Neural Comput & Applic (2018) 29:1015–1037
https://doi.org/10.1007/s00521-016-2513-3

ORIGINAL ARTICLE

Evolutionary framework for coding area selection from cancer data

Sarwar Kamal[1] · Nilanjan Dey[2] · Sonia Farhana Nimmy[3] · Shamim H. Ripon[1] · Nawab Yousuf Ali[1] · Amira S. Ashour[4] · Wahiba Ben Abdessalem Karaa[5] · Gia Nhu Nguyen[6] · Fuqian Shi[7]

Received: 23 May 2016 / Accepted: 20 July 2016 / Published online: 2 August 2016
© The Natural Computing Applications Forum 2016

Abstract Cancer data analysis is significant to detect the codes that are responsible for cancer diseases. It is significant to find out the coding regions from diseases infected biological data. The infected data will be helpful to design proper drugs and will be supportable in laboratory assessments. Codes bear specific meaning on various features as well as symptoms of diseases. Coding of biological data is a key area to get exact information on animals to discover the desired medicine. In the current work, four different machine learning approaches such as support vector machine (SVM), principal component analysis (PCA) technique, neural mapping skyline filtering (NMSF) and Fisher's discriminant analysis (FDA) were applied for data reduction and coding area selection. The experimental analysis established that the SVM outperforms PCA and FDA. However, due to the mapping facility, NMSF outperforms SVM. Thus, the NMSF achieved the preeminent results among the four techniques. Matthews's correlation coefficient was used to evaluate the accuracy, specificity, sensitivity, *F*-measures and error rate of the four methods that are used to determine the coding area. Detailed experimental analysis included comparison study among the four classifiers for the deoxyribonucleic acid dataset.

Keywords Principal component analysis (PCA) · Support vector machine (SVM) · Neural mapping skyline filtering (NMSF) · Fisher's discriminant analysis (FDA) · Cancer DNA dataset · Matthews's correlation coefficient (MCC)

1 Introduction

All cancers instigate in cells that endure selection/mutation process. It starts with alterations in one cell/a small group of cells. Generally, in the cancer cell, most mutations DNA occur in the noncoding regions in the genome length.

✉ Amira S. Ashour
amirasashour@yahoo.com

Sarwar Kamal
sarwar.saubdcoxbazar@gmail.com

Nilanjan Dey
neelanjan.dey@gmail.com

Sonia Farhana Nimmy
snimmy.1984@gmail.com

Shamim H. Ripon
dshr@ewubd.edu

Nawab Yousuf Ali
nawab@ewubd.edu

Wahiba Ben Abdessalem Karaa
wahiba.bak@gmail.com

Gia Nhu Nguyen
nguyengianhu@duytan.edu.vn

Fuqian Shi
sfq@wmu.edu.cn

[1] Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh

[2] Techno India Institute of Technology, Kolkata, India

[3] Notre Dame University, Dhaka, Bangladesh

[4] Department of Electronics and Electrical Engineering, Faculty of Engineering, Tanta University, Tanta, Egypt

[5] Department of Computer Science, Taif University, Taif, Kingdom of Saudi Arabia

[6] Duy Tan University, Danang, Vietnam

[7] College of Information and Engineering, Wenzhou Medical University, Wenzhou, People's Republic of China

Predicting the noncoding DNA impact remains tremendously difficult. Cancer cells may endure alterations or loss in their functions. The DNA sequencing is a high-throughput approach to measure the biological expression that consists thousands of gene codes concurrently. The DNA sequence includes a set of codes that labeled the thousands of gene expression regions. These coding regions make the DNA spots. Each DNA spot represents individual gene that contains the multiple identical DNA strands. The labeled codes find and stick their perfect matching for biological data. This requires big data processing, which is one of the popular and significant procedures in computational biology. There are several numbers of codes in DNA, ribonucleic acid (RNA) and proteins. These codes carry all the features and information of human, insects, plants, fishes and paddies. Human diseases analysis is one of the fundamental aspects of civilized life. Here, cancer datasets are used to detect the codes that are responsible for human genomes.

The basic idea is to identify complementary base pair to measure the different types of DNA synthesis sequencing and expression code of gene [1, 2]. Presentation of the DNA coding data for disease classification based on different pattern of gene expression has an impact on medical research. The DNA sequencing technology helps the researchers to know the various types of diseases related to gene, including cancer, heart disease and mental disease. Moreover, DNA allows the researchers to identify all differences between any two different cells types, e.g., normal (healthy) and diseased (cancer).Scientists have classified the cancers based on gene pattern of the tumor cancer cells activities that has become easy due to coding regions of DNA. Therefore, DNA coding regions finding and analysis are important for diagnosis and pathogenic mechanism disease-related gene. These biological operations raise huge challenges to the researchers in statistical and computational environments due to high-dimensional and low sample nature of biological data. Fuzzy standard additive model (FSAM) [3] concentrates to find out the cancer responsible codes from DNA dataset. The genetic algorithm (GA) incorporated with FSAM approach to enhance FSAM learning process for generating rules that enhanced the efficiency of the FSAM approach with high-dimensional low sample data.

The DNA sequences or microarray data often contain small in size or large number of samples(ten to thousands).This is responsible for variations of codes and noncoding part of the collected DNA sequences [4, 5]. Many researchers demonstrate the classification accuracy that increased with the increasing data dimensionality. Conversely, the classification accuracy decreases with the decrease in the data dimensionality. Thus, the learning process of gene sequencing depends on the classification process providing high throughput for coding area analysis. Therefore, classification ensures the best accuracy for large biological sample. Nevertheless, large sample data require more accurate learning approach for handling increased number of biological features. So, it is difficult to obtain large biological sample because of high code selections experimental costs. Moreover, the DNA contains several many redundant and irrelevant genes that increase the feature dimensionality and decrease the learning/classification accuracy. These redundant and irrelevant data increase the noise probability and affect the construction and classification model results. Therefore, it is critical to reduce the adverse effect and to reduce the dimension of the feature space by removing the irrelevant and redundant data from the original dataset. Consequently, data reduction effectively and efficiently improves both the gene classification without filtering approach and the performance of gene selection method.

Code findings and gene selection refer to the process of removing redundant and irrelevant data and classify microarray data based on features of genes. Most common code and gene selection approaches are based on ranking [6]. Each gene evaluated individually and assigned a correlation score with different class with their certain criteria. Genes are ranked based on their score, and then, the top scored genes are considered to be feature genes. For feature gene selection, rank based includes t test [7], χ^2 test [8], information gain [9, 10], threshold number of miss classification [11], feature filtering [12], Relief [13] and entropy [14]. Additionally, mutual information (MI) is widely used for effective gene selection classification approach [15]. The MI approach is applied to measure the information of random variable that contains the information of another random variable. The MI analysis shows the degree of linear and nonlinear dependency between the random variables. In MI computing, probability distribution and joint distribution for every random variable are essential. The MI approach estimates the joint probability for variable features. If the feature is discrete, histogram is used to estimate feature probability. The probability is measured from relative frequency of features samples. If the features are continuous, two methods are used to estimate the features probabilities, namely the Parzen window and the discretization. Parzen window [16] is used to estimate the features information, but it does not estimate accurate probabilities for high-dimensional multivariate density samples and estimation cost is very high. Basically, multivariate density sample is spare distribution, so it is difficult to estimate accurate probabilities. Biological DNA data are continuous and spares, and it leads to the limitations of Parzen window approach. The other approach is discretization [17] that partitions the domain features into several subparts. Discretization process may loss some

information from the original datasets, which affects the classification accuracy due to the information loss of features and not all information is fully utilized. In order to address this problem, neighbor mutual information (NMI) approach [18] is used to measure the relevance between continuous genes and discrete features. The NMI approach is constructed by integrating the concept of neighborhood information theory and natural generalization information of numerical feature spaces.

Consequently, in the present work, the principal component analysis (PCA), support vector machine (SVM), Fisher's discriminate analysis (FDA) and mapping-based neural skyline filter (MNSF) are utilized to maintain efficient code finding. Confusion matrix is also obtained to check the specificity, sensitivity, accuracy, *F*-measures among all the classifiers implemented. Among all the machine learning approaches used in this research activity, support vector machine outperforms principal component analysis and Fisher's discriminate analysis. The main SVM advantage is that it marginalizes the related dataset in a specific regions that are absent in other two methods. This area is called the maximum margin hyperlength, which helps to process data faster due its shorter area of key data points. Threshold values are used for certain area selections in the SVM from collected cancer dataset. Neural skyline filtering is a dynamic process that controls the cancer code predictions under the artificial neural network (ANN). Mapping-based ANN clusters the whole DNA sequences into certain groups with limited segments. These limited segments help to process the cancer codes from large DNA sequences. For mapping, mathematical arrangements assess all the training datasets.

The organization of the remaining sections is as follows. Literature review and the methodology are narrated at Sects. 2 and 3, respectively. Section 3 describes the PCA for DNA codes finding, the SVM, FDA and the MNSF. Section 4 presents the results and implementation, which is followed by the conclusion in Sect. 5.

2 Literature review

Classification problems aim to build an effective and efficient model for predicting class members. The learner performs training for the data that selected from input space and their class interval. A building hypothesis not only classifies on the training data, but also predicts accurate output for unseen data. Binary classification refers to classification problems that consist of two classes, while multiclass classifier refers to the existence of more than two classification labels. Real problems are multiclass classifiers with complex classification approach. On the other hand, binary classifier is simple and easy to classification. Multiclass classifier is roughly divided two types. First one is the binary classification approach that directly extends to handle the multiclass problem, such as the discriminate analysis, regression and decision tree [19].The second type is the decomposition of multiclass classifier, such as the one-versus-the-rest method [20], pairwise comparison [21], error-correcting output coding [22] and multiclass objective function [23]. There are two prior works for multiclass classification. A comparative heuristic approach [24] is used gene classification for two datasets. Discriminate classification method [25] is used tumor cell classification for multiple datasets.

Recently, a significant number of research activities have been done on cancer disease identifications. Most of the studies were performed on the chemical-based molecules. Chemical centric analysis is good for small numbers of DNA sequences. Automated approach for cancer codes finding is essential [26]. In order to ensure proper treatments for cancer patients, set of biomarkers have been used and proposed. These markers are expensive and very much sensitive to the environments. These biomarkers are genomic, proteomic, metabolomics, imaging and psychological factors. Among all cancer types, codes are considered to be the key for all controlling aspects. Several gene selection methods have been used for cancer identification. However, these are not essential for complete process. Various cancer organizations have initiated couple of projects for cancer investigations. Large-scale dataset handling techniques such as next-generation sequencing and microarray have imposed to measure the DNA copy number alterations, messenger RNA formations and DNA mutations. However, all of these methods are not in machine learning environments [27–30].

Laplace naïve Bayes [31] model for microarray data classification focused on the robustness of gene outliers. Gene pair combination inputs [32] are used for cancer classification algorithm rather than gene original profiler. Supervised and unsupervised approaches [33] are used for microarray gene classification. Supervised classification classified the tissues based on specific gene, and unsupervised techniques classified the gene based on tissues. Computational protocols [34] used gene markers for various cancer tissues. Under sampling method [34] was used for the idea of ant colony optimization to classify imbalanced microarray data analysis. Association rules [35] were also used for gene classification, but it required system complexity enhancement. The author suggested that the transcript expression interval demonstration discriminates subtype in the same class. A Web-based interactive tool [36] is used to assess the discriminate of hypothesis performance of biological gene datasets. The tool is able to evaluate for medical diagnosis and management decision. Many methods and classification approaches are used to

classify microarray data classification. These approaches are applicable and comprehensive for clinical and real practice. The behavior of classification rules is also used for biological data size [37].

3 Methodology

The present work is an integrated environment that mechanically finds the coding area from cancer infected DNA sequences. These codes are responsible for cancer and help to design drugs for cancer removal diseases. These datasets are used to detect the codes that are responsible for human genomes. Real-world datasets have been collected from ICDDRB (International Centre for Diarrhoeal Disease Research, www.icddrb.org), Bangladesh. These data are verified by the experts around the globe. The proposed integrated environment approach is illustrated in Fig. 1.

