.

[32] P. Mitra, C. Murthy, S.K.J.I.t.o.p.a. Pal, and m. intelligence, Unsupervised feature selection using feature similarity, 24 (3) (2002) 301-312.

[33] T.M. Cover, J.A. Thomas, Elements of Information Theory, John Wiley & Sons, 2012.

[34] P. Baldi, K. Hornik, Neural networks and principal component analysis: Learning from examples without local minima, Neural Netw. 2 (1) (1989) 53–58.

[35] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Cogn. Model. 5 (3) (1988) 1.

[36] N. Japkowicz, S.J. Hanson, M.A. Gluck, Nonlinear autoassociation is not equivalent to PCA, Neural Comput. 12 (3) (2000) 531–545.

[37] Y. Bengio, Learning deep architectures for AI, Found. Trends Mach. Learn. 2 (1) (2009) 1–127.

[38] S. Pirmoradi, M. Teshnehlab, N. Zarghami, A. Sharifi, The self-organizing restricted Boltzmann machine for deep representation with the application on classification problems, Expert Syst. Appl. 149 (2020) 113286, http://dx.doi.org/10.1016/j.eswa.2020.113286.

[39] M. Robnik-Šikonja, I. Kononenko, Theoretical and empirical analysis of ReliefF and RReliefF, Mach. Learn. 53 (1–2) (2003) 23–69.

[40] P. Hanchuan, L. Fuhui, C. Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005) 1226–1238, http://dx.doi.org/10.1109/TPAMI.2005.159.

[41] L. Yu, H. Liu, Feature selection for high-dimensional data: A fast correlation-based filter solution, in: ICML, Vol. 3, 2003, pp. 856–863.

[42] F. Fleuret, Fast binary feature selection with conditional mutual information, J. Mach. Learn. Res. 5 (Nov) (2004) 1531–1555.