Optimizing Feature Selection via a Top-Down Dimensionality Reduction Method

Mehdi Ayar
Department of Computer Engineering, Shab.C., Islamic Azad University, Shabestar, Iran
mehdiayar@iau.ac.ir

Receive Date: 20 May 2025 Revise Date: 11 June 2025 Accept Date: 29 June 2025

Abstract

Feature selection is a main step in the data classification process, and it is applied to determine optimal features and develop an accurate model. This paper presents a novel top-down correlated data reduction method which is employed by feature selection process. The method removes redundant features by clustering and placing correlated features in a cluster. This operation continues until reaching a certain and desired number of features. The proposed feature selection method resulted in performance rates of 88.21%, 89.41%, 87.64%, and 86.54% in terms of accuracy, sensitivity, specificity, and f-measure, respectively which are highly effective compared to the current state-of-the-art approaches.

Keywords: Correlated Features, Data Reduction, Feature Selection, Top-Down Method

1. Introduction

Since feature selection can select the class-related features and remove redundant and noisy features, it is one of the essential steps in the classification problem. Removing redundant features, for example, would make the classification possible with fewer features and resulted in time saving in the classification process. The redundant features do not have any significant impact on the result of the classification. Furthermore, noise features need to be eliminated to be prevented the confusion of the classification algorithm. According to [1], wrapper methods are a useful way for removing noise features. They examine various combinations of features with different classifiers, so they are considered as time-consuming methods. Several criteria can be used to evaluate the correlation between two individual features and to distinguish redundant ones. Some of these criteria include mutual information, correlation coefficient, chi-square, and distance functions, such as the Euclidean distance. However, pair-wise evaluation of the correlation between features is a time-consuming way especially when the number of features is high.

This study presents a top-down method for the feature selection problem by reducing the number of correlated features using feature clustering. The proposed algorithm can potentially be employed for large feature sets while reducing the time complexity of the algorithm. There is no need to examine the pair-wise dependency of the features when using this algorithm compared to conventional methods.

Related Work

The proliferation of high-dimensional datasets in domains such as genomics, computer vision, and sensor networks has intensified the need for robust preprocessing techniques to address computational inefficiency, overfitting, and interpretability challenges. Data reduction and feature selection are two foundational strategies to mitigate these issues, each offering distinct advantages depending on the application

context. This section synthesizes classical and modern methodologies in both categories, emphasizing their mathematical principles, comparative strengths, and practical applications.

Data Reduction Techniques

Data reduction transforms high-dimensional data into a lower-dimensional manifold while preserving critical structural or statistical relationships. These methods are broadly categorized as linear or non-linear.

Linear methods, such as Principal Component Analysis (PCA), project data onto orthogonal axes (principal components) that maximize variance [1]. Despite its computational efficiency, PCA assumes linearity and Gaussianity, limiting its utility for non-linear datasets like images or text [2]. Linear Discriminant Analysis (LDA) addresses supervised scenarios by maximizing class separability through scatter matrix optimization [3]. However, LDA struggles with high-dimensional, non-linear data and singularity issues [4].

Non-linear methods like t-Distributed Stochastic Neighbor Embedding (t-SNE) and Uniform Manifold Approximation and Projection (UMAP) have gained prominence for visualizing complex data. t-SNE minimizes the Kullback-Leibler divergence between pairwise similarities in high- and low-dimensional spaces, excelling at cluster visualization [5]. UMAP improves scalability using fuzzy topological representations [6], making it indispensable in single-cell RNA sequencing [7]. Neural network-based autoencoders further advance non-linear reduction by training encoder-decoder architectures to reconstruct inputs. Variants like Variational Autoencoders (VAEs) introduce probabilistic latent spaces for generative modeling [8], while Convolutional Autoencoders excel in image denoising [9]. These methods require large datasets and regularization to avoid overfitting. Matrix factorization techniques, such as Non-Negative Matrix Factorization (NMF), decompose data into interpretable, parts-based representations (e.g., topics in text mining) [10] but are sensitive to missing data.