Figure 1 depicts that the PCA was initially checked the whole data for dimension reduction to find the desired DNA codes. Each method determined new codes that PCA was unable to detect. From the PCA code and the newly measured codes, mapping-based neural skyline filter counted the maximum number of codes in short computation time. The integrated environment in Fig. 1 demonstrated the training DNA sequences process being collected from real-world DNA databases in the first step. Biological raw datasets were collected from the NCBI database that the ICDDRB used. In these databases, various types of biological data are available. There were some noise types in the collected datasets including symbols as well as characters. In the second phase, PCA was used to reduce the irrelevant dimensions of the dataset. Here, noise and other symbols were removed and cleared by the global processing of PCA. The FDA, SVM and mapping-based neural skyline filtering were imposed to verify the whole dataset. Only mapping centric neural skyline filtering permitted the parallel data processing. Typically, these mapping consumes less time for codes finding in the real-world DNA sequences. Through the next step, the whole datasets are transformed into specific frameworks to reduce the spaces as well as time. This is known as the evidential reasoning, which individually checked the entire datasets with a specific threshold value. It took half time than that of Markov chain as well as maximum likelihood estimation. The comparative outcomes of each and every method are essential to verify; thus, the Matthews's correlation coefficient (MCC)-based performance evaluation was calculated. This is the final step of the proposed framework. More detailed description of the proposed approach methodology is as follows.

3.1 Principal component analysis

Principal component analysis (PCA) is broadly used to reduce the dataset dimensionality by summarizing the data from many variables to a minimal amount of variables in such a way that the present component has the maximum variance than the upcoming one. Therefore, principal component is the first component that will have the maximum variance than the others and the covariance of any component is zero.

$$\text{Total variance} = \sum_{i=1}^{n} \text{variance}(i) \tag{1}$$

where n refers to the variables. The total variance in principal component analysis refers to the sum of the

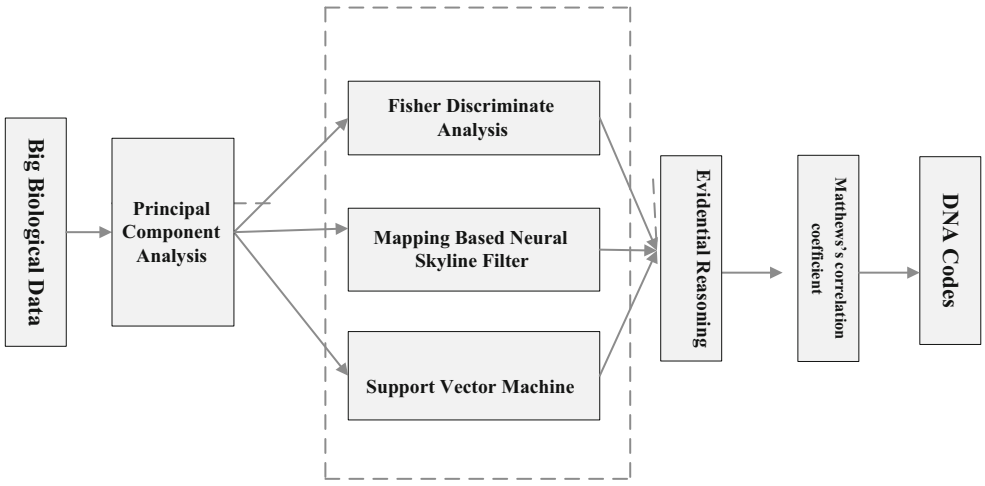

Fig. 1 Integrated method for codes finding system

variances of the observed data. The observed data are standardized, where all variables have a mean of zero and variance of one. The PCA is considered a preprocessing step for analyzing as it determines linear relations among the variables. It disperses the dataset into principal components (PCs), which are linear combinations of original measurements.

For large DNA dataset, assume z DNA base pair in each sequence with w dimensions. Globally, the whole dataset is merged with D^w. The DNA dataset is accomplished with their mean, eigenvalues, eigenvectors and covariance of the collected dataset. Each $\overline{d}$ denotes the mean of a small group. In this regard, the mean value of all DNA base pairs in a certain DNA sequence is given as follows:

$$\overline{\mu} = \frac{1}{z}\left(\overline{d_1} + \overline{d_2} \cdots \overline{d_z}\right). \tag{2}$$

The mean in Eq. (2) defines the changes coding and noncoding areas of DNA base pair. However, all the changes are not equal due to the massive changes of the DNA dataset. The variance of whole DNA sequence is given by:

$$\mathrm{Var}(\text{Whole DNA}) = \frac{1}{z-1}(d_1 - \overline{\mu_1})^2 + (d_2 - \overline{\mu_2})^2 + \cdots + \left(d_z - \overline{\mu_z}\right)^2. \tag{3}$$

This calculated variance can assist to get accurate outcomes of single code. In order to adjust the value of mean and variances of the training DNA sequences, it is required to redirects the mean to zero. Afterward, each redirected mean is subtracted from each DNA base pair d. The outcomes are categorized into a $w \times z$ matrix that is given by:

$$M = [d_1 - \overline{\mu}| \cdots |d_2 - \overline{\mu}]. \tag{4}$$

The covariance of the whole training dataset of DNA sequence that covers efficient codes regions in terms of matrix evaluation is expressed as follows:

$$S = \frac{1}{z-1} MM^{\mathrm{T}}. \tag{5}$$

This analysis enables symmetric matrix evaluations for the test and training data. In order to examine the quality of two matrices, thus, both of them should have nonzero eigenvalues. So, the MMT and MTM share nonzero eigenvalues. Suppose β is a eigenvector of MTM with eigenvalues λ not equal zero. Hence, the interchanges produce the same values for both matrices as follows:

$$(M^{\mathrm{T}}M)\beta = \lambda\beta \tag{6}$$

$$MM^{\mathrm{T}}(M\beta) = \lambda(M\beta). \tag{7}$$

Since, the PCA applies orthogonal matrix during the execution of eigenvalues measurements. These eigenvalues reduce data redundancy from large training data. In addition, the PCA has several advantages including (1) it has fewer complexities in grouping the datasets, (2) it stores only trainee sample in its database; thus, it takes small space in database, (3) it reduces the noises and irrelevant factors from DNA sequences due its small changes in collected datasets, (4) it does not consider primary class format, (5) it requires small computational analysis, and (6) it checks the grater class variations. Consequently, the PCA approach is employed in the current work.

3.2 Fisher's discriminate criterion

Several methods can be used for DNA code selection from large datasets [38, 39]. One of the powerful adaptive learning approaches for DNA code selection techniques is the Fisher's discriminate analysis (FDA). It achieves good performance for classification by using covariance matrix among the groups [40]. The FDA has used for code finding that classifies the datasets into different class groups. In massive processing, data are selected according to the specified data of interest. The FDA is classified the training dataset into two different approaches: bidirectional and global approaches. FDA can cover both local and global datasets. When, the FDA is applied for local DNA training dataset, it can compute the codes. However, for local training data, the FDA is costly in terms of time and space.

3.2.1 Bidirectional approach

The FDA was introduced by Fisher [41] for two classes to transform multivariate observations x to uni-varied observations y. The uni-varied observation y is classified into different groups that are derived from the two possible classes. Suppose there is a set of z samples $d_1, d_2, \ldots, d_n$ that belong to two different classes' c_1 and c_2. The scatter matrix for the two classes is given by:

$$SM_i = \sum_{x \in c_i} \left(d - d_i'\right)(d - d)' \tag{8}$$

where $d_i' = \frac{1}{z}\sum_{d \in c_i} d$ and m_i is the number of sample of c_i. The total inter-class scatter matrix is given by:

$$S^{(w)} = SM_1 + SM_2 = \sum_{i=1} \sum_{d \in c_i} \left(d - d_i'\right)\left(d - d_i'\right)'. \tag{9}$$

The inter-class scatter matrix is given by:

$$S^{(b)} = (d - d)(d - d)'. \tag{10}$$

The linear discriminant analysis (LDA) is a generalization form of Fisher's linear discriminant. It is used to find the projected vector W that maximizes the fisher separation criteria, which is expressed by:

$$J = \frac{|W^T S^b W|}{|W^T S^W W|}. \quad (11)$$

To determine the value of W, the eigenvalues problem of $S^b W = \lambda S^w W$ with its eigenvalue was generalized. Assume here n number of the original feature set to be $\{f_1, f_2, \ldots, f_n\}$. Then, the feature selection is required to select certain number of feature d, $F_d = \{f_{d1}, f_{d2}, \ldots f_{dn}\}$ from the original features that have the largest fisher's selection value. Here, *d(i)* is the selected feature index in the features' subset. The selected feature set F_d was denoted for the class scatter and within class scatter as $S^b(f_d)$ and $S^w(f_d)$, respectively. The fisher selection criterion $J(F_d)$ is based on separation criterion using the formula:

$$F_d = \text{maxarg}(J(F_d)) \quad (12)$$

where $J(F_d) = J(F_1, F_2, \ldots, F_d)$ is defined as:

$$J(F_d) = \frac{|W^T S^b(F_d) W|}{|W^T S^W(F_d) W|}. \quad (13)$$

3.2.2 Global approach

The global approach is used when the training DNA sequence contains more than two classes. Thus, Fisher's linear discriminate will be global discriminate analyses (GDA) [42]. However, the maximum value is computed for several competing classes. The intra-class matrix for n classes is calculated by:

$$S^{(w)} = SM_1 + \cdots + SM_n = \sum_{i=1}^{n} \sum_{x \in c_i} (d - d)\left(d - d_i'\right)' \quad (14)$$

The inter-class scatter matrix is computed by:

$$S^{(b)} = \sum_{i-1}^{z} z\left(d - d_i'\right)\left(d - d_i'\right)' \quad (15)$$

where z_i is the training sample for every class, d_i' is the mean of every class and d' is total mean vector expressed by $d' = \frac{1}{z}\sum_{i=1}^{w} z_i d_i'$. After obtaining $S^{(w)}$ and $S^{(b)}$, the linear transformation W can be calculated by generalized the eigenvalue problem:

$$S^b W = \lambda S^w W. \quad (16)$$

By solving the eigenvalue problem, the data were classified into multiple classes. Once the transformation W is obtained, the classification is performed based on distance matrix that is calculated by the Euclidian distance using the following expression:

$$\text{dist}(d, y) = \sqrt{\sum_{i=1}^{n} (d_i - y_i)^2}. \quad (17)$$

In addition, the cosine distance is used:

$$d(x, y) = 1 - \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}. \quad (18)$$

If any new instance Z is arrived, it is classified as new class that is generated by:

$$\arg\min\left(ZW, x_k' W\right). \quad (19)$$

where x_k' is the central point in kth class. Generally, the data are classified based on the centre classified point.

3.3 Support vector machine

Machine learning is a subfield of artificial intelligence and different statistical methods. Different supervised and unsupervised learning techniques, such as support vector machine (SVM), self-organization map (SOM), are used for classification and regression [43].The SVM is a classification and regression approach that maximizes the prediction accuracy and avoiding data overfitting. It can be defined as a system in which a linear function is used as a hypothesis to minimize the classification errors. It is used a kernel function that is solved by the quadric programming for hypothesis searching. All hypothesis space identifies maximum matching hyperplane (MMH) that classifies the best and almost correct data that are demonstrated in Fig. 2. Length of the MMH depends on the threshold values used in the data integration. For big dataset, the threshold value has maximum value of 1, while in another cases, this limit is always <1.

Suppose m number of datasets represented by a matrix X with ith row $A_i(1, 2, \ldots, m)$. Let $y_i \in \{1, 1\}$ belongs the classes of datasets. The hyperplane of SVM can be expressed by:

$$y_i(x_i w + b) \geq 1 \quad (20)$$

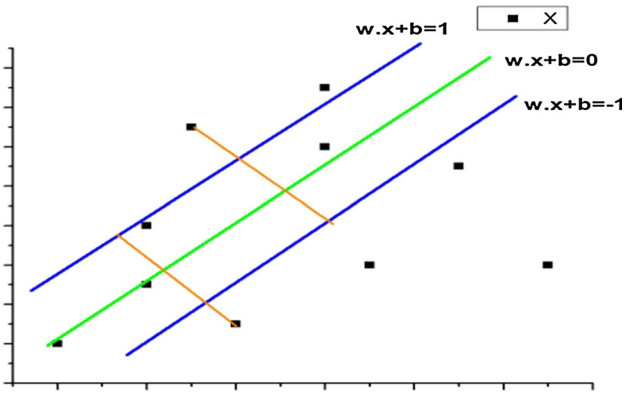

Fig. 2 Representation of hyperplane of SVM

where w is weighted vector and b is the constant for a linear equation. The hyperplane of the SVM described by $w^{\mathrm{T}}x + b = 0$ lies in midline between the boundary hyperplane described by $wx^{+} + b = 1$ and $wx^{-} + b = -1$. The distance of a point x is calculated by a distance function:

$$\text{matching} = \underset{x \in D}{\overset{\text{argmind}(x)}{}} = \text{argmin}\frac{|w \cdot x + b|}{\sqrt{\sum_{i=2}^{d} w_i^2}}. \tag{21}$$

Distance of the closest point on hyperplane to the origin can be found by maximizing the data point x on the hyperplane. Similarly, for the other side points the similar distances were calculated. Maximum margin is calculated by subtracting the two distances from the separating hyperplane to the nearest points. Subtracting distances is $w(x^{+} - x^{-}) = 2$.. The maximum matching *hyperplane* is given by:

$$M = \frac{w(x^{+} - x^{-})}{\|w\|} = \frac{2}{\|w\|}. \tag{22}$$

3.4 Map-centric neural approach

Neural network-based filtering suffers for probabilistic independent values. The Naive skyline measures the interactions of the dataset by using the conditional probabilities. On the other hand, the neural skyline filtering measures the interactions by back propagations. Probability approaches suffer from uncertainty due to the long sequences. However, the back propagation adjusts the error rates and uncertainty repeatedly by certain mathematical formula for big dataset. Hence, the neural skyline filtering achieves better options to measure accurate coding area. The back propagation iteration for codes identification is conducted for the mapping-based neural skyline filtering. Most of the cases, large dataset analysis generates greater percentages of errors due to lack of appropriate interpretations to all datasets. Another noticeable problem is that it consumes excessive time computing even for small number of DNA sequences. Figure 3 illustrates the map centric (featured neural approach) that can overcome these drawbacks. Mapping facilities organize the datasets in specific format. Certain randomized algorithm is used to arrange the mapping. This will be applicable when the total dataset is in the skyline region [44–49]. Naïve skyline measures the interactions of the dataset by using the conditional probabilities. On the other hand, neural skyline filtering measures the interactions by back propagations. Probability approaches suffer uncertainty due to the long sequences. However, back propagation adjusts the error rates and uncertainty repeatedly by certain mathematical formula for big dataset. So neural skyline filtering has better options to measure accurate coding area.

