International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V70I5P213

Volume 70 Issue 5, 103-110, May 2022

Original Article

Feature Raking and Stacked Sparse Autoencoder based Framework for the Prediction of Breast Cancer

Atul Kumar Ramotra[1], Vibhakar Mansotra[2]

[1, 2] *Department of Computer Science & IT, University of Jammu, Jammu, J&K, India.*

[1]ramotraatul@gamil.com

Received: 07 February 2022 Revised: 21 March 2022 Accepted: 19 April 2022 Published: 19 May 2022

Abstract *- Achieving higher classification accuracy using machine learning is a challenging process. It is important to understand each input variable's significance and contribution to the target class to accomplish this goal. Learning from the suitable representation of the original feature set also enhances the performance of the learning algorithms. This work proposes a framework based on the feature ranking and feature learning techniques for the prediction of Breast Cancer. The main components of the proposed framework include ranking the input variables using the Pearson Correlation method and feature representation of the dataset using Stacked Sparse Autoencoder. The experimental result shows that the proposed framework has achieved an accuracy of 98.42%.*

Keywords *- Disease Prediction, Breast Cancer, Feature Ranking, Pearson Correlation, Stacked Sparse Autoencoder.*

1. Introduction

Health risks related to the lives of patients suffering from chronic diseases can be reduced using reliable and accurate disease prediction systems. Extracting useful information from healthcare datasets is important for the diagnosis of diseases. Predictive analysis of chronic disease using machine learning techniques is a challenging process. The presence of many input variables brings additional complications in achieving accurate predictive results. The relationship between the input variables and the target variable is often complex. Intelligent methods and approaches are required to analyze the dependence of related factors responsible for causing the disease. Models and frameworks developed using machine learning techniques help predict various chronic diseases at an early stage [1,2]. The solutions provided by these techniques are inexpensive in terms of healthcare costs and require less computational time [3].

The feature ranking method assigns values to the features of the input data according to an evaluation criterion [4]. A high score indicates high relevance of the attribute concerning the target class. Features with lesser scores share minimum contribution in the prediction of the disease and complicate the process of predictive analysis; hence can be removed from the dataset. The form in which the learning algorithms represent data plays a major role in classification problems. Machine learning offers many supervised and unsupervised feature learning techniques to determine the effective way of data representation [5]. The performance of learning algorithms can be improved using these techniques. The application of autoencoder (AE) as a feature learning and representing technique has gained popularity among researchers [6]. AE learns the input data representation using an unsupervised learning technique. This work proposes the combination of feature ranking and feature learning techniques to perform the accurate prediction of Breast Cancer. The work includes the application of Pearson Correlation (PC) to rank and eliminate the low scoring features of the original dataset, unsupervised learning-based feature learning of the new feature set using Stacked Sparse Autoencoder (SSAE) to obtain an improved feature space, and the final classification task is done using a softmax layer.

2. Autoencoder

An AE is a feedforward neural network that encodes the original data using an unsupervised learning method [7,8]. Encoder and decoder are the two main components of AE, as shown in Fig. 1. The Encoder transforms the original input vector into code in the hidden layer, and the decoder reconstructs the input data with the help of code using (1) and (2), respectively.

$$h(x) = f(W_1 x + b_1) = g \quad (1)$$

$$z(g) = f(W_2 g + b_2) = x' \quad (2)$$

Where x represents the input vector, f represents the activation functions used for the transformation, and [W_1, W_2, b_1,b_2] represents the weight matrices and the bias vectors of the neurons connected in the network.

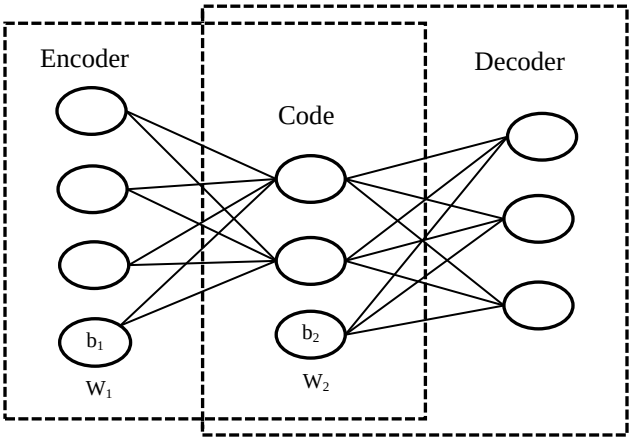


Fig. 1 The basic architecture of an Autoencoder

Multiple encoding and decoding layers can be added to the AE network but the des, but these layers' designs are retained symmetrically. However, the number of nodes in the code can be higher or lower than the encoding layer. AE can also suffer from merely copying the input data through the network. To avoid this problem, a restriction or sparsity is introduced, which is a small penalty added to the layers of AE. Weight regularization is also added to the AE network to avoid overfitting [9,10]. An AE can be used to achieve both linear and nonlinear transformations of the input data.