Feature Selection Strategies

Feature selection identifies optimal subsets of original features, retaining interpretability critical for domains like healthcare. Three primary paradigms dominate this field: filter, wrapper, and embedded methods [11]. Filter methods rank features using statistical metrics independent of the learning algorithm. For example, mutual information quantifies non-linear dependencies between features and targets [12], while ANOVA F-scores evaluate linear relationships. Though computationally efficient, filter methods ignore feature interactions and model-specific synergies. Wrapper methods, such as Recursive Feature Elimination (RFE), iteratively train models to evaluate feature subsets, removing low-importance features (e.g., SVM weights) until optimal accuracy is achieved [13]. Genetic algorithms employ evolutionary operations (mutation, crossover) to explore subsets, optimizing objectives like AUC-ROC [14]. Embedded methods integrate selection into model training. LASSO regression applies L1 regularization to induce sparsity, discarding irrelevant features [15].

Feature Selection via Clustering Strategies

In recent years, numerous methods have been developed for feature selection based on clustering and correlation analysis. For instance, the *Correlation-Based Feature Selection with Clustering* approach utilizes the correlation coefficients among features to form clusters and select representative features accordingly [21]. Similarly, the *Region and Correlation-based Heuristic Feature Selection with Clustering Analysis (RCH-FSC)* method constructs a correlation matrix and applies clustering techniques to identify relevant features and extract an optimal subset [22]. In addition, the *Correlated Clustering and Projection (CCP)* method partitions features into highly correlated clusters and projects each cluster into a one-dimensional space to achieve effective dimensionality reduction without relying on matrix decomposition [23]. While these approaches adopt a predominantly bottom-up perspective, the method proposed in this study introduces a top-down, hierarchical framework to guide the feature selection process from a higher-level abstraction.

Comparative Analysis

The choice between data reduction and feature selection hinges on trade-offs between interpretability, computational resources, and data complexity (Table 1). Reduction techniques like PCA excel at handling multicollinearity but obscure feature meaning, whereas feature selection methods like LASSO retain original variables for domain-specific analysis [19]. Hybrid approaches (e.g., PCA + RFE) combine dimensionality reduction with iterative selection for high-dimensional datasets [20].

Table.1. Comparative analysis and applications of some classical and modern methodologies

Method	Strengths	Weaknesses	Applications
PCA	Fast; Handles multicollinearity	Linear assumptions	Image compression, Finance
t-SNE	Captures non-linear clusters	Computationally heavy	Single-cell RNA sequencing
LASSO	Sparse solutions; Interpretable	Sensitive to correlated features	Genomics, Economics
Random Forests	Robust to outliers; Non-linear	Black-box nature	Remote sensing, Marketing

2. The Proposed Top-Down Method

The correct selection of features required to design a realistically decision model for a system. In fact, removing redundant and irrelevant features simplifies the model, reduces training time, and reduces over-fitting. An over-fitted model significantly depends on the data and probably has a higher error rate on unseen data. Due to the enormous number of features and the large number of calculations, calculating pair-wise similarity of features to find similar ones and keeping just one of them is almost impossible.

This paper presents a method which has a top-down algorithm as its central core. The top-down property of the algorithm allows the method to be suitable for applications with many features. In this algorithm, the

feature set is clustered based on correlation and a representative feature is then selected. The feature clustering based on correlation and selecting class-related representatives for each cluster can remove extra features. Fig.1and Fig.2show the schematic and the pseudocode of the method, respectively.

First, we identified those features which statistically behave similar and assigned them to a single cluster. Then, we selected a feature from each cluster which could represent the whole members while discarded the rest. This process continued until there was no improvement in the selected features, i.e.,the representative features were same as in the previous step. To determine the behavioral similarity between the features, the correlation coefficient between feature x and y, i.e., Cor (x,y), was calculated using (1) while assuming n objects.