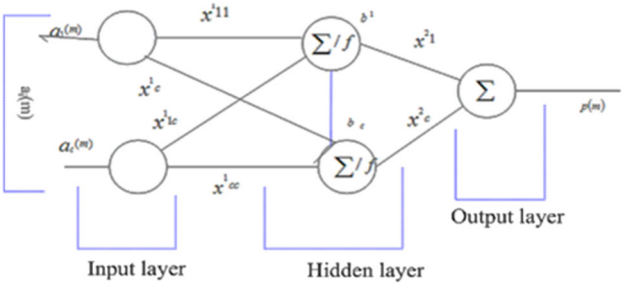

Fig. 3 Structure of feature neural skyline filter

Mathematically, the input layer processes the datasets S_{train} with associated features c faster than any other filtering by covering more datasets as follows:

$$a'(m) = [\min(a_1(m), \ldots a_c(m)), \text{sum}(a_1(m), \ldots a_c(m))]. \tag{23}$$

Here, $\min(\cdot)$ is for the first node and $\text{sum}(\cdot)$ is for the second node. Here, m_i indicates the total DNA segments under training datasets. This was repeated for rest of the input nodes. Subsequent layer (hidden layer) maintains complete relationships with a specific function called hyperbolic function using the following equations [40–51]:

$$h_i(m) = \sum_{j=1}^{2} x_{ij}^{1} a'(m) + b_i \tag{24}$$

$$\text{Result}_i^{(2)} = \frac{\text{incre}(h_i(m)) - \text{incre}(-h_i(m))}{\text{incre}(h_i(m)) + \text{incre}(-h_i(m))} \tag{25}$$

$$p(m) = \sum_{i=1}^{2} x_i^2 \text{Result}_i^2(m). \tag{26}$$

Figure 4 deploys the map-based neural approach.

Incre is a method that adjusts the weights for the desired code findings. $P(m)$ is the probability estimation functions that measures the chances to become a code in a segments. The flowchart represented in Fig. 4 included the following systematic processes:

1. Map datasets according to their priority [52–56].
2. Proper associations with input datasets and training datasets [57–60].
3. Conversion of the input training data y (neural approach) to p_f. The map-centric approach of the data is performed using the following expressions:

$$y_{f_{ij}}^{1} = \left[\min\left(f_j - \frac{i}{z}(f_i - \phi^1)\right), sum\left(f_j - \frac{i}{z}(f_j - \phi^1)\right)\right] \tag{27}$$

$$y_{f_{ij}}^{2} = \left[\min\left(f_j + \frac{i}{z}(\phi^2 - f_j)\right), sum\left(f_j + \frac{i}{z}(\phi^2 - f_j)\right)\right] \tag{28}$$

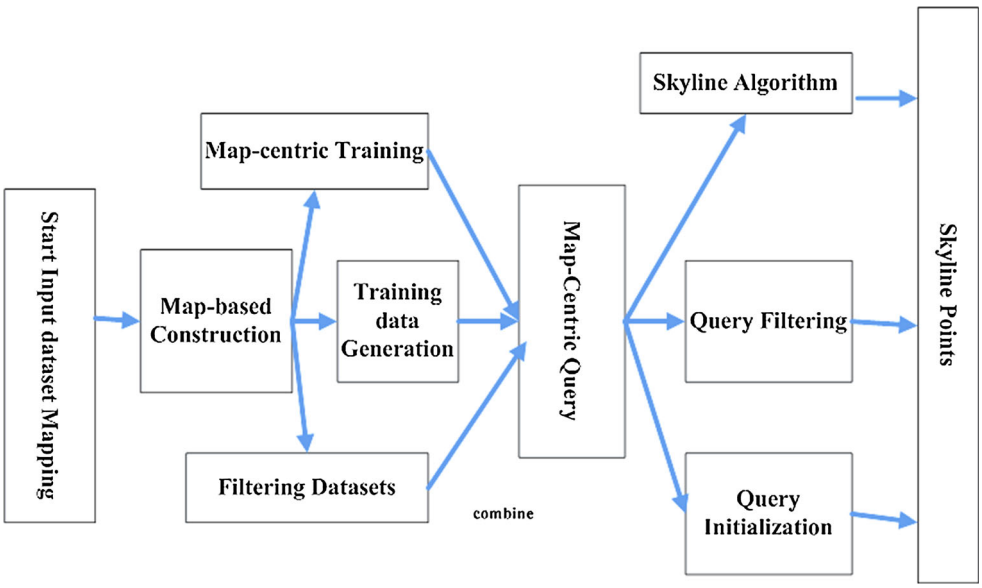

Fig. 4 Flowchart representing map-centric neural skyline filter

4. Finally, the queries have been imposed for mapped datasets [61–63] using the following expression:

$$S_{Train}(m) = [\min(S(m)), Sum(S(m))] \tag{29}$$

$S(m)$ is a method that computes the summations of whole probability.

3.5 Data transformation by using evidential reasoning approach

Data transformation is imperative for large biological data analysis. The system complexity becomes high when data size is large. For that, data transformation approach is used to reduce the data size for next size biological data operation. In the present work, data transformation approach was carried out using the evidential reasoning approach (ER). The ER approach covers the whole dataset with multidimensional weight, while the Bayesian reasoning covers the weight by single operational probability. As a result, the scope of Bayesian reasoning is small and slow. The ER approach can be used to generate rules from sample biological data. It transforms the biological data based on degree of belief β_{ij} and all $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, M$. The basic probability mass is assigned for gene expression data by using the following ER approach [1, 2]:

$$p_{ij} = w_i \beta_{ij}, \quad i = 1, 2 \ldots N \tag{30}$$

$$p_{L,j} = 1 - \sum_{i=1}^{N} p_{ij} = 1 - w_i \sum_{i=1}^{N} \beta_{ij} \tag{31}$$

$$\bar{p}_{L,j} = 1 - w_i \tag{32}$$

$$\tilde{p}_{L,j} = 1 - w_i \left(1 - \sum_{i=1}^{N} \beta_{ij}\right) \tag{33}$$

where $p_{L,j} = \bar{p}_{L,j} + \tilde{p}_{L,j}$, $j = 1, 2, \ldots, M$ and the total weight $\sum_i^M w_i$. The probability mass assigned for the input biological dataset L. The unassigned probabilities before biological data transformation are split into two parts: one cause is relative importance of J biological dataset $\bar{m}_{L,j}$ and other cause due to incompleteness of the biological dataset $\tilde{m}_{L,j}$. Data incompleteness arises for the presence of spurious datasets and noisy biological data elements.

Aggregation of all input biological data L generated the output biological dataset D. Degree of belief is assigned for every input dataset $(I_1, \ldots, I_k)$ and generate the output datasets $(O_1, \ldots, O_k)$.Suppose $m_{J,I}(k)$ is assigned degree of belief and $m_{D,I}(k)$ is unassigned degree of belief for the output datasets. Then, the overall degree of belief β_j in D_j is calculated as:

$$\{L_j\} : p_{j,I(k+1)} = K_{I(k+1)} \left[p_{jI(k)} p_{j,k+1} + p_{jI(k)} \times p_{L,k+1} + p_{L,I(k)} p_{j,k+1}\right] \tag{34}$$

$$p_{L,j(k)} = \bar{p}_{L,j(k)} + \tilde{p}_{L,j} \quad k = 1, 2, \ldots, N \tag{35}$$

$$\tilde{p}_{L,I(k+1)} = K_{I(k+1)} \left[\tilde{p}_{L,I(k)} \tilde{p}_{L,k+1} + \tilde{p}_{L,I(k)} \times \bar{p}_{L,k+1} + \bar{p}_{L,I(k)} \tilde{p}_{j,k+1}\right] \tag{36}$$

$$\bar{p}_{L,I(k+1)} = K_{I(k+1)} \left[\bar{p}_{L,I(k)} \bar{p}_{j,k+1}\right], \quad k = 1, \ldots, N \tag{37}$$

$$K_{I(k+1)} = \left[1 - \sum_{j=1}^{N} \sum_{\substack{i=1 \\ i \neq j}}^{N} p_{j,I(k)} p_{i,k+1} \right]^{-1} \tag{38}$$

$$\{L_j\} : \beta_j = \frac{p_{j,I}(N)}{1 - \bar{p}_{L,I}(N)} \tag{39}$$

$$\{L\} : \beta_L = \frac{\tilde{p}_{L,I}(N)}{1 - \tilde{p}_{L,I}(N)}. \tag{40}$$

$$\text{Specificity} = \frac{\text{True Negative}}{\text{False Positive} + \text{True Negative}} \tag{42}$$

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{43}$$

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{Positive} + \text{Negative}} \tag{44}$$

$$F\text{-measure} = \frac{2\,\text{True Positive}}{2\,\text{True Positive} + \text{False Positive} + \text{False Negative}} \tag{45}$$

$$\text{MCC} = \frac{\text{True Positive} \times \text{True Negative} - \text{False Positve} \times \text{False Negative}}{\sqrt{(\text{True Positive} + \text{False Positive})(\text{True Positive} + \text{False Negative})(\text{True Negative} + \text{False Positive})(\text{True Negative} + \text{False Negative})}} \tag{46}$$

Here, β_L represents the unsigned degree of probability to any input sequence L. It has been proved that $\sum_{j=1}^{N} \beta_j + \beta_L = 1$ [3]. The final output sequences are generated by the aggregation of L sequences. The lower bound of the likelihood is β_j that assess the output sequence L_j and the upper bound of likelihood represented by $(\beta_j + \beta_L)$.

The logic behind the ER approach is that all output kth sequences are activated by weight; then, the overall output sequence must be assigned L_j degree of belief. The degree is measured by both the degree by which is assigned by the kth output and degree of belief is assigned by the activation weight of input sequences. The ER approach is used for biological data reduction and generates small size of data volume from large biological datasets. It is based on the Dumpster's theory which is P-complete approach [64–66]. It also solved the data conflict and removed noisy data from large biological data sample.

3.6 Matthews's correlation coefficient (MCC)-based performance evaluation

Modern computational analysis supports set of metrics to assess the mining algorithms. The MCC enables set measurements to assess the performances of the classification techniques. These metrics are measured under confusion matrix with sensitivity, specificity, precision, accuracy, F-measures, negative predictions and false positive rate that expressed as follows:

$$\text{Sensitivity} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{41}$$

These metrics are used to evaluate the performance of the proposed classification techniques.

4 Result and Implementation

Java platform is used for implementing the proposed approach. The NetBeans IDE (integrated development environment) with Java Development Kit 1.7 (JDK 1.7) has supported for the overall design and assessment. Two different systems have been used to validate the result. First one is core i3 with memory 2 GB, and second one is core i5 with memory 3 GB. The designed system occupied very less space and platform independent as illustrated in Fig. 5. It is possible to design the whole system in the .net framework. However, the Java is more available.

The implemented system user interface as illustrated in Fig. 5 shows that every button on this tool enables the user to perform specific operation. The first button is associated for the DNA clustering process. After selecting this button, four options will show as principal component analysis (PCA), support vector machine (SVM), neural skyline filter (NSF) and finally Fisher's discriminate analysis (FDA).