2.1. Types of Autoencoders

According to the type of functionality they offer [11,12], AEs are classified into the following main types:

2.1.1. Denoising autoencoder (DAE)

DAE purposely introduces noise in the original data to generate a corrupted copy of the data. The noise is created by setting the values of a few randomly selected input features to zero. The main purpose of corrupting the input is to obtain a robust representation suitable for recovering the original undistorted input.

2.1.2. Sparse Autoencoder (SAE)

In SAE, sparsity is introduced by adding some regularization in the layers of AE to penalize the total cost function for certain behaviors.

2.1.3. Variational Autoencoder (VAE)

It is a generative model which modifies the process of input encoding from a single-point approach to a probability distribution close to normal in latent space.

2.1.4. Stacked Autoencoder (SA)

Multiple AEs are stacked together to design SA to add more hidden layers. Layer by layer pertaining to the SA is done, and the first layer is provided as input to the next layer.

2.1.5. Convolutional Autoencoder (CAE)

This type of AE is specially designed to handle image data by replacing the densely connected layers of the network with the convolutional layers.

Different types of AEs can also be designed by using these basic functionalities as layers of the AE network. Many researchers have suggested the employment of AEs for resolving complications related to classification tasks in health care and other domains. Zenbout et al. [13] implemented SSAE to learn new feature representations from miRNA data set for the classification of 31types of cancer and achieved 95% of accuracy. Pathirage et al. [14] applied SAE to identify structural damage. A pertained Deep Neural Network is added to the SAE to extract useful relationships from different parameters. Wang et al. [15] used a combination of DAE and convolutional neural networks for the modulation classification of signals generated by the GNU radio toolkit. Law et al. [16] proposed using SA for extracting underlying properties of the original features and reducing the dimensions of the data.

The authors concatenated the SA network with another network of an extreme learning machine to enhance the performance of the multi-label classification tasks. Li et al. [17] developed a hybrid model for the diagnosis of faults occurring in the process of distillation and attained 92.20% of accuracy. The authors used a convolutional neural network for feature extraction and SAE for classification in the hybrid model. Lu et al. [18] proposed a multi-task learning model for sentiment analysis using VAE. The model employed long, and short-term memory as encoder and multilayer perceptron as the decoder. Chen et al. [19] pointed out the region of interest labeling problems and inadequacy of labeled medical computed tomography image data and proposed a solution based on CAE. The proposed solution required a large amount of unlabeled data for training and less labeled data for fine-tuning to predict the presence of pulmonary lung nodules. Farahnakian et al. [20] designed a DA-based intrusion detection system and evaluated the system using the KDD-CUP'99 dataset. The authors claimed to have obtained 94.71% detection accuracy using the proposed approach. Ng et al. [21] proposed a new method for handling the classification problem of imbalanced datasets using AE. Dual autoencoding features are created by combining different data characteristics obtained from two separate SA networks employing two different activation functions, tanh and sigmoid. The results so obtained are compared with resampling-based methods. Aslam et al. [22] proposed applying SSAE architecture for the early prediction and classification of gastric cancer based on breath analysis. The authors claimed to have achieved the accuracy of 98.7% for advanced classification of gastric cancer and 97.3% accuracy for early detection of gastric cancer.

3. Proposed Framework

In this study, a new framework has been developed using the feature raking technique and SSAE to predict breast cancer. The architecture of the proposed framework is presented in Fig. 2.

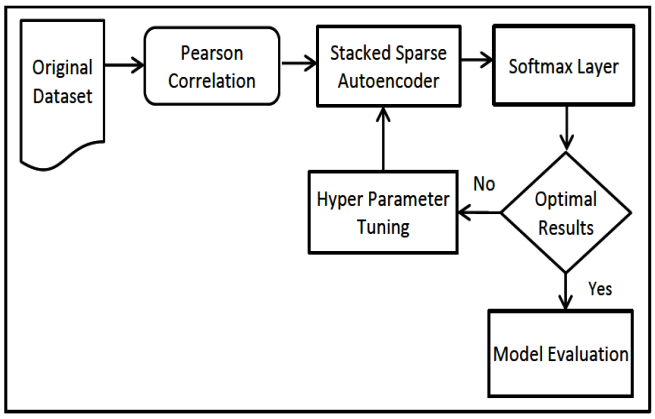


Fig. 2 Architecture of the proposed framework

In the first step, the PC feature ranking technique is applied to the original dataset, and the input features are ranked according to the scores obtained. PC measures the correlation between the input variable x and the target variable y using