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

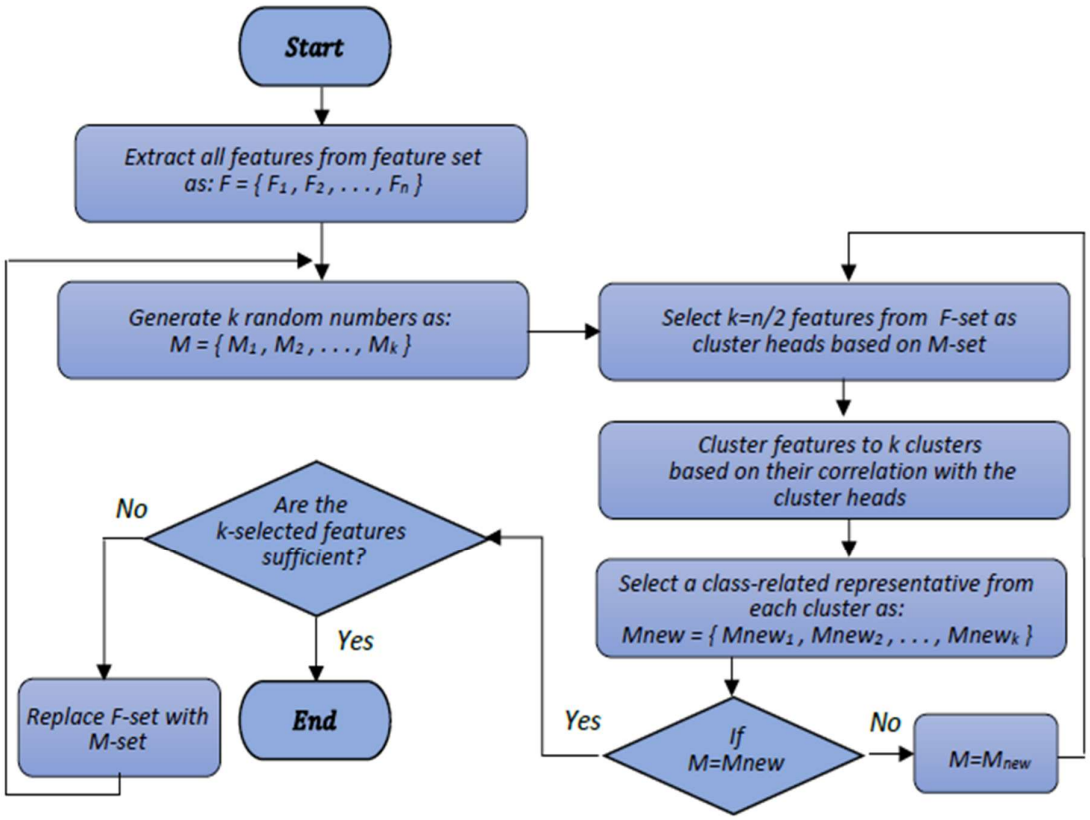

Fig. 1. Schematic view of the proposed Algorithm

```
The Proposed Method
1     Data = Read the Data set
2     K = Number of required features
3     Kc = inf
4     while (Kc > K)
5     {
6        Kc = max (((Number of features in Data)/2), K)
7        SF = SelectFeatures (Data, Kc, ExpClass)
8        Data = Data[: , SF]
9     }
10    ------------------------------------
11    SF = SelectFeatures (Data, Kc, ExpClass)
12    {
13       Col = Number of Data Columns
14
15       M[1..Kc] = Generate Kc Random Numbers
16    while (True)
17    {
18    fori = 1 to Col (for each feature in Data)
19    {
20    for j = 1 to Kc (for each cluster)
21                Cor[j]=Correlation(Data[ , i] ,Data[: , M[j]])
22    if Cor[j] == max (Cor[1] ,Cor[2] , …. ,Cor[Kc])
23                Add feature Fi to cluster Cj
24    }
25    for j = 1 to Kc (for each cluster)
26    {
27             Members = All members of the cluster Cj
28             LC = Length of the cluster Cj (Members)
29    fori=1 to LC (for each member in the cluster)
30    {
31                mem=members[i]
32    Cor[i]=Correlation(Data[: , mem], ExpClass)
33    }
34    if Cor[i] == max (Cor[1] ,Cor[2] ,… ,Cor[LC])
35    Mnew[j] = members[i]
36    }
37    ifMnew == M
38    Break while loop
39    Else
40             Replace M with Mnew and Empty all clusters
41    }
42       SF = Mnew
43    }
```