4.1 The PCA versus the SVM

In the current work, the PCA has two different options, namely intra-class and inter-class variations; its performance degrades when it checks datasets on global domain. On the other hand, SVM determines the whole dataset into a common platform that is called maximum margin hyperlane (MMH). Repeatedly this margin has been

Fig. 5 System implementation

reduced to get the better area of the datasets. This is a mechanical adjustment that deals the large data size for code finding from DNA sequences. The first column in Table 1 represents the testing DNA sequences that are started from 1-million base pair. This table holds 18-million base pair. Initially, the PCA is used with these datasets for dimension reductions and codes finding over cancer dataset. The PCA continuously measures the eigenvalues of the collected data. Infected sequences generate less number of codes. The related eigenvalues with the associated vectors denote all noncoding and coding parts of the infected sequences. The same data size is used by the SVM. Table 1 includes the comparison between the PCA and SVM in terms of the numbers of codes. Moreover, there are more codes than that of counter parts that exist in the table. The dataset length beyond the lengths used here is also checked and noticed that the differences between methods are equally changing over the lengths increases.

Table 1 portrays that the coding regions determined by the SVM are increased with the increase in the DNA sequence. For 20-million DNA base pair, the PCA codes are 246, whereas it is 267 for the SVM counts. For

Table 1 Numbers of codes between PCA and SVM

Data size (bp)	PCA	SVM
10,000,000	123	137
20,000,000	246	267
30000,000	376	412
40,000,000	512	601
50,000,000	678	765
60,000,000	821	901
70,000,000	981	1123
800,000,000	1132	1421
90,000,000	1324	1678
100,000,000	1523	1874
1,100,000,000	1743	2098
120,000,000	2013	2345
130,000,000	2212	2674
140,000,000	2451	2905
150,000,000	2678	3345
160,000,000	2901	3601
170,000,000	3263	4012
180,000,000	3576	4521

1-million data, there is about (267 − 246/267 = 7.86 %) increase in SVM coding area selections. For 30-million data, the counts of the PCA and SVM are 376, and 412, respectively. Thus, there are about (412 − 376/412 = 8.73 %) changes for SVM. It was about 1 % changes from the previous dataset findings. Consecutively, there are 2.5, 3.13, 4.54 and 6 % for 40, 50, 60 and 70-million DNA base pair for SVM over PCA, respectively. For each and every finding, ten to twenty times repetitions are performed during this experimental outcome. As the DNA sequences increases, the differences are also increased. From 100- to 200-million DNA base pair, the differences are larger than that of previous half part of the DNA base pair. The differences for second parts are 9 % for 100-million data, 12 % for 120-million data, 14 % for 150-million dataset, and 17 % for 170-million DNA base pair and finally 20 % for 200-million DNA nucleotides. The relationships and changes are demonstrated in Fig. 6, where the green stair is the coding areas from SVM and the red stair represents the outcomes from PCA. The green part is little higher than the red. This 3-D demonstrates the findings between the PCA and SVM. The *X*-axis denotes the data sizes that are stated from 1-million DNA bases to 18-million DNA bases. The *Y*-axis depicts the findings of both methods. Initial point is 0 where last point is 5000. The last highest value of the SVM is 4521; therefore, the maximum *y*-axis limit is 5000. *Z*-axis portrayed the graphical differences between these two processes.

Figure 6 depicts that from the starting point till the end, the SVM outcomes are higher than PCA. At the beginning, the differences are slight and invisible as compared to individual stair. From second stair to last one, there are clear graphical differences between the coding regions finding between the PCA and SVM. It is easy to conclude that for large DNA base pair, the SVM has better capabilities than PCA for DNA coding area findings. Same impact is noticed for data size beyond the table. For any size of the dataset, same differences are verified. Due to the space limitations, only small size is depicted.

4.1.1 The PCA and SVM confusion matrix measurements

Table 1 depicts that PCA finds 123 coding areas from 10-million infected DNA base pairs, whereas the SVM detects 137 coding areas. Thus, the SVM outperforms the PCA due to repeated adjustment of the MMH. Moreover, the confusion matrix is included to measure the accuracy and error rate of both processes. The confusion matrix analysis for these codes using the PCA is illustrated in Table 2.

Table 2 depicts that the overall evaluation for PCA with the 10-million DNA base pair (last column in Table 1) has error rate and accuracy rate of values 5 and 95 %,

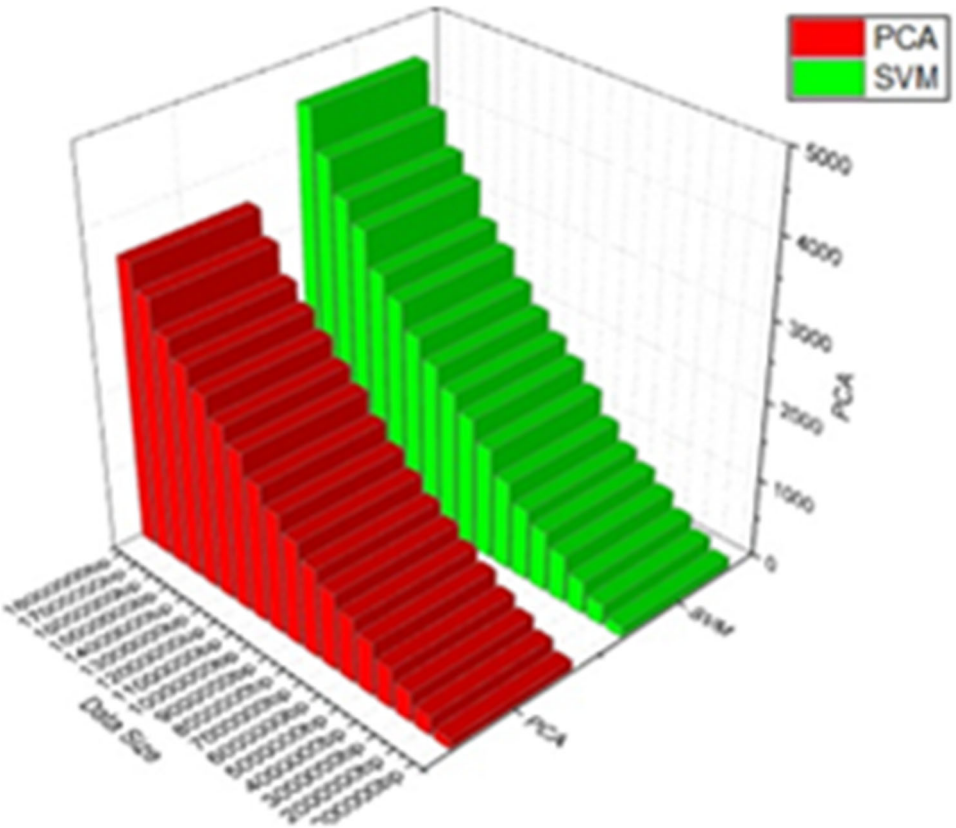

Fig. 6 Coding regions counts by PCA and SVM

Table 2 PCA confusion matrix

	Predicted code (total codes = 140)	Predicted not 140
Actual codes (total codes = 140)	**123**	17
Actual not codes	483	9764

Table 3 SVM confusion matrix

	Predicted code (total codes = 140)	Predicted not code
Actual code (total codes = 140)	**137**	3
Actual not code	321	9926

respectively. The confusion matrix analysis for these codes using the SVM is illustrated in Table 3.

Table 3 depicts that for the SVM with 10-million DNA base pair, the overall outcomes for the SVM achieve error rate and accuracy rate of 3 and 97 %, respectively. The confusion matrix of SVM reflects better impact than that of PCA, where the accuracy rate has increased. The accuracy rates for the 18-million dataset are 94 % for the PCA and 95.7 % for the SVM. Thus, as data size gradually increased, the accuracy rate of SVM also increased in all level of data volume. However, the accuracy rate does not always show the actual predictions in machine learning analysis. Therefore, the sensitivity, specificity and *F*-measures assure perfect predictions or identification in DNA coding part experimental analysis.

4.1.2 The PCA and SVM sensitivity measurements

In the current work, the sensitivity defines how well the PCA finds the coding area. The complement of the coding area as a false negative measurement is the noncoding part of the DNA, where Sensitivity + False Negative rate = 1. By applying Eq. (41), the PCA sensitivity is approximately 20 % with accurate prediction value of approximately 88 %.

Consequently, it is noticed that the accuracy measurement was 95 %, while the Sensitivity = 1 − False Negative rate in the confusion matrix measures 88 %. So, there is some limitation in the accuracy computing. This difference between sensitivity and accuracy for the PCA indicates the significance of sensitivity measurements for machine learning assessments. It is obvious that PCA is also very impactful techniques to address the problems in automated and dynamic environments.

Meanwhile, the SVM generates better sensitivity for coding area selections and investigation. The SVM sensitivity tells about the prediction rate of the collected datasets by imposing the idea of maximum margin hyperlane (MMH) of SVM.

The experimental results illustrate that by a confusion matrix for 10-million DNA base pair, the SVM identifies 137 coding regions from the DNA long sequences, while it was on 123 coding regions for the PCA. Measurement of the SVM sensitivity by taking value from the confusion matrix (Table 3) is 30 % approximately. For the PCA, from the confusion matrix (Table 2), the sensitivity is 20 %. Thus, the overall improvement in SVM is 33 %. This is a significant progress of SVM due to its repeatedly adjustment of maximum margin of datasets variations.

Moreover, the absolute predicted coding regions of the SVM are 95.5 %. However, the accuracy measurement of SVM was 97 % at initial assessment. So, the sensitivity measurements reduced 1.5 % negative predictions than the accuracy. On the other hand, the sensitivity accuracy of PCA is 88 %. So there is about 7 % better renovation of SVM than that of PCA. Thus, the SVM outperforms PCA for DNA coding selection from large DAN datasets. As data length increases, these differences will be also increase. Above measurement and comparisons are done for 10-million DNA base pair.

4.1.3 The PCA and SVM specificity measurements

In the experimental results, there are two different outcomes such as target values and nontarget values. Targeted points are the key values. However, sometimes nontargeted values are very significant for its existence in total dataset. In current machine learning techniques, specificity deals with this phenomenon. Consecutively, the specificity refers to the correctness of identifying the noncoding areas accurately from collected dataset. Rather finding exact outcomes, specificity measures the negative part of an experiment. Though, specificity deals with the less important features of the collected dataset. It has equal impact as pivotal features finding. In the case of very large data size, this process predicts wrong values with slow processing. In a sense, it can be defined as summation of corrected area and false positive area. Mathematically, the specificity is calculated using Table 4 and Eq. (47).

$$\text{Specificity} = 1 - \text{False Positive DNA coding area} \quad (47)$$

From the specificity measurements, the experimental result can be easily concluded that PCA accurately detected the noncoding parts of the total datasets with 99 % specificity, while 99.98 % specificity is calculated for the SVM.

The outcomes of specificity provide the confident that the assessment and analysis done over the datasets are valid.

Table 4 Specificity of PCA

	Prediction		
	Coding area	Noncoding area	
PCA			
Coding area	123	17	140
Noncoding area	483	9764	10,247
	606	9781	10,387

4.1.4 The F-measures for PCA and SVM

The *F*-measures indicates the exact outcomes of the PCA or the SVM using a ratio between precision and recall measure. For the PCA, the precision, recall and the *F*-measure are 88, 20 and 4.33 %, respectively. The value of 4.33 %indicates that the PCA predictions are almost accurate in all respects of accuracy, sensitivity and specificity. Figure 7 illustrates the graphical relationship between the precision and recall. In Fig. 7, the range is normalized for better results representation. These small changes help to fit the training data point perfectly over any line graph. It establishes the impact of PCA for codes finding from DNA sequences.

In Fig. 7, the recall line indicates that it has less value that going down from a certain value to the last value that is 1. Similarly, the precision value is also decreased due to the wrong predictions. The range of the both values can be easily extendable to any range. Meanwhile, for the SVM, the precision, recall and the *F*-measure are 95.5, 30 and 3.18 %, respectively. Figure 8 (blue line) illustrates the relationship between precision and recall for the SVM.