(3)

$$PC = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad (3)$$

Where n denotes the total number of samples, x_i and y_i are the sample points and $\bar{x}$ and $\bar{y}$ represent the mean values. The range of values obtained from PC lies between [-1, 1]. After calculating the correlation values between the input variables and the target classes using PC, some features with minimum correlation scores are dropped as they are the least contributing features in predicting the disease. SSAE is applied to the new dataset obtained from the previous step in the next step. Layer by layer pertaining to the SSAE is done using an unsupervised learning technique. The Encoder parts of two Sparse AEs are connected in succession to construct the SSAE network. The new dataset features created in the first step are given as input to the first SAE, as shown in Fig. 3(a), and unsupervised training is done. The SAE1 transforms the features of the new dataset into the feature set 1. Then this feature set 1 is used as the input to the second SAE, as shown in Fig. 3(b), and again unsupervised training is done to reduce the reconstruction error. The SAE2 transforms feature set 1 into the feature set 2. Fig. 3(c) shows the complete structure of the SSAE. A softmax layer is added to the framework for the final classification task. Minimizing the reconstruction error is achieved by optimizing the set of parameters for each SAE, which comprises the number of nodes in the layers, sparsity coefficient (β), regularization coefficient (λ), and sparsity proportion.

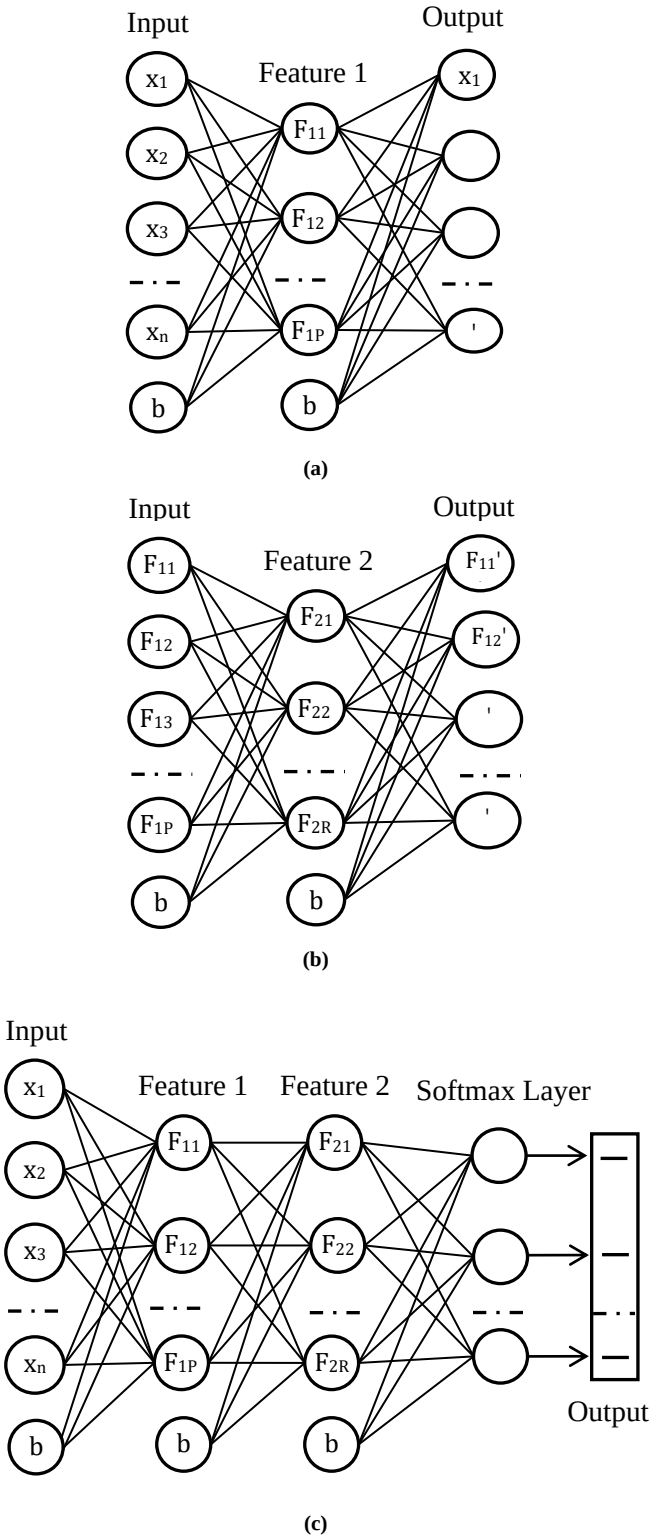


Fig. 3 (a) Sparse Autoencoder 1 (b) Sparse Autoencoder 2 (c) Structure of the Stacked Sparse Autoencoder