Fig. 2. Algorithm of the proposed method

The main section begins after initialization (see lines 4 to 9) as indicated in Fig. 1 and Fig. 2. In this section, by using the SelectFeatures() either selection of n/2 good features or removal of n/2 redundant features would be occurred in each iteration (n is the number of features in the dataset). In fact, the SelectFeatures() first generates random numbers with the half number of features (line 15). These numbers show the features which are initially chosen at random as cluster heads. In a while loop using two nested for loops, the correlation values of the remaining features with the cluster-heads are calculated. Any elements with highest correlation with the cluster-heads are placed into its corresponding cluster (see lines 18 to 24). In the next step, to select a representative from each cluster, the correlation value of each cluster element is calculated with the class label for each cluster. Therefore, in lines 25 to 36, the feature with the highest correlation with the class label is chosen as the representative. Finally, if the selected features are the same in two consecutive repetitions, the operation ends; otherwise, it will continue with the new cluster heads (see lines 37 to 41). This

operation would be repeated with n/2 new features to get n/4 optimal features.

Fig. 3 illustrates an example function of the proposed algorithm. First, five cluster-heads were selected based on a random way (F2, F3, F7, F8, and F9). Next, other features based on the highest correlation with these cluster-heads were placed in related clusters. Then, a representative feature with highest correlation to class feature was selected from each cluster as a new cluster-head (F1, F3, F7, F6, and F9). This time the same operation was performed with new cluster-heads. When the representative features were matched with the cluster-heads of the previous step, the operation was completed and the selected features were specified (F1, F3, F10, F5, F9).

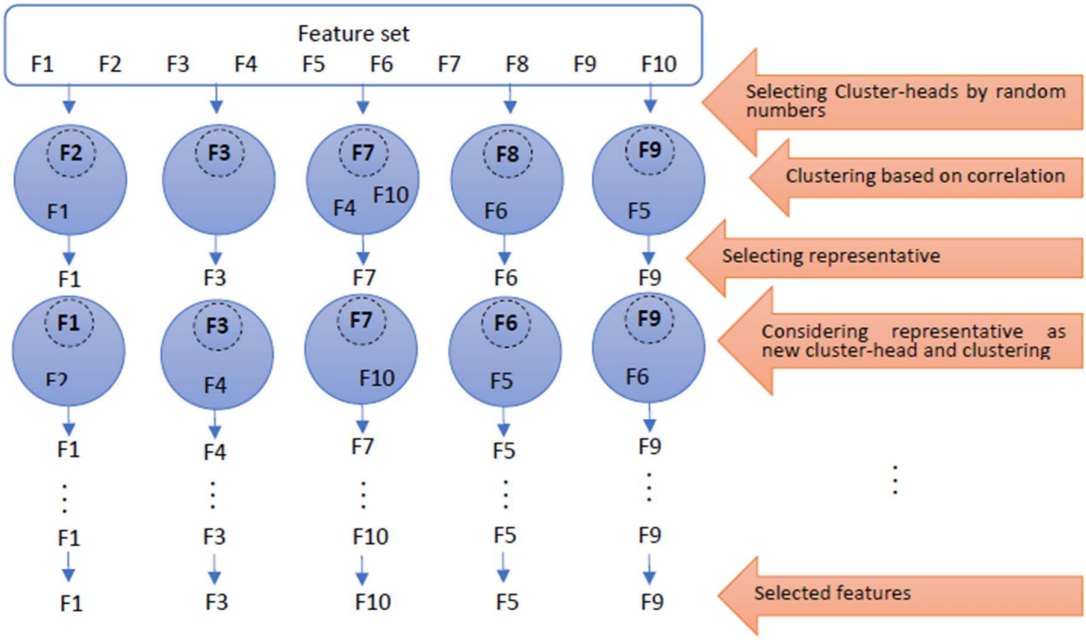

Fig.3. An example of the proposed algorithm functionality

3. Using the Proposed Algorithm for Classification

The algorithm can be used wherever optimal features need to be selected. This section examines removing additional features based on the proposed algorithm so that the data can be classified more accurately.