Figure 8 demonstrates the comparative representation for the recall and precision for both the PCA and SVM. The *F*-measure value of the PCA is 4.33, whereas it has

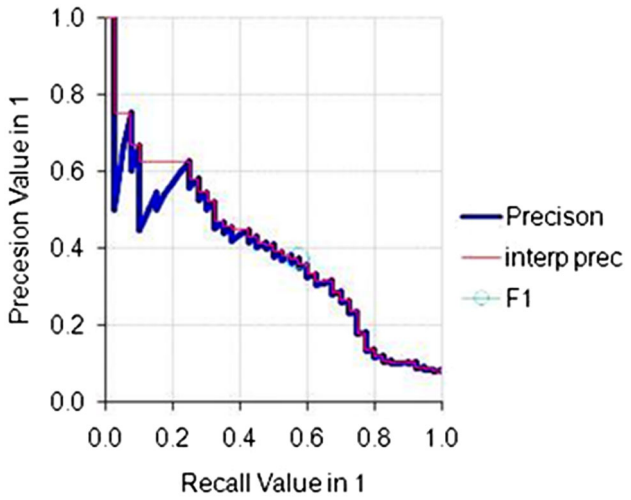

Fig. 7 Relationship between precision and recall for PCA

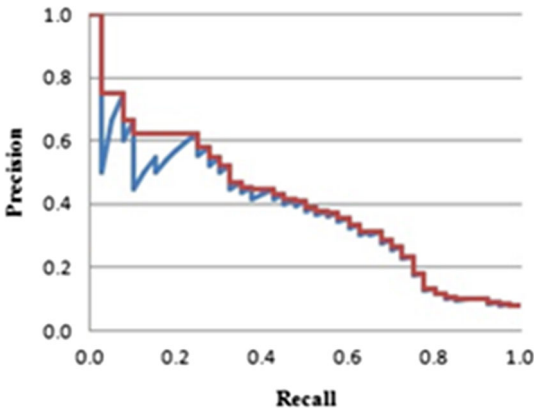

Fig. 8 Relationship between precision and recall for SVM and PCA

value of 3.18 for the SVM. Reduced *F*-measures proved also that the SVM achieves better outcomes compared to the PCA. In the case of comparison for 10-million datasets, the SVM has perfect impact of 36 % than the PCA. For 20-million dataset, the PCA achieves *F*-measures of 4.78 % and SVM of 3.21 %, which indicates 48 % superiority for the SVM. For 1-million DNA base pair, there are about 12 % differences between these two machine learning processes. For 90-million dataset (middle of the dataset), PCA assesses the *F*-measure is 5.23, whereas the SVM achieves 3.32 % with difference of 57 %. So, at the half of dataset size, the SVM is more perfect compared to the PCA. For 100- and 80-million DNA sequences, the 20-million datasets, the differences are about 72 % and it will be increased more when the DNA sequences are going large scale.

In Fig. 8, the blue and pink lines indicate the SVM and PCA corresponding *F*-measures, respectively. Consequently, for the obtained results the SVM accuracy shows better impact compared to the PCA. Additionally, sensitivity, specificity and *F*-measure also show the superiority of the SVM over the PCA.

4.2 The FDA versus the SVM and PCA

Previous experiments proved the superiority of the SVM to the PCA. Therefore, for same data length in Table 1, new experiments have been analyzed to compare between the FDA and SVM. The system that is developed to provide an option to solve the dataset with the desired techniques is illustrated in Fig. 9.

By selecting the first button of Fig. 1, the user will find a new screen as shown in Fig. 9. The browse button enables the user to choose the file from database or any drive or internet. Then, the FDA or SVM buttons guide the researchers to perform their experiments to find the specific features from training dataset. In the current work, the numbers of coding regions are the key outcomes from FDA

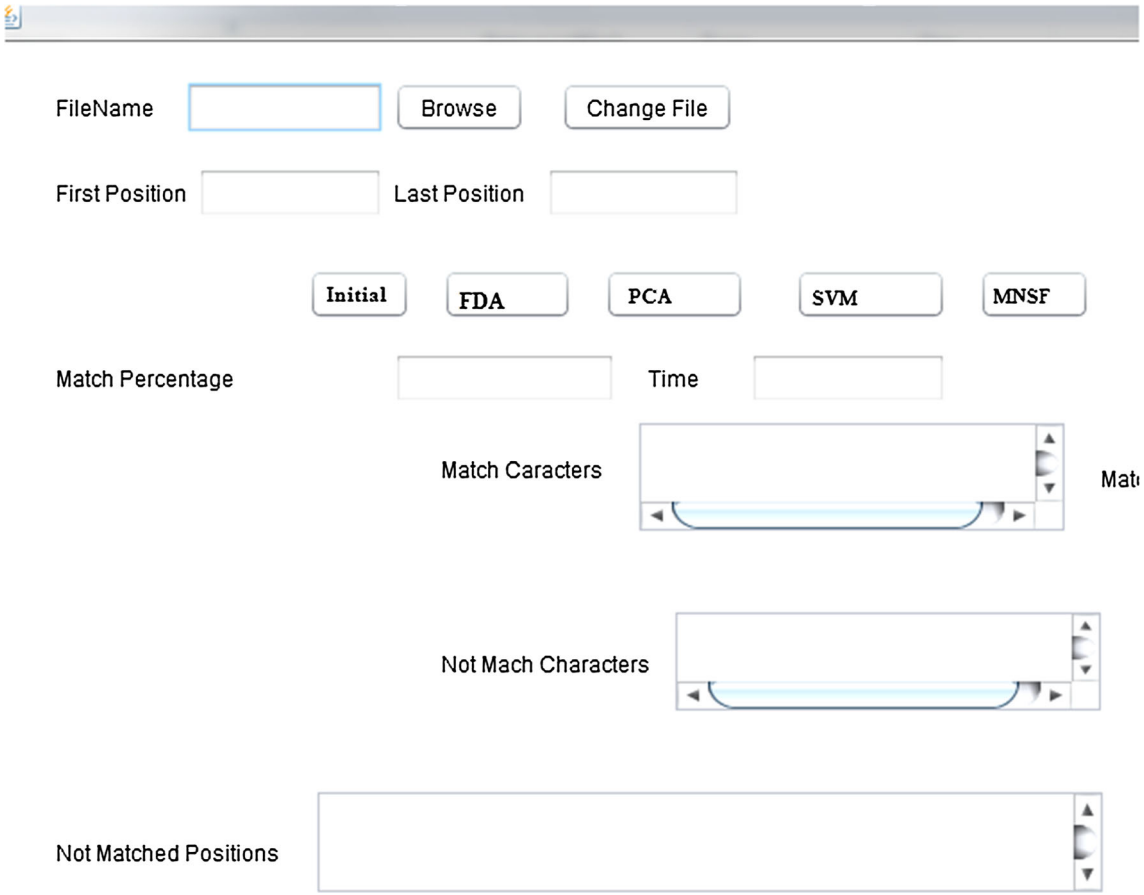

Fig. 9 Options for FDA and SVM comparisons

or SVM. The comparison of the outcomes of the FDA and SVM as well as the PCA is depicted in Table 5.

Table 5 establishes that FDA provides superior measurements compared to the SVM and the PCA. For some initial dataset, the differences between the FDA and SVM are very small. For 10–15-million dataset, there is no difference between the FDA and SVM. Ten times are verified for the same data, and there are no differences for the codes finding, where both the FDA and SVM generate the same amount of codes. However, when the DNA base pair reaches 6,000,000 bp, then the SVM counts 901 codes, and for same data the FDA measures 1020 codes. Table 5 depicts that for 60-million large DNA base, the SVM finds 901 codes, while the FDA finds 1020 codes. However, the PCA finds 821 codes. So, it counts (1020 − 901/1020 = 11.66 %) greater than the SVM and (1020 − 821/1020 = 19.50 %) greater than PCA. For the next subsequent dataset, FDA counts 1453 codes from the whole DNA, whereas SVM counts 1123 codes with difference of 22.71 %. Meanwhile, with the increase in the data size, the difference between the previous counts and current counts becomes almost double. For 100-million DNA base pair, the difference in codes finding when using FDA and SVM has a value of 24 %. For last DNA base pair (hundred eighty millions), the difference between these two methods is 1594 codes. The FDA can find almost 1600 more codes than that of SVM. These counts are almost 26 % more than SVM and 42 % more than PCA. Due to long data transformation capability and distributions, the FDA can compute the whole DNA sequences together. The FDA manages all the data lengths either inter- or intra-class variations. These variations allow counting the whole data in a certain length. PCA and SVM are not able to do the same, and it missed some parts of the training data. On the contrary, the SVM divides the whole sequences into two separate regions. Therefore, the SVM might lose some DNA segments uncovered due to lack of adjustment of two separated regions into one. However, FDA combines all dataset into a common compact distribution that is harmonic distribution. These are the key facts FDA outperforms the PCA and SVM. Later section will validate the FDA findings. As a result, FDA covers the whole DNA data and finds more codes from given training dataset.

Table 5 Coding areas findings among PCA, SVM and FDA

Data size (bp)	PCA	SVM	FDA
10,000,000	123	*137*	*137*
20,000,000	246	*267*	*267*
30,000,000	376	*412*	*412*
40,000,000	512	*601*	*601*
50,000,000	678	*765*	*765*
60,000,000	821	901	**1020**
70,000,000	981	1123	**1453**
80,000,000	1132	1421	**1765**
90,000,000	1324	1678	**2098**
100,000,000	1523	1874	**2452**
110,000,000	1743	2098	**2786**
120,000,000	2013	2345	**3198**
130,000,000	2212	2674	**3578**
140,000,000	2451	2905	**3999**
150,000,000	2678	3345	**4564**
160,000,000	2901	3601	**5012**
170,000,000	3263	4012	**5563**
180,000,000	3576	4521	**6109**

From Table 5, the 3-D relationship among PCA, SVM and FDA demonstrates that FDA finds better prediction than other two processes as illustrated in Fig. 10.

Figure 10 establishes that the FDA finds maximum number of codes from the training dataset. In the current work, cancer dataset is used to detect the codes that are responsible for cancer diseases. It is clear form Fig. 10 that the FDA is higher than other two stairs. For the last DNA base pair, the differences show the maximum differences among these three classification methods. Since, FDA holds dynamic features to maintain automatically large DNA data, it can compute better outcomes. The FDA uses two different mathematical functions during the assessment of the whole DNA sequences. There are inter-class and intra-class variability, where with small data lengths, intra-class variability is used, while with very large data sizes, big dataset inter-class variability.

4.2.1 *Confusion matrix of the FDA compared to the PCA and SVM*

The FDA shows the same accuracy for 10–50-million DNA base pair as the SVM. From 60-million DNA base pair for FDA, the following metrics are studied. The first column in the confusion matrix of the FDA indicates the exact findings and wrong predictions. The second column denotes the number of codes not measured. The first row indicates the actual determined codes from the existing total codes in the given training DNA base pair. The second row portrays the complementary of the codes, the noncoding area findings. Table 6 illustrates the FDA confusion matrix.

For very large dataset, it is difficult to compute the coding areas compared to small DNA base pair. From Table 6, the computed error rate for the FDA is 3 %, while for same data length the error rate for the SVM is 8.7 %. There are about 5 % benefits of the FDA over SVM. Moreover, the obtained FDA accuracy is 97 %, while from the previous experiments the SVM gained 97 % accuracy SVM for 10-million DNA base pair. Thus, the FDA predicts the same impact for 60-million DNA base pair. Moreover, FDA accuracy is about 22 % more than the PCA for similar DNA base pair. So, the FDA can easily handle big DNA dataset than that of PCA and SVM. Moreover, FDA can perform its analysis for supervised and unsupervised learning. In this research, when the DNA base pair is small, it can be considered as supervised learning, while in the case of large datasets, it is easy to consider as unsupervised learning. Nowadays, unsupervised learning is significant for large data handlings irrespective of bioinformatics, data mining, image processing or signal processing.

4.2.2 *Sensitivity of the FDA compared to the PCA and SVM*

The sensitivity of FDA reflects the applied dynamic environment during the experiments. The FDA covers whole dataset into a certain area where repeated counting is done. These iterative analyses identify all the coding regions exist in certain DNA sequences. For large DNA sequences of 60-million base pair, the confusion matrix has portrayed the outcomes of the FDA findings. The sensitivity confusion matrix includes rows reflecting the FDA and columns indicating the resultant values of FDA as demonstrated in Table 7. The first column and first row relate the actual findings. The second row and first column associate the misjudgment of coding area selections. Similarly, the first row and second column make the relationship that failed to find the coding area from total number of coding area. The second row and second column show the not code values, which indicated exactly not coding area. Thus, the sensitivity of FDA $= 1 -$ wrong perditions.

Here, for 60-million DNA base pair, resultant narrated into the matrix. This FDA-automated computing finds 1020 codes out of 1022 coding areas for 60,000,000 DNA base pair. SVM outcomes are 901 and PCA finding are 821.