The supervised learning method is applied to fine-tune the entire framework for performance enhancement using the training data by minimizing the total error function given as:

$$E_{Total} = E_{MSE} + \beta * E_{Spr} + \lambda * E_{Rglr} \quad (4)$$

Where E_{MSE} is the mean square error function, E_{Spr} represents the sparsity factor, and E_{Rglr} represents the l_2 weight regularization factor calculated using (5), (6), and (7), respectively.

$$E_{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \quad (5)$$

Where Y_i is the observed value and $\hat{Y}_i$ denotes the predicted value.

Kullback-Leibler (KL) divergence introduces sparsity to the cost function.

$$E_{Spr} = \sum_{i=1}^{k} p \log\left(\frac{p}{\hat{p}_i}\right) + (1-p)\log\left(\frac{(1-p)}{(1-\hat{p}_i)}\right) \quad (6)$$

Where k denotes the number of neurons in a layer, p and $\hat{p_i}$ represent the desired and the average activation values of a neuron i in the network.

$$E_{Rglr} = \frac{1}{2}\sum_{i}^{h}\sum_{j}^{n}\sum_{l}^{f}(w_{jl}^{i})^2 \quad (7)$$

In (7), h represents the number of hidden layers, n stands for the number of samples, f shows the number of features in the input data, and the term w_{jl}^{i} represents the connection of the l-th neuron in (i-1) layer with an l-th neuron in i layer. E_{Rglr} is the weight regularization used to prevent overfitting by controlling the network's weights.

4. Experimentation And Discussion

4.1. Dataset and Experimental setup

This study used Wisconsin Diagnostic Breast Cancer Dataset (WDBC) obtained from the UCI Machine Learning dataset repository [23]. The dataset contains 32 attributes and 569 entries with no missing values. Of these 32 attributes, 2 attributes show the ID number and Diagnosis (Malignant (212 entries) and Benign (357 entries)) of each record. The mean, standard error, and extreme value of 10 real-valued attributes (radius, Texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, fractal dimension) are evaluated from the digitized fine needle aspirate image of breast mass, representing the information related to cell nuclei, which results in other 30 attributes (in this study these 30 attributes are assigned names from V_1 to V_{30} for convenience). The framework has been designed using Matlab 2018b environment running on Core(TM) i7 processor, 3.40 GHz CPU, 4 GB RAM, and 64 bit Microsoft 8.1 operating system.

4.2. Evaluation Method

Precision, recall, f1-score, and accuracy values calculated from the confusion matrix are used to measure the overall performance of the models. The confusion matrix contains the values of True Positive, True Negative, False Positive, and False Negative.

True Positive (α): Actual positive and predicted as positive.
True Negative (γ): Actual negative and predicted as negative.
False Positive (μ): Actual negative predicted as positive.
False Negative (δ): Actual positive predicted as negative.

$$\text{Precision} = \alpha / (\alpha + \mu)$$

$$\text{Recall} = \alpha / (\alpha + \delta)$$

$$F_1\text{-score} = \alpha / (\alpha + 1/2 * (\alpha + \delta))$$

$$\text{Accuracy} = (\alpha + \gamma) / (\alpha + \gamma + \mu + \delta)$$

4.3. Results and Discussion

The original dataset contained 32 features. The attribute ID number is dropped from the dataset as it is not important for predicting the disease, and the attribute Diagnosis is considered the target variable for the study. PC is applied to the remaining 30 attributes to calculate the correlation between the features and the class labels. Features obtaining low PC scores are dropped from the original dataset, generating a new dataset. Table 1 shows the correlation values attained by the features removed from the dataset. Fig. 4 shows the ranking of the features in descending order of absolute values obtained using PC on all the features of the original dataset. Min-max normalization is used as a feature scaling method for rescaling the feature values [0, 1].

Table 1. Set of features dropped from the original dataset using Pearson Correlation score

Attribute	Pearson Correlation score
V_{10}	0.0128
V_{12}	0.0083
V_{15}	0.0670
V_{19}	0.0065
V_{20}	0.0779

In the next step, the new dataset features are given as input to the rest of the framework for the prediction of breast cancer. K-fold cross-validation is used to validate the results, and the value of k is taken as 5.