The Feature Selection

This research examined 5 to 60 features where for all features the algorithm was executed a hundred times, and then we calculated the average for each execution. For example, to select ten optimal features from 156 available features, 78 random numbers (half the number of elements) were first generated using a chaos map, e.g., a logistic map. These 78 numbers represent the 78 features that will be the cluster heads.

Then, the correlation coefficient between the 156 elements and each of the cluster heads was calculated, and the item was placed in the cluster-head's group with which it has the highest correlation. Ultimately, 78 clusters were identified, and one representative was selected from each cluster. These representatives will be the new cluster-heads, and again the correlation between all the features and these heads will be examined. The same process continues until no difference remains between the current and previous results. This operation halved the number of features from 156 to 78. Again, the same process continues with the remaining 78 features and the elimination of the other half. Finally, the algorithm resulted in the last ten features as the optimal features.

The Classification Stage

In this paper, classification was performed in two classes of the dataset (normal and abnormal). Among different ways for data classification such as decision trees, SVM, and Naïve Bayes, the decision tree was chosen regardless of the strengths or weaknesses of the classifier. To this end, we selected 80% of the total data as a training set and 20% as a testing set. First, we trained the model using the training data, and then tested it using the testing data.

4. Experimental Results

To evaluate the effectiveness of the algorithm in selecting optimal features and its application in the classification of cardiac arrhythmias, we implemented this algorithm in MATLAB.

Table 2. Average of evaluation parameters after a hundred execution.

Row	Number of Features	Purity	NMI	Accuracy	F- measure	Sensitivity	Specificity
1	27	85.67	72.62	87.05	84.32	86.70	87.23
2	31	81.13	64.76	82.36	81.27	85.80	80.73
3	32	81.89	66.00	82.53	81.16	85.40	81.17
4	36	86.03	73.32	87.52	85.04	87.48	87.54
5	37	85.53	72.40	86.88	83.64	85.74	87.43
6	38	85.77	73.14	87.00	84.24	86.75	87.14
7	44	85.98	73.11	87.33	84.92	87.48	87.24
8	46	**86.33**	74.21	**87.92**	**85.23**	87.51	**88.15**
9	47	**86.43**	**74.41**	87.89	84.98	87.21	**88.19**
10	48	85.80	73.74	87.71	85.05	87.46	87.85
11	49	83.12	67.79	83.26	82.22	86.27	81.79
12	52	83.39	68.88	84.66	82.18	85.74	84.13
13	53	**87.06**	**75.06**	**88.21**	**86.54**	**89.35**	87.59
14	54	84.00	69.68	84.88	82.36	85.36	84.62
15	55	86.16	73.95	87.59	**85.12**	87.81	87.46
16	56	85.29	71.37	85.45	84.50	**88.15**	84.10
17	57	84.83	70.57	84.88	84.18	88.12	83.18
18	58	85.71	72.97	86.77	84.97	88.13	86.16
19	59	84.86	70.84	85.49	84.14	87.68	84.47
20	60	84.86	71.18	85.41	84.42	**88.17**	84.13

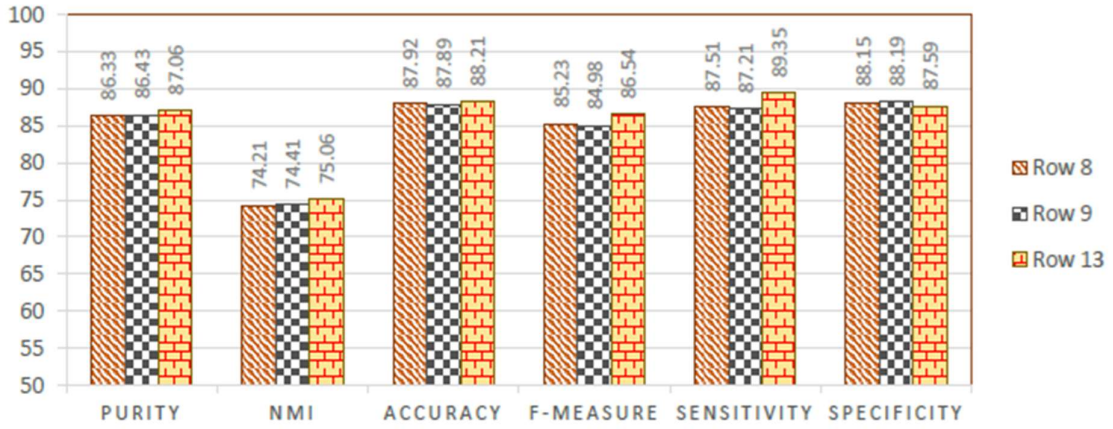

Fig. 4. Comparisonof three selected rows in table 2 from the perspective of the evaluation parameter