Table 7 depicts that the sensitivity of FDA is 9 % for 60-million large dataset, while 7 % is achieved for the SVM over same data. So, there are about 2 % better measurements on the FDA over SVM. Moreover, the PCA sensitivity for same data is about 5 %. For 60-million DNA

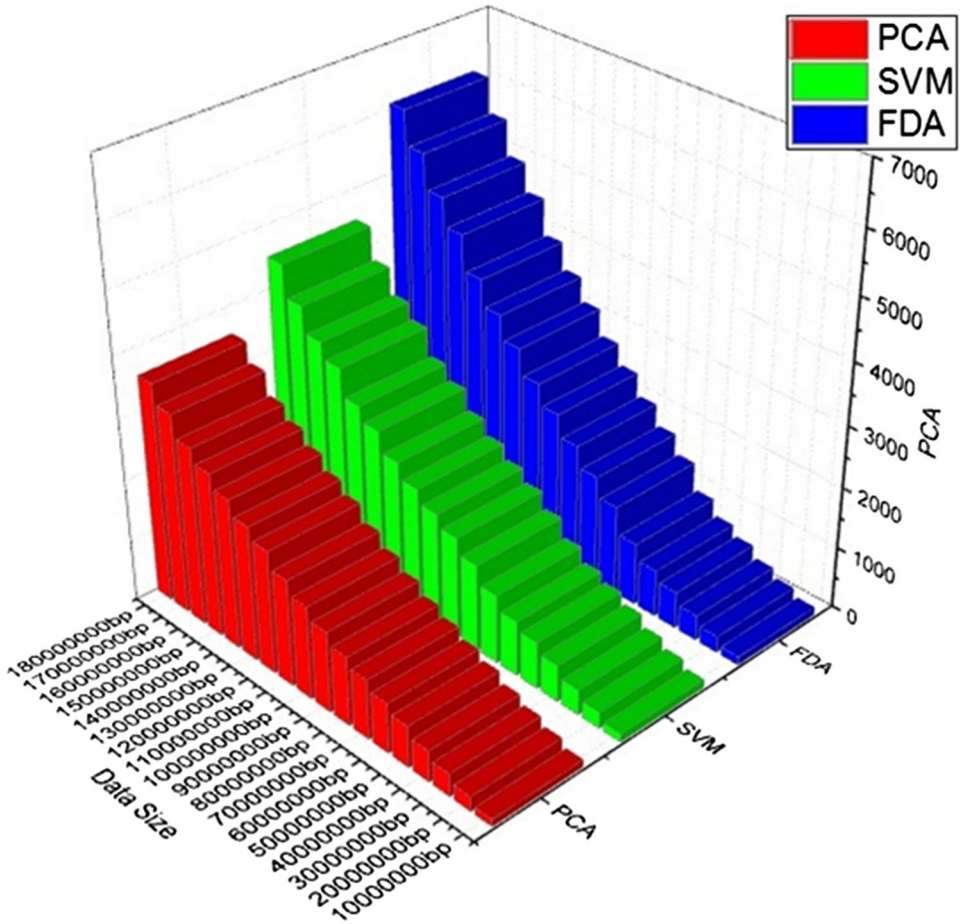

Fig. 10 Codes finding relationships among PCA, SVM and FDA

Table 6 FDA confusion matrix

	Predicted codes (total codes = 1022)	Predicted not codes
Actual codes (total codes = 1022)	1020	2
Actual not codes	11,134	296,782

Table 7 FDA sensitivity confusion matrix

	Prediction		
	Coding area	Noncoding area	
Coding area	1020	2	1022
Noncoding area	11,134	296,782	307,916
	12,154	296,784	308,938

base pair, the FDA has 4 % better impact than PCA. As data size increases, the FDA measures achieve greater values compared to both the SVM and PCA. Subsequently, for 70-million DNA base pair, the FDA sensitivity is 11 %, while the SVM and the PCA count 8 and 5 %, respectively. In addition, the increase of 1-million DNA sequences increases the sensitivity of the FDA by 1 %. For 80-million DNA sequences, and the FDA achieves 14 % increase, whereas the SVM and PCA achieve 9 and 7 %, respectively. Thus, for these dataset, the FDA outperforms PCA by 50 % and gains 5 % more than SVM. These increases are continued up to last dataset. The FDA measurements are more perfectly compared to the SVM and PCA. However, some other metrics such as the specificity and *F*-measures are also vital to prove the FDA superiority.

4.2.3 Specificity of the FDA compared to the PCA and SVM

In this work, for measuring noncoding regions with large DNA data, it consumes 120 s for 500-million DNA bases. So, the specificity = 1 − coding areas from whole training DNA bases has a value of 3.71 % for the FDA.

A small value of specificity defines the true impacts of large data handling problems in bioinformatics assessments. This value defines the validity of specific DNA sequences for FDA.

4.2.4 *F-measures of the FDA Compared to the PCA and SVM*

Since, the measured precision and recall for the FDA are 99 and 8 %, respectively. Thus, the *F*-measure of FDA is 12.4 %. For same data length, the *F*-measure for SVM and the PCA are 16.45 and 20.87 %, respectively. For the 60-million datasets, the FDA has perfect predictions compared to the SVM that achieves 32 % *F*-measure. Afterward, for 70-million DNA base pair, the differences become almost 34 %. Additionally, the differences between PCA and FDA are about 47 and 52 % for 60 and 70-million DNA base pairs, respectively. Eighty-million DNA dataset generates more differences of 42 % than SVM and 58 % than PCA. Due to the large data processing capabilities, FDA always outperforms PCA and SVM for any amounts of DNA data. Figure 11 illustrates the comparative plots of the *F*-measure performance for both the FDA and SVM.

Figure 11 establishes that gradually as data size reduces, the *F*-measure values for both methods reduce to certain points. However, irrespective of DNA data size, the FDA always computes better outcomes. For SVM and PCA, ranges of recall and precisions are 0–1; however, for the FDA random ranges have been used to make dynamic prediction environments.

4.3 The MNSF versus the FDA

Mapping-based neural skyline filtering enables faster and efficient processing than PCA, SVM and FDA. Mapping is a function that computes large dataset in a parallel environment. This function repeatedly measures the whole dataset at a time. In a second, 32 mapping functions are used to find the coding area. Other methods serially compute the DNA sequences, whereas the MNSF imposes 32 functions in a row. This facility makes MNSF distinct from PCA, FDA and SVM. The mapping quarry that MNSF applied in the current work is an integrated environment, which are implemented as follows.

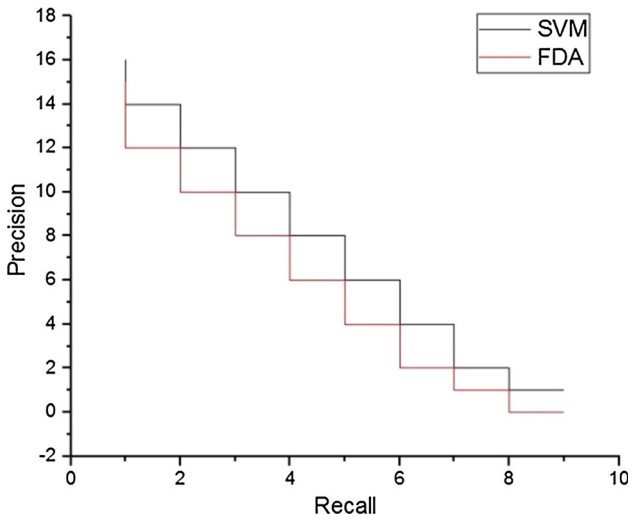

Fig. 11 *F*-measures between FDA and SVM

Mapping Quarry Using The MNSF

```
Generate Outer DNA Data Table for Codes
With
    (
    Type=Combined_MAP_Function,
    Location=localhost.DNA database
    Database=Mapping DNA
    Documentation= Sarwar Kamal,
    Combined_MAPPING=Codes Map
    )
```

This quarry followed by some other significant quarries that are essential for codes finding. There are huge numbers of DNA segments for over all experiments. However, faster analysis requires more DNA sequencing. In this work, there are 32 parallel quarries that are used to perform all the included datasets. The parallel environment is the key improvement for MNSF codes findings. The quarry for the implementation of the whole DNA segments with small dataset is given as follows.

Implementation Quarry of the whole DNA Segmen ts with Small Dataset

```
Produce outer Tables External Table
 [DNA].[All codes excess ]
     (
     [DNA Codes ID] int,
     [Non codes ID] int,
     [Time] of Codes findings,
     [Type] varchar(30000000),
     [Instruction] varchar(234432),
     [SQL Management] varbinary(453),
     [Beginning offset] int,
     [Last offset] int,
     [System Time] int,
     )
Under
     (
     DNA Segment source = DNA Table,
     Schema = DNA_System
     Object = Codes,
     Environment=Iterative
     )
```

The union quarry is then applied to bind all table codes in a certain area. The DNA table is the physical storage area to store the training and testing DNA dataset. There are *n* numbers of tables to store the whole DNA sequences.

The iterative environment assures the continuous support to codes selections from table.

Selection Procedure

```
SELECT DNA_Codes  FROM Segment 1
UNION ALL
SELECT DNA_Double Codes FROM Segment 2
```

The specific quarry that is critical for sequencing of large DNA segment finding is given by:

Quarry for Sequencing of Large DNA Segment Finding

```
Select Code Symbol, having first_A, last_T
From DNA Bank
   Accurate Match_Identify (
  Separated byC,G, Order By A,C,T,G
Compute First (A.DNA Bank) as first_A,
        Last(T.DNA Bank) as last_T
Whole rows matches
 Segments ((A+ T+) (G+ C+) A)
 IntroduceA as (Initial),
G as (Middle nucleotide),
      T as end (G.DNA Bank) – Initial (A.DNA Bank <= 3
)
Where symbols (AGCT)
```

For all data lengths, the remaining processes compute less number of DNA codes. There are significant improvements in the codes findings due to the mapping facilities of the MNSF. There are twenty times iterations to check the outcomes among these four methods. Symbol arrangement of the quarry might be changed at any time. Last line of the last quarry plays a vital role for checking complete sequences of the training DNA dataset. The difference between *G* and *A* makes as less than or equals 3 due to the common length which is always 3 for all DNA codes sequences. Segments that enlarge with plus sign indicate the more same nucleotides belong to the DNA segments. The whole dataset is stored into database named as DNA bank. Dot operator enables to access the desired nucleotide from the DNA bank to move forward to check whole DNA sequences. This process supports the finding of frequent codes from uninterrupted DNA sequences. Traditional database is used here for whole sequencing. However, Hadoop and Spark are checked for some part of the sequencing and the outcomes are remarkable than traditional database. The positive part of this research is that it is adjustable to both traditional and big database. Traditional database indicates MySQL, Microsoft access and so on. On the other hand, Hadoop, Spark and Shark are the big data environments. Among these three, Spark is the best big data handling environments for bioinformatics data and algorithms. Hadoop focused mainly on Web links and resource description framework (RDF). However, DNA, RNA (ribonucleic acid) and proteins are the large biological networks that deal lots of interaction that are very complex as well as large in size. Consequently, huge numbers of interactions are created as well as distributed environments. Spark enables to process such big interactions and distributed hub.

The outcomes of MNSF compared to the FDA, SVM and PCA are illustrated in Table 8.

Table 8 illustrates that in all aspects of sequencing, there are large differences between the MNSF compared to the FDA, SVM and PCA. For 10-million DNA dataset, the MNSF finds 140 codes out of 140, while both the FDA and SVM find only 137 each and the PCA finds 123 codes. There are about 5 % prediction improvements for mapping achieved with the MNSF. For 20-million DNA base pair, the MNSF measures 9 % better than the SVM and 11 % better than PCA. For next data lengths of Table 8, the difference is continued the same performance for the MNSF. Moreover, for 50-million DNA base pair, there are about 15 % higher codes for the MNSF than SVM and 22 % differences than PCA. These distinctions are continued till the last DNA base pair is reached. The large differences are noticed for the last DNA base pair in Table 8. There are about 55 % differences between the PCA and MNSF and 40 % differences between the SVM and MNSF and 25 % differences between the MNSF and FDA. The significance of the current work is that FDA and MNSF are superior to both the PCA and SVM. However,

Table 8 Outcomes of four methods used here as PCA, SVM, FDA and MNSF

Data size (bp)	PCA	SVM	FDA	MNSF
10,000,000	123	137	137	140
20,000,000	246	267	267	300
30,000,000	376	412	412	480
40,000,000	512	601	601	700
50,000,000	678	765	765	880
60,000,000	821	901	1020	1209
70,000,000	981	1123	1453	1786
80,000,000	1132	1421	1765	2023
90,000,000	1324	1678	2098	2321
100000000	1523	1874	2452	2653
110,000,000	1743	2098	2786	2908
120,000,000	2013	2345	3198	3442
130,000,000	2212	2674	3578	3896
140,000,000	2451	2905	3999	4432
150,000,000	2678	3345	4564	5098
160,000,000	2901	3601	5012	5754
170,000,000	3263	4012	5563	6432
180,000,000	3576	4521	6109	7432