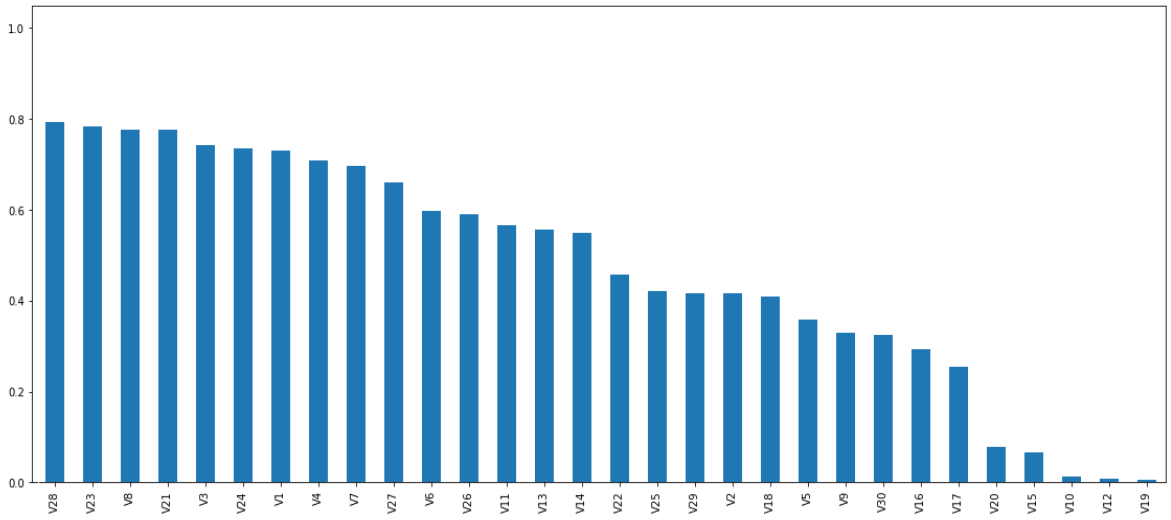


Fig. 4 Features ranking in descending order of absolute values obtained using Pearson Correlation on the original dataset.

The total number of Benign class and Malignant class samples allotted in the training set and the testing set of all the 5 folds are shown in table 2.

Table 2. Total number of benign class samples and malignant class samples allotted in the training set and testing set of each fold

Fold Number	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
Benign class samples in the training set	286	285	285	285	286
Malignant class samples in the training set	169	170	170	170	170
Benign class samples in the testing set	71	72	72	72	71
Malignant class samples in the testing set	43	42	42	42	42

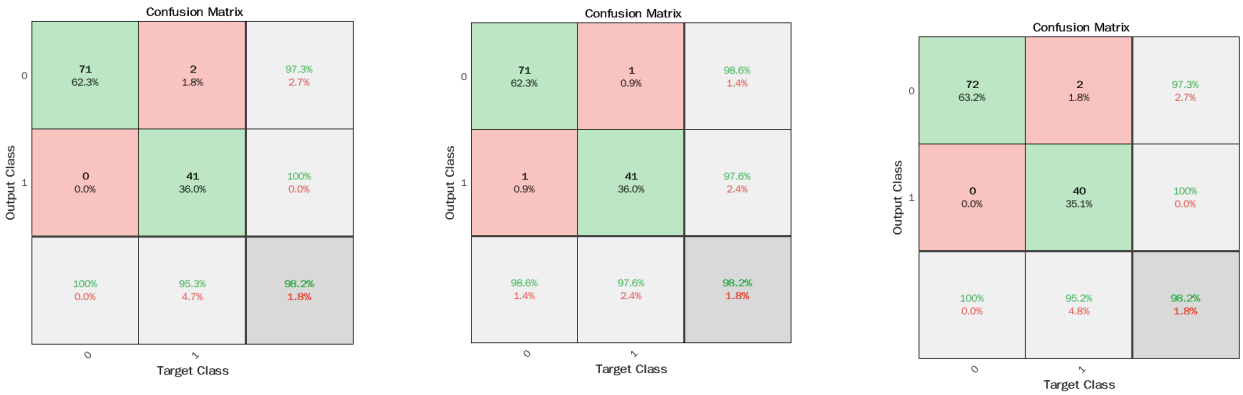
The input features V_{10}, V_{12}, V_{15}, V_{19}, and V_{20} obtained the least PC scores. After removing these 5 features from the original dataset, the remaining 25 features are given as input to the SSAE. Different SSAE models with different configurations and hyper-parameters values are tested to obtain optimal results. Each setup that is SAE1, SAE2, and SSAE with supervised training runs for 450 epochs. Some of the tested models are shown in table 3. Model 1 acquired the lowest accuracy of 62.26 % when the values of
4, 0.1, and 22 are applied as β, λ, and hidden layer nodes in SAE1, respectively. For SAE2 in Model 1, 5, 0.1, and 18 are used as β, λ, and the number of hidden layers nodes, respectively.