We executed the proposed algorithm to find the optimal set of features (5 to 60), hundred times for each set, and in two modes, i.e., with and without using chaos sequences. Finally, we used the decision tree to train the model and classify the arrhythmias. The average efficiency of the features selected after one hundred executions is presented in Table 2. In each column of this table, the top values were distinguished. Because of having the best values in parameters such as purity, NMI, accuracy, F-measure, sensitivity, and specificity, rows 8, 9, and 13 are our options.

Table 3. Comparison of the proposed algorithm and other methods

Proposed algorithm	Test Method	Classification Type	Evaluation Results
Feature elimination based random subspace ensembles learning	Hold-out	Binary	Accuracy= 91.11
Soft-Margin SVM, Feature Selection using Improved Elitist GA and 10-fold CV SVM fitness function	10-Fold Cross-Validation	Binary	Accuracy= 87.83
Neural Network Models	Hold-out	Binary	Accuracy= 86.67
Top-Down Dimensionality Reduction Method (This Study)	Hold-out	Binary	Accuracy= 88.21, Sensitivity= 89.4 Specificity= 87.64, Purity= 87.06 NMI= 75.06, F-measure= 86.54

Discussion

To highlight the performance of the proposed method, we compared the achieved results with the current methods' achievements in Table 3. We considered different perspectives, such as the proposed algorithm, test method, classification type, and evaluation criteria results. It can be said that despite the same test and classification, the proposed method is about 5.71% more accurate than the best method in all-classes classification. Several important criteria besides accuracy such as NMI, F-measure, specificity, sensitivity, classification time, and so on were reported in our paper for the first time.

5. Conclusion

This paper proposed a novel feature selection method based on chaos theory and divide-and-conquer paradigm. Although the proposed algorithm is application-independent, we used it for feature selection and classification of cardiac arrhythmias to evaluate its efficiency in biomedical applications. The UCI arrhythmia database was utilized in this research whereas various experiments selected 53 important features from the 279 features of this database. Furthermore, this method improved some parameters such as accuracy and F-measure by an average of 0.83% and 0.82%, respectively, which led to an enhanced performance of the proposed algorithm. The proposed method largely addressed filter-mode and wrapper-mode methods weaknesses. We plan to examine the impacts of chaotic maps on the effectiveness of the proposed method in the future.

References

[1] M. Ayar, and S. Sabamoniri, "An ECG-based feature selection and heartbeat classification model using a hybrid heuristic algorithm," Informatics in Medicine Unlocked, vol. 13, pp. 167-175, 2018.

[2] S. Chen, W. Hua, Z. Li, J. Li, and X. Gao, "Heartbeat classification using projected and dynamic features of ECG signal," Biomedical Signal Processing and Control, vol. 31, pp. 165-173, 2017.

[3] K. N. Rajesh, and R. Dhuli, "Classification of ECG heartbeats using nonlinear decomposition methods and support vector machine," Computers in biology and medicine, vol. 87, pp. 271-284, 2017.

[4] M. Balouchestani, and S. Krishnan, "Advanced K-means clustering algorithm for large ECG data sets based on a collaboration of compressed sensing theory and K-SVD approach," Signal, Image and Video Processing, vol. 10, no. 1, pp. 113-120, 2016.

[5] S. Dilmac, and M. Korurek, "ECG heart beat classification method based on modified ABC algorithm," Applied Soft Computing, vol. 36, pp. 641-655, 2015.