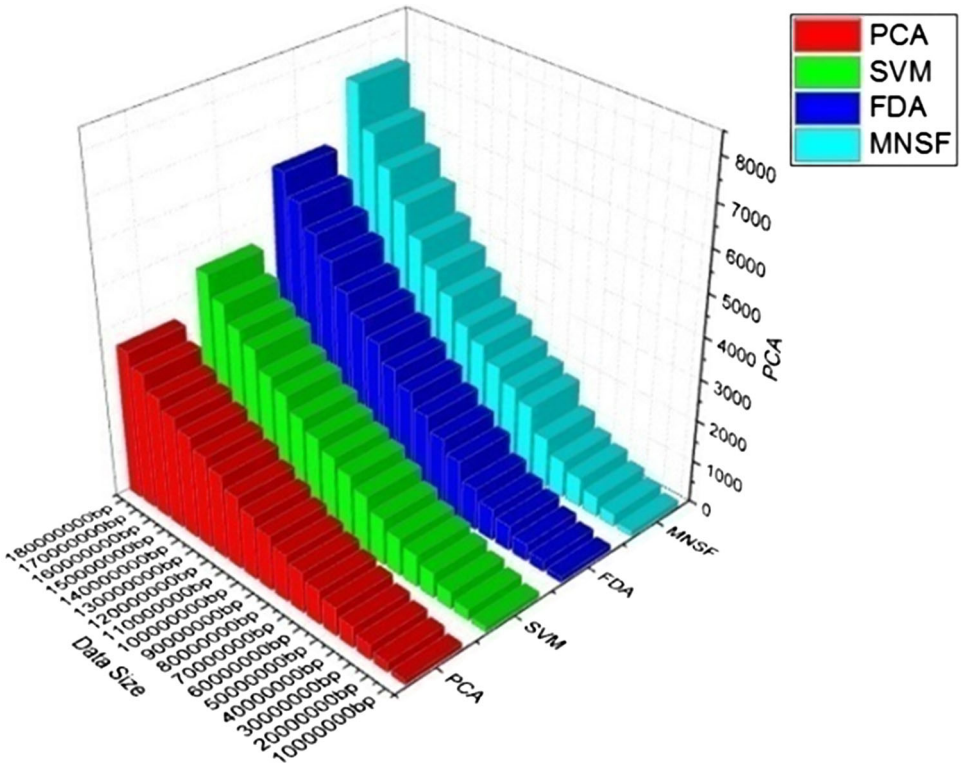

Fig. 12 Comparative demonstrations among PCA, SVM, FDA and MNSF

the MNSF is the best prediction method due to its mapping facilities as demonstrated in Fig. 12. The 3-D view enables clear sketching of the outcomes.

Figure 12 establishes that the MNSF representing the codes finding outcomes of mapping neural skyline filtering outperforms the other techniques. Due to the gradual improvements in the MNSF than all other methods over the all DNA dataset, thus, the specificity, accuracy, sensitivity and *F*-measures are not required for the MNSF.

4.4 The time differences comparison of the four techniques

Since the MNSF maintains mapping facilities, it takes very less time compared to the other three methods as illustrated in Table 9.

Table 9 shows that the MNSF consumed less time compared to the other three methods. For 10-million DNA base pair, the MNSF consumes 498 nanoseconds (ns), whereas the SVM and FDA consume 981 and 988 ns, respectively. Conversely, the PCA consumes only 723 ns that are smaller than that of the SVM and FDA. Thus, the MNSF is (723 − 498 = 225/723 = 32 %) faster than PCA, 49 % faster than SVM and 49.5 % faster than FDA. So, for small length DNA sequences, the MNSF process is quicker than other methods only for its mapping capabilities. The time differences are gradually increased for all remaining data lengths. For 50-million DNA nucleotides, the differences with PCA are about 34 %, with SVM it is about 51 %, and with FDA the differences are about 55 %. The FDA takes more time than all techniques used due to its validity checking criteria. For 100-million DNA sequences,

Table 9 Time consumed for codes finding by PCA, SVM, FDA and MNSF

Data size (bp)	PCA	SVM	FDA	MNSF
10,000,000	**723**	981	988	498
20,000,000	1054	1563	1590	**900**
30,000,000	1686	2452	2489	**1280**
40,000,000	2154	2954	2987	**1564**
50,000,000	2567	3532	3590	**2078**
60000000	3245	4100	4109	**2456**
70,000,000	3678	4658	4695	**2767**
80,000,000	4211	5400	5432	**3098**
90,000,000	4654	6123	6198	**3200**
100,000,000	5432	7600	7654	**3456**
110,000,000	5908	8176	8286	**3665**
120,000,000	6892	9123	9234	**4042**
130,000,000	7643	10,123	11,002	**4496**
140,000,000	8123	11,232	12,121	**4709**
150,000,000	8654	12,098	13,121	**5143**
160,000,000	9012	13,095	14,098	**5422**
170,000,000	9456	14,777	15,098	**5809**
180,000,000	9765	15,643	16,543	**6241**

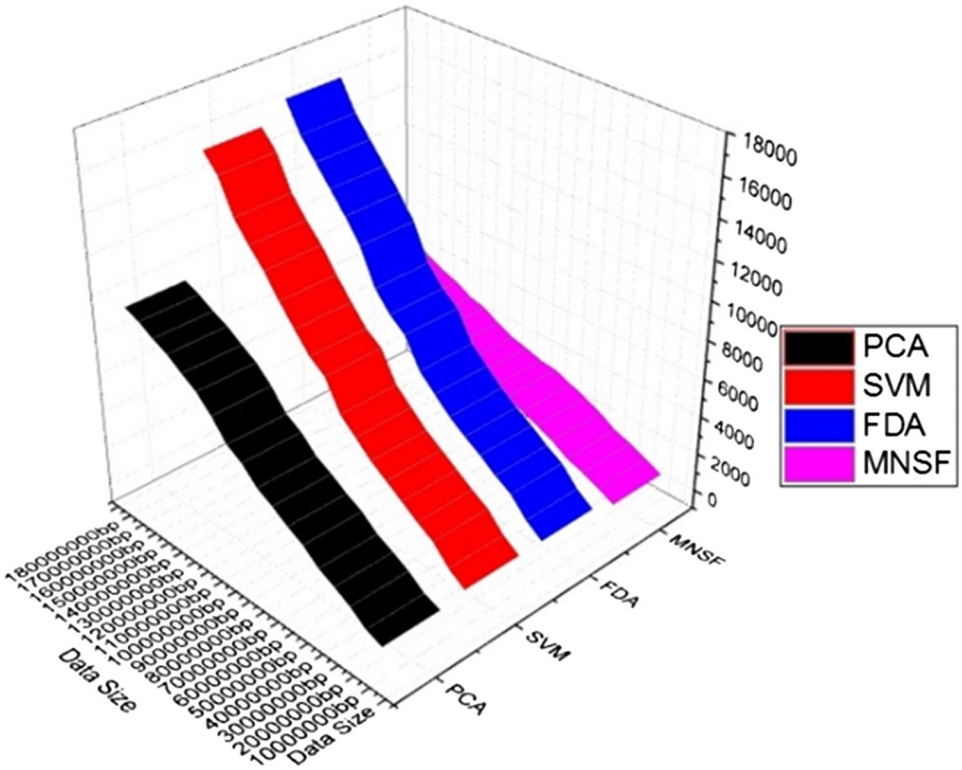

Fig. 13 Timing comparisons among four methods principal component analysis, support vector machine, Fisher's discriminant analysis and mapping-based neural skyline filter

these difference are 40, 54 and 58 %, respectively, for PCA, SVM and FDA. For 18-million DNA base pairs, these timing differences reach at 45 % for PCA, 58 % for SVM and 63 % for FDA.

Consequently, the PCA takes less time than SVM to check the codes; however, its predictions are poorer than the SVM. The PCA investigates only some part of the sequences that are matched with their conditions, while the SVM and FDA confirm their checking with confidence. Moreover, PCA takes less memory space than SVM and FDA. For small DNA sequences, PCA performs better than FDA and SVM; however, for very big DNA dataset MNSF outperforms other three methods. Figure 13 represents graphically the differences obtained in Table 9. However, in codes predictions, FDA and SVM are clearly ahead than PCA. This is the prime concentration of the current work.

Figure 13 establishes that the SVM and FDA consume double time than the MNSF, while the PCA consumes average time among all four methods.

4.5 The ER-MNSF versus MNSF

Reduction is a new phenomenon to arrange big dataset into small area. Recently, reduction processes are frequently applied in each and every parallel data processing. Reduction phase compressed similar data that are not related to the codes for initial consideration. When the iteration receives the same data again, it simply ignores these data and moves forward. This skipping process helps to cover entire DNA sequences within shorter periods of time. Due to repeated patterns in DNA sequences, more than half of the patterns do not matches with the codes and they are considered as valueless dataset. Sometimes, redundant data are also being removed from the training data. Moreover, irregular data such as un- expected characters, symbols and digits are removed from training and testing data. Hence, the consumed time by ER-MNSF is less than MNSF. In the case of increased data lengths, the differences are also getting as shown in Table 10. All the testing time is checked for ten times and then stored to Table 9. For each and every checking, the results are same and ER-MNSF is better than all other methods in all respect.

Table 10 represents that for ten-megabyte DNA sequences, the time for MNSF is 498 ns, while it is only 200 ns for the ER-MNSF. Thus, the ER-MNSF is faster by about 60 % than the MNSF. For the next data size, the differences are 61 %, i.e., evidential reasoning made the processing quicker and efficient. For all other lengths, there are gradual improvements in time for ER-MNSF over MNSF. The line graph in Fig. 14 represents the timing outcomes for evidential reasoning-based analysis and the MNSF.

Figure 14 depicts that the starting point to ending point, the timing differences are increasing gradually between the methods. As time progresses, ER-MNSF performs better and better, whereas the timing is increasing for MNSF over the whole data lengths. Therefore, the evidential reasoning

Table 10 Time differences between evidential reasoning-based MNSF without MNSF

Data size (MB)	MNSF	ER-MNSF
10	498	200
20	900	367
30	1280	584
40	1564	689
50	2078	867
60	2456	908
70	2767	1098
80	3098	1109
90	3200	1256
100	3456	1398
110	3665	1488
120	4042	1587
130	4496	1654
140	4709	1765
150	5143	1823
160	5422	1908
170	5809	2000
180	6241	2097

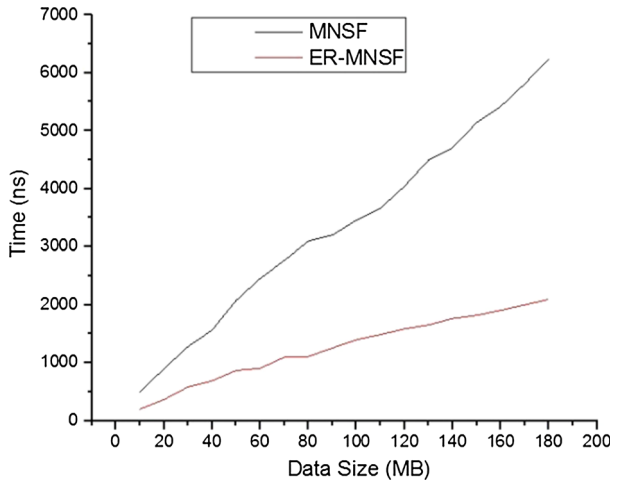

Fig. 14 Timing differences between ER-MNSF and MNSF

(ER)-centric MNSF outperforms all four methods used in the current work. Consequently, the proposed approach proves its efficiency to find the DNA codes. It provides a significant scope in biological data processing laboratories for reducing large datasets, drugs design, agriculture and medicine. However, there are some other factors that are also responsible for cancer, such as proteins–proteins interactions and RNA synthesis. These protein–protein interactions are not measured in this work. The RNA synthesizes can also be considered in the future work. Since proteins interactions are also critical for cancer, this analysis is also recommended in the further future work. Moreover, in the future, proteins interactions and RNA mechanism can be simulated by imposing machine learning and advanced data mining algorithms.

5 Conclusion

Cancer is a precarious disease; thus, cancer data analysis is significant. Small datasets can lead to erratic rate estimation, sensitivity to missing data and other data errors. The current work finds out the key parts from cancer DNA data. These key components are the coding regions of the DNA that control set of factors in the human body. It helps to detect the mutations, damages and changes in long DNA sequences. Genes induct the mistakes occur when cell divide. These mistakes are called the mutations. Coding areas plays vital role to detect and repair these mutations. These mutations are caused by various natural and man-made regions, such as smoke, radiation, chemical mixed environments, ultraviolet radiations from the sun and bad substances in food. Biological processes are very expensive to detect the codes from DNA.