In the final framework, the number of hidden layer neurons, β, λ, and sparsity proportion of 17, 2, 0.001, and 0.15, respectively, are used for SAE1. Twelve nodes are used in the hidden layer, and the values of 2, 0.001, and 0.15 are used as the β, λ, and sparsity proportion, respectively, for SAE 2. Fig. 5 shows the confusion matrix generated in each fold of the 5-fold cross-validation step. The framework's performance is calculated based on the final confusion matrix, which is obtained by adding the values of the individual confusion matrix attained for each fold. Each of the folds 1, 2, 3, and 5 obtained a predictive accuracy of 98.2%. The highest predictive accuracy of 99.1% is obtained for $k = 4$. The results acquired by the proposed approach are also compared with the performance of other classification techniques applied to the original dataset, as shown in table 4. The decision tree attained 93.85%, and logistic regression attained 94.56% of accuracy. The lowest accuracy of 92.79% is achieved by Gaussian Naïve Bayes learning technique. The bagged tree, a bagging-based ensemble approach, achieved the highest precision, recall, F_1-score, and accuracy of 0.949, 0.950, 0.949, and 95.25%, respectively, among all the classification techniques. The comparative analysis of the accuracy acquired by the machine learning algorithms and the proposed framework is shown in Fig. 6. The framework attained the precision, recall, F_1-score, and accuracy of 0.985, 0.981, 0.983, and 98.42%, respectively.

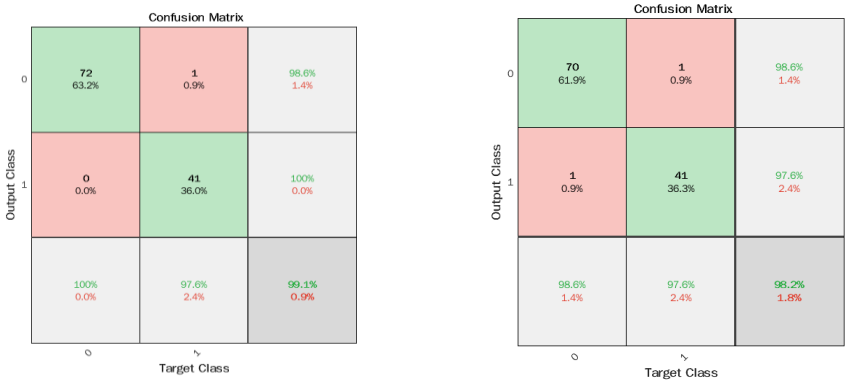
Table 5 presents the comparative analysis of the proposed framework with the work done by other authors, employing k-fold cross-validation on the WDBC dataset.

Table 3. Comparative analysis of accuracy achieved by different Stacked Sparse Autoencoder models

Model	Sparse Autoencoder (SAE)	Input layer nodes	Hidden layer nodes	Sparsity coefficient (β)	Regularization coefficient (λ)	Accuracy (%)
Model 1	SAE 1	25	22	4	0.1	62.26
	SAE 2	22	18	5	0.1	
Model 2	SAE 1	25	17	1	0.01	93.91
	SAE 2	17	14	4	0.001	
Model 3	SAE 1	25	20	3	0.001	94.70
	SAE 2	20	16	4	0.01	
Model 4	SAE 1	25	15	4	0.003	96.50
	SAE 2	15	10	2	0.002	
Final Model	SAE 1	25	17	2	0.001	98.42
	SAE 2	17	12	2	0.001	



***k* = 1** ***k* = 2** ***k* = 3**



***k* = 4** ***k* = 5**

Fig. 5 Confusion matrix obtained for each fold of 5-fold cross-validation

Table 4. Comparison of results achieved by various classification algorithms applied to the original dataset with the proposed approach

Classification Algorithm	Precision	Recall	F_1-score	Accuracy (%)
Decision Tree	.938	.930	.934	93.85
Logistic Regression	.941	.942	.941	94.56
Gaussian Naïve Bayes	.924	.922	.923	92.79
Bagged Tree	.949	.950	.949	95.25
Proposed Framework	.985	.981	.983	98.42