[6] S. M. Jadhav, S. L. Nalbalwar, and A. A. Ghatol, "Artificial neural network models based cardiac arrhythmia disease diagnosis from ECG signal data," International Journal of Computer Applications, vol. 44, no. 15, pp. 8-13, 2012.

[7] S. Goel, P. Tomar, and G. Kaur, "A fuzzy based approach for denoising of ECG signal using wavelet transform," Int J Bio-Sci Bio-Technol, vol. 8, no. 2, pp. 143-156, 2016.

[8] D. Pal, K. Mandana, S. Pal, D. Sarkar, and C. Chakraborty, "Fuzzy expert system approach for coronary artery disease screening using clinical parameters," Knowledge-Based Systems, vol. 36, pp. 162-174, 2012.

[9] D. K. Atal, and M. Singh, "Arrhythmia Classification with ECG signals based on the Optimization-Enabled Deep Convolutional Neural Network," Computer Methods and Programs in Biomedicine, pp. 105607, 2020.

[10] Q. Wu, Y. Sun, H. Yan, and X. Wu, "Ecg signal classification with binarized convolutional neural network," Computers in Biology and Medicine, pp. 103800, 2020.

[11] I. Guler, and E. D. Ubeyli, "Multiclass support vector machines for EEG-signals classification," IEEE transactions on information technology in biomedicine, vol. 11, no. 2, pp. 117-126, 2007.

[12] B. Tripathy, D. Acharjya, and V. Cynthya, "A framework for intelligent medical diagnosis using rough set with formal concept analysis," International Journal of Artificial Intelligence & Applications (IJAIA), vol. 2, no. 2, 2013.

[13] S. Dalal, and R. Birok, "Analysis of ECG signals using hybrid classifier," International Advanced Research Journal in Science, Engineering and Technology, vol. 3, no. 7, pp. 89-95, 2016.

[14] S. Raj, K. C. Ray, and O. Shankar, "Cardiac arrhythmia beat classification using DOST and PSO tuned SVM," Computer methods and programs in biomedicine, vol. 136, pp. 163-177, 2016.

[15] İ. Kayikcioglu, F. Akdeniz, C. Köse, and T. Kayikcioglu, "Time-frequency approach to ECG classification of myocardial infarction,"

Computers & Electrical Engineering, vol. 84, pp. 106621, 2020.

[16] X. Song, G. Yang, K. Wang, Y. Huang, F. Yuan, and Y. Yin, "Short Term ECG Classification with Residual-Concatenate Network and Metric Learning," MULTIMEDIA TOOLS AND APPLICATIONS, 2020.

[17] X. Zhaia, Z. Zhoua, and C. Tina, "Semi-Supervised Learning for ECG Classification without Patient-specific Labeled Data," Expert Systems with Applications, pp. 113411, 2020.

[18] A. M. Shaker, M. Tantawi, H. A. Shedeed, and M. F. Tolba, "Generalization of Convolutional Neural Networks for ECG Classification Using Generative Adversarial Networks," IEEE Access, vol. 8, pp. 35592-35605, 2020.

[19] G. Boeing, "Chaos Theory and the Logistic Map. 2015," URl: http://geoffboeing. com/2015/03/chaos-theory-logistic-map, 2016.

.

[20] D. Dheeru, and G. Casey, "{UCI} Machine Learning Repository," 2017.

[21] S. Chormunge and S. Jena, "Correlation based feature selection with clustering for high dimensional data," J. Electr. Syst. Inf. Technol., vol. 5, no. 3, pp. 542–549, Sep. 2018, doi: 10.1016/j.jesit.2017.06.004.

[22] A. Atmakuru, G. Di Fatta, G. Nicosia, and A. Badii, "Improved filter-based feature selection using correlation and clustering techniques," in Proc. Int. Conf. Mach. Learn., Optim. Data Sci. (LOD), Naples, Italy, Sep. 2023, pp. 379–389, doi: 10.1007/978-3-031-53969-5_28.

[23] Y. Hozumi, R. Wang, and G.-W. Wei, "CCP: Correlated clustering and projection for dimensionality reduction," arXiv preprint arXiv:2206.04189, Jun. 2022. [Online]. Available: https://arxiv.org/abs/2206.04189