Simulation and computational analysis reduces cost and the required processes by scaling down the very large data. Four different machine learning techniques, namely the PCA, SVM, FDA and the MNSF, are used to measure the codes. Different outcomes are found as codes by applying these four methods to the used cancer dataset. For some special criteria, these techniques differ from each other. Sensitivity, specificity, accuracy and *F*-measures reflect their performances with proper mathematics. Due to large-scale cancer infected DNA data, mapping-based Neural Skyline filter outperforms all other three factors. The key point of this work is that mapping enables the coding area findings to be more accurate and faster. Evidential reasoning is used as a mapping orientation mathematical approach.

Compliance with ethical standards

Conflict of interest The authors declared that no conflict of interest.

References

1. Cui P, Liu H, Aggarwal C, Wang F (2016) Uncovering and predicting human behaviors. IEEE Intell Syst 31(2):77–88
2. Subbian K, Aggarwal CC, Srivastava J (2016) Mining influencers using information flows in social streams. ACM Trans Knowl Discov Data 10(3):26
3. Li J, Le TD, Liu L, Liu J, Jin Z, Sun B, Ma S (2016) From observational studies to causal rule mining. ACM Trans Intell Syst Technol 7(2):14
4. Wu CJ, Ku CF, Ho JM, Chen MS (2016) A novel pipeline approach for efficient big data broadcasting. IEEE Trans Knowl Data Eng 28(1):17–28

5. Leis V, Kemper A, Neumann T (2016) Scaling HTM-supported database transactions to many cores. IEEE Trans Knowl Data Eng 28(2):297–310
6. Bhowmick SS, Seah BS (2016) Clustering and summarizing protein-protein interaction networks: a survey. IEEE Trans Knowl Data Eng 28(3):638–658
7. Zhou C, Cule B, Goethasls B (2016) Pattern based sequence classification. IEEE Trans Knowl Data Eng 28(5):1285–1298
8. Zhong J, Ong YS, Cai W (2016) Self-learning gene expression programming. IEEE Trans Evol Comput 20(1):65–80
9. He J, Lin G (2016) Average convergence rate of evolutionary algorithms. IEEE Trans Evol Comput 20(1):316–321
10. Deadman E, Higham NJ (2016) Testing matrix function algorithms using identities. ACM Trans Math Softw 42(1):4
11. Kiah HM, Puleo GJ, Milenkovic O (2016) Codes for DNA sequence profiles. IEEE Trans Inf Theory 62(6):3125–3146
12. Chien JT, KuBayesian YC (2016) Recurrent neural network for language modeling. IEEE Trans Neural Netw Learn Syst 27(2):361–374
13. Turcu A, Palmieri R, Ravindran B, Hirve S (2016) Automated data partitioning for highly scalable and strongly consistent transactions. IEEE Trans Parallel Distrib Syst 27(1):106–118
14. Deng SP, Zhu L, Huang DS (2016) Predicting hub genes associated with cervical cancer through gene co-expression networks. IEEE/ACM Trans Comput Biol Bioinf 13(1):27–35
15. Hsieh SY, Chou YC (2016) A faster cDNA microarray gene expression data classifier for diagnosing diseases. IEEE/ACM Trans Comput Biol Bioinf 13(1):43–54
16. Dhulekar N, Ray S, Yuan D, Baskaran A, Oztan B, Larsen M, Yene B (2016) Prediction of growth factor-dependent cleft formation during branching morphogenesis using a dynamic graph-based growth model. IEEE/ACM Trans Comput Biol Bioinf 13(2):350–363
17. Borroto OM, Vega JMG, Ponce YM, Grau R (2016) Relational agreement measures for similarity searching of cheminformatic data sets. IEEE/ACM Trans Comput Biol Bioinf 13(1):158–167
18. Sáez JA, Luengo J, Herrera F (2016) Evaluating the classifier behavior with noisy data considering performance and robustness: the equalized loss of accuracy measure. Neurocomputing 176:26–35
19. Saez JA, Galar M, Luengo J, Herrera F (2016) INFFC: an iterative class noise filter based on the fusion of classifiers with noise sensitivity control. Inf Fusion 27:505–636
20. Palacios A, Sanchez L, Couso I (2016) An extension of the FURIA classification algorithm to low quality data through fuzzy rankings and its application to the early diagnosis of dyslexia. Neurocomputing 176:60–71
21. Fdez JA, Alonso JM (2016) A survey of fuzzy systems software: taxonomy, current research trends and prospects. IEEE Trans Fuzzy Syst 24(1):40–56
22. Martin D, Fdez JA, Rosete A, Herrera F (2016) NICGAR: a niching genetic algorithm to mine a diverse set of interesting quantitative association rules. Inf Sci 355–356:208–228
23. González M, Bergmeir C, Triguero I, Rodríguez Y, Benítez JM (2016) On the stopping criteria for k-nearest neighbor in positive unlabeled time series classification problems. Inf Sci 328:42–59
24. Morente-Molinera JA, Pérez IJ, Ureña MR, Herrera-Viedma E (2016) Creating knowledge databases for storing and sharing people knowledge automatically using group decision making and fuzzy ontologies. Inf Sci 328:418–434
25. Dong Y, Zhang H, Herrera-Viedma E (2016) Integrating experts' weights generated dynamically into the consensus reaching process and its applications in managing non-cooperative behaviors. Decis Support Syst 84:1–15
26. Fernandez A, Carmona CJ, del Jesus MJ, Herrera F (2016) A view on fuzzy systems for big data: progress and opportunities. Int J Comput Intell Syst 9(1):69–80
27. Peralta D, Triguero I, García S, Herrera F, Benítez JM (2016) DPD–DFF: a dual phase distributed scheme with double fingerprint fusion for fast and accurate identification in large databases. Inf Fusion 32:40–51
28. Galar M, Fernandez A, Barrenechea E, Bustince H, Herrera F (2016) Ordering-based pruning for improving the performance of ensembles of classifiers in the framework of imbalanced datasets. Inf Sci 354:178–196
29. Lozano M, Rodriguez FJ, Peralta D, García-Martínez C (2016) Randomized greedy multi-start algorithm for the minimum common integer partition problem. Eng Appl Artif Intell 50:226–235
30. Cavalcante RG, Patil S, Weymouth TE, Bendinskas KG, Karnovsky A, Maureen A (2016) Sartor ConceptMetab: exploring relationships among metabolite sets to identify links among biomedical concepts. Bioinformatics 32(10):1536–1543
31. Domínguez JG, Schmidt B (2016) ParDRe: faster parallel duplicated reads removal tool for sequencing studies. Bioinformatics 32(10):1562–1564
32. Machado MR, Pantano S (2016) SIRAH tools: mapping, backmapping and visualization of coarse-grained models. Bioinformatics 32(10):1568–1570
33. Burkett KM, McNeney B, Graham J (2016) Sampletrees and Rsampletrees: sampling gene genealogies conditional on SNP genotype data. Bioinformatics 32(10):1568–1570
34. Liu Y, Zhao M (2016) lnCaNet: pan-cancer co-expression network for human lncRNA and cancer genes. Bioinformatics 32(10):1595–1597
35. Meyer MJ, Geske P, Yu H (2016) BISQUE: locus- and variant-specific conversion of genomic, transcriptomic and proteomic database identifiers. Bioinformatics 32(10):1598–2000
36. Lyu Y, Li Q (2016) A semi-parametric statistical model for integrating gene expression profiles across different platforms. BMC Bioinf 17(5):51
37. Ferlay J, Soerjomataram I, Dikshit R, Eser S, Mathers C, Rebelo M et al (2014) Cancer incidence and mortality worldwide: sources, methods and major patterns in GLOBOCAN 2012. Int J Cancer 136(5):E359–E386
38. Sancho-Asensio A, Orriols-Puig A, Casillas J (2016) Evolving association streams. Inf Sci 334–335:250–272
39. Sáez JA, Luengo J, Herrera F (2016) Evaluating the classifier behavior with noisy data considering performance and robustness: the equalized loss of accuracy measure. Neurocomputing 176:26–35
40. Ramentol E, Gondres I, Lajes S, Bello R, Caballero Y, Cornelis C, Herrera F (2016) Fuzzy-rough imbalanced learning for the diagnosis of high voltage circuit breaker maintenance: the SMOTE-FRST-2T algorithm. Eng Appl Artif Intell 48:134–139
41. Ramírez-Gallego S, García S, Mouriño-Talín H, Martínez-Rego D, Bolón-Canedo V, Alonso-Betanzos A, Benítez JM, Herrera F (2016) Data discretization: taxonomy and big data challenge. Wiley interdisciplinary reviews. Data Min Knowl Disc 6(1):5–21
42. Fisher RA (1936) The use of multiple measurements in taxonomic problems. Ann Eugen 7:179–188
43. Verbiest N, Derrac J, Cornelis C, García S, Herrera F (2016) Evolutionary wrapper approaches for training set selection as preprocessing mechanism for support vector machines: experimental evaluation and support vector analysis. Appl Soft Comput 38:10–22
44. Wu A, Wen S, Zeng Z (2012) Synchronization control of a class of memristor-based recurrent neural networks. Inf Sci 183(1):106–116

45. Wu A, Zeng Z (2013) Anti-synchronization control of a class of memristive recurrent neural networks. Commun Nonlinear Sci Numer Simul 18(2):373–385
46. Zhang G, Shen Y (2014) Exponential synchronization of delayed memristor-based chaotic neural networks via periodically intermittent control. Neural Netw 55:1–10
47. Zhang G, Shen Y, Sun J (2012) Global exponential stability of a class of memristor-based recurrent neural networks with time-varying delays. Neurocomputing 97:149–154
48. Zhang G, Shen Y, Yin Q, Sun J (2013) Global exponential periodicity and stability of a class of memristor-based recurrent neural networks with multiple delays. Inf Sci 232:386–396
49. Zhang G, Shen Y, Wang L (2013) Global anti-synchronization of a class of chaotic memristive neural networks with time-varying delays. Neural Netw 46:1–8
50. Zhang G, Shen Y (2013) New algebraic criteria for synchronization stability of chaotic memristive neural networks with time-varying delays. IEEE Trans Neural Netw Learn Syst 24(10):1701–1707
51. Wen S, Zeng Z, Huang T (2012) Exponential stability analysis of memristor-based recurrent neural networks with time-varying delays. Neurocomputing 97:233–240
52. Chen J, Zeng Z, Jiang P (2014) Global Mittag–Leffler stability and synchronization of memristor-based fractional-order neural networks. Neural Netw 51:1–8
53. Wang X, Li C, Huang T, Duan S (2014) Global exponential stability of a class of memristive neural networks with time-varying delays. Neural Comput Appl 24(8):1707–1715
54. Guo Z, Wang J, Yan Z (2013) Global exponential dissipativity and stabilization of memristor-based recurrent neural networks with time-varying delays. Neural Netw 48:158–172
55. Guo Z, Wang J, Yan Z (2014) Attractivity analysis of memristor-based cellular neural networks with time-varying delays. IEEE Trans Neural Netw Learn Syst 25(4):704–717
56. Sun J, Shen Y, Yin Q, Xu C (2013) Compound synchronization of four memristor chaotic oscillator systems and secure communication. Chaos 23(1):013140
57. Bo-Cheng B, Zhong L, Jian-Ping X (2010) Transient chaos in smooth memristor oscillator. Chin Phys B 19(3):030510
58. Wu CW (2001) Synchronization in arrays of coupled nonlinear systems: passivity, circle criterion, and observer design. IEEE Trans Circuits Syst I Fundam Theory Appl 48(10):1257–1261
59. Zhang Y, Wang J, Wang X (2014) Review on probabilistic forecasting of wind power generation. Renew Sustain Energy Rev 32:255–270
60. Quan H, Srinivasan D, Khosravi A (2015) Incorporating wind power forecast uncertainties into stochastic unit commitment using neural network-based prediction intervals. IEEE Trans Neural Netw Learn Syst 26(9):2123–2135
61. Yuan Y, Mou L, Lu X (2015) Scene recognition by manifold regularized deep learning architecture. IEEE Trans Neural Netw Learn Syst 26(10):2222–2233
62. Zhang W, Tang Y, Wong WK, Miao Q (2015) Stochastic stability of delayed neural networks with local impulsive effects. IEEE Trans Neural Netw Learn Syst 26(10):2336–2345
63. Chang C (2015) Deep and shallow architecture of multilayer neural networks. IEEE Trans Neural Netw Learn Syst 26(10):2477–2486
64. Yang JB, Singh MG (1994) An evidential reasoning approach for multiple attribute decision making with uncertainty. IEEE Trans Syst Man Cybern 24(1):1–18
65. Yang JB, Sen P (1994) A general multi-level evaluation process for hybrid MADM with uncertainty. IEEE Trans Syst Man Cybern 24(10):1458–1473
66. Yang JB (2001) Rule and utility based evidential reasoning approach for multi-attribute decision analysis under uncertainties. Eur J Oper Res 131(1):31–61