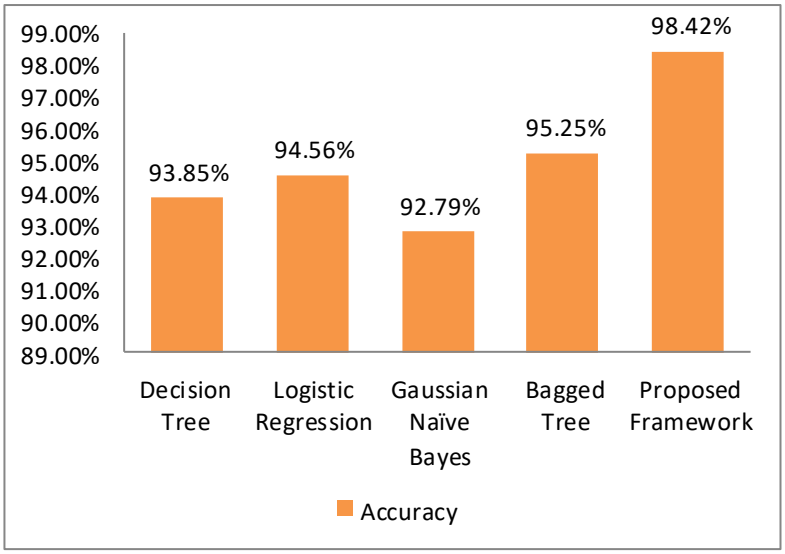


Fig. 6 Comparative analysis of the accuracy achieved

Table 5. Comparison of work done by other authors with the proposed framework

Name of the authors	Techniques used and accuracy achieved
Bennet et al. [24]	Support vector machine (SVM) for Decision tree (DT) - 97.20%
Prasad et al. [25]	Ant colony optimization and SVM - 95.96 % Particle swarm optimization and SVM - 97.37% Genetic algorithm and SVM - 97.19%
Lavanya et al. [26]	Feature selection and Ensemble DT - 95.96%
Mert et al. [27]	Probabilistic neural network with feature reduction technique - 96.31%
Peng et al. [28]	Immune system and semi-supervised learning-based approach - 98.00%
Marandi et al. [29]	Enhanced Artificial neural network - 96.19%
Darapureddy et al. [30]	Logistic regression - 88.39% DT - 90.70% SVM - 90.15%, *k* nearest neighbours (*k*NN) - 89.63%
This Study	Feature raking and stacked sparse autoencoder based approach - 98.42%

5. Conclusion

Disease prediction systems based on techniques offered by data mining help reduce the health-related risks of patients suffering from life-threatening chronic diseases. Various methods are available in this domain to understand the contribution of every input variable toward the prediction of the disease. Data mining also provides many approaches which help find the appropriate form in which the data can be presented to the prediction systems. This paper proposes a new framework that employs the PC method for feature ranking and SSAE for feature learning. 98.42% of accuracy has been archived using the proposed setup.

References

[1] Y. Kumar, A. Koul, R. Singla and M. F. Ijaz, Artificial Intelligence in Disease Diagnosis: A Systematic Literature Review, Synthesizing Framework and Future Research Agenda, Journal of Ambient Intelligence and Humanized Computing. (2022) 1-28.

[2] M. E. Hossain, A. Khan, M. A. Moni, and S. Uddin, Use of Electronic Health Data for Disease Prediction: A Comprehensive Literature Review, IEEE/ACM Transactions on Computational Biology and Bioinformatics. 18(2) (2019) 745–758.

[3] K. Verma, S. Bhardwaj, R. Arya, M. S. U. Islam, M. Bhushan, A. Kumar, and P. Samant, Latest Tools for Data Mining and Machine Learning, International Journal of Innovative Technology and Exploring Engineering. 8(9S) (2019).

[4] H. Wang, T. M. Khoshgoftaar and K. Gao, A Comparative Study of Filter-Based Feature Ranking Techniques, IEEE International Conference on Information Reuse & Integration. (2010) 43-48.

[5] G. Zhong, L. N. Wang, X. Ling, and J. Dong, An Overview on Data Representation Learning: From Traditional Feature Learning to Recent Deep Learning, Journal of Finance and Data Science. 2(4) (2016) 265-278.

[6] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, A Survey of Deep Neural Network Architectures and their Applications, Neurocomputing. 234(19) (2017) 11-26.

[7] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, Greedy Layer-Wise Training of Deep Networks, Proceedings of the 19th International Conference on Neural Information Processing Systems, (2006) 153-160.

[8] Y. Qi, C. Shen, D. Wang, J. Shi, X. Jiang, and Z. Zhu, Stacked Sparse Autoencoder-Based Deep Network for Fault Diagnosis of Rotating Machinery, IEEE Access. 7 (2017) 15066-15079.

[9] Y. F. Niu, Z. Lin, S. J. Ting, and X. Xue, L. Gang, Theories and Applications of Auto-Encoder Neural Networks: A Literature Survey, Chinese Journal of Computers. 1 (2019) 203-230.

[10] J. Yu, X. Zheng, and S. Wang, A Deep Autoencoder Feature Learning Method for Process Pattern Recognition, Journal of Process Control. 79 (2019) 1-15.

[11] D. Charte, F. Charte, S. García, M. J. Jesus and F. A Herrera, A Practical Tutorial on Autoencoders for Nonlinear Feature Fusion: Taxonomy, Models, Software and Guidelines, Information Fusion. 44 (2018) 78-96.

[12] J. Zhai, S. Zhang, J. Chen and Q, He Autoencoder and its Various Variants, IEEE International Conference on Systems, Man, and Cybernetics. (2018) 415-419.

[13] I. Zenbout, A. Bouramoul, and S. Meshoul, Stacked Sparse Autoencoder for Unsupervised Features Learning in PanCancer miRNA Cancer Classification, Conference on Innovative Trends in Computer Science. (2019).

[14] C. S. N. Pathirage, J. Li, L. Li, H. Hao, W. Liu and R. Wang, Development and Application of a Deep Learning-Based Sparse Autoencoder Framework for Structural Damage Identification, Structural Health Monitoring. 18(1) (2019) 103–122.

[15] J. Wang, W. Wang, F. Luo, and S. Wei, Modulation Classification Based on Denoising Autoencoder and Convolutional Neural Network with GNU Radio, The Journal of Engineering. 19 (2019) 6188–6191.

[16] A. Law and A. Ghosh, Multi-label Classification using a Cascade of a Stacked Autoencoder and Extreme Learning Machines, Neurocomputing. 358(17) (2019) 222-234.

[17] C. Li, D. Zhao, S. Mu, W. Zhang, N. Shi, and L. Li, Fault Diagnosis for Distillation Process Based on CNN–DAE, Chinese Journal of Chemical Engineering. 27(3) (2019) 598-604.

[18] G. Lu, X. Zhaoa, J. Yin, W. Yang and B. Li, Multi-Task Learning Using Variational Auto-Encoder for Sentiment Classification, Pattern Recognition Letters. 132 (2020) 115-122.

[19] M. Chen, X. Shi, Y. Zhang, D. Wu, and M. Guizani, Deep Feature Learning for Medical Image Analysis with Convolutional Autoencoder Neural Network, IEEE Transactions on Big Data. 7(4) (2017) 750-758.

[20] F. Farahnakian and J. Heikkonen, A Deep Auto-Encoder Based Approach for an Intrusion Detection System, 20th International Conference on Advanced Communication Technology. (2018).

[21] W. W. Y. Ng, G. Zeng, J. Zhang, D. S. Yeung, and W. Pedrycz, Dual Autoencoder Features for the Imbalance Classification Problem, Pattern Recognition. 60 (2016) 875-889.

[22] M. A. Aslam, C. Xue, Y. Chen, A. Zhang, M. Liu, K. Wang and D. Cui, Breath Analysis Based Early Gastric Cancer Classification from Deep-Stacked Sparse Autoencoder Neural Network, Scientific Reports. 11 (2021) 4014.

[23] (2013). M. Lichman, UCI Machine Learning Repository, Irvine: the University of California, School of Information and Computer Science. [Online]. Available: https://archive.ics.uci.edu/

[24] K. P. Bennett and J. A. Blue, A Support Vector Machine Approach to Decision Trees, IEEE International Joint Conference on Neural Networks. 3 (1998) 2396-2401.

[25] Y. Prasad, K. K. Biswas, and C. K. Jain, SVM Classifier Based Feature Selection Using GA, ACO and PSO for siRNA Design, Proceedings of the First International Conference on Advances in Swarm Intelligence. 2 (2010) 307–314.

[26] D. Lavanya and KU. Rani, Ensemble Decision Tree Classifier for Breast Cancer Data, International Journal of Information Technology Convergence and Services. 2(1) (2012).

[27] A. Mert, N. K l ç and A. Akan, An Improved Hybrid Feature Reduction for Increased Breast Cancer Diagnostic Performance, Biomedical Engineering Letters. 3(3) (2014) 285-291.

[28] L. Peng, W. Chen, W. Zhou, F. Li, J. Yang and J. Zhang, An Immune-inspired Semi-supervised Algorithm for Breast Cancer Diagnosis, Computer Methods and Programs in Biomedicine. 134 (2016) 259-265.

[29] R. J. Marandi, S. Davarzani, M. S. Gharibdousti and B. K. Smith, An Optimum ANN-Based Breast Cancer Diagnosis: Bridging Gaps Between ANN Learning and Decision-Making Goals, Applied Soft Computing. 72 (2018) 108-120.

[30] N. Darapureddy, N. Karatapu, and T. K. Battula, Research of Machine Learning Algorithms using K-Fold Cross Validation, International Journal of Engineering and Advanced Technology. 8(6S) (2019).