The Accuracy of KNN, SVM, SVDD, neural network and DBSCAN, k-means, NN-chain for Determining the Cancer by Gene-expression Signatures And the Arrhythmia Types

Hsieh Cheng-Han, Hsu Ting-Hao, Sun Shih-Yu, Lu Che-Yuan, Huang Chia-Yen

May 2023

Abstract

Nowadays, using machine to give an early diagnosis of a cancer type is widely studied. In this paper, based on the gene-expression dataset on Synapse.org, we compare the performance of machine learning techniques, including data preprocessing, e.g., Principal components analysis (PCA), Autoencoder (AE), classification, e.g., k-nearest neighbors (KNN), support vector machine (SVM), support vector data description (SVDD), neural network (NN) and naive Bayes classifier; and clustering for dealing with unknown types, e.g., k-means, density-based spatial clustering of applications with noise (DBSCAN).

The result of gene dataset shows that, with PCA reducing the dimension of the dataset to 32, use KNN for classifying the known types, and finally apply DBSCAN on the remaining data, the accuracy of predicting can reach about 94%, which beats the other methods out. In the other way, the result of arrhythmia dataset shows that, with raw data, use KNN and k-means, the highest accuracy of predicting can reach about 53%.

The source code can be found in https://github.com/OEmiliatanO/CSE_data_mining.

1 Introduction

Over the years, machine learning methods for early predict various disease are widely used in medical field. [1] [2] From diabetes [3], heart disease [4], to lung adenocarcinoma [5] etc., machine learning methods is universally studied attributed to the promising perspective. With a well-built database of disease and a well-choose combination of machine learning mathods, within seconds, a machine may determine the type of disease that this person may have by collecting the data from the person, and thereby achieve the goal of personalized medicine. As a result, machine learning methods have become a popular tool for medical researchers.

In real application, besides from the konwn diseases in the train data, there're numerous unkown types of diseases. When a new type of disease appears, how shall the machine results? In ideal situation, this new type of disease should be reported as an unkonwn type, which can be done by the combination of classification and clustering. However, how to choose the proper machine learning methods is a hot potato, since the different combinations of machine learning methods have different performance.

In this work, the dataset on Synapse.org is used, which contains 20531 RNA sequences and 3 types of cancer in the train data, i.e., kidney renal clear cell carcinoma (KIRC), breast invasive carcinoma (BRCA), lung adenocarcinoma (LUAD), and 2 new types in test data, i.e., colon adenocarcinoma (COAD), prostate adenocarcinoma (PRAD). And also, the dataset of arrhythmia is included, which caintains 8 types in train data, and 5 new types in test data. To find out which techniques is more suitable to fit these two datasets, this work examines the the accuracy of the combinations of some major machine learning techniques, including PCA, AE, KNN, SVM, SVDD, NN, naive Bayes classifier, NN-chain, k-means, DBSCAN.

2 Related works

Data preprocess

Autoencoder (AE): An AE is a type of neural network used for unsupervised learning and dimensionality reduction. It is designed to learn efficient representations or encodings of input data by training the network to reconstruct its own inputs. The AE consists of an encoder and a decoder. The encoder compresses the input data into a lower-dimensional representation, often called the latent space or code. The decoder then aims to reconstruct the original input data from this compressed representation. By learning a compact representation of the input data, AEs can capture important features and discard noise or irrelevant information [6]. This makes them effective in tasks such as image [7] [8] [9] or text classification [10] [11], where reducing the dimensionality of the data can improve performance.

Principal components analysis (PCA): PCA is a statistical technique used for dimensionality reduction and data exploration. It aims to transform a dataset with a large number of variables into a lower-dimensional space while preserving the most important information. PCA accomplishes this by identifying the principal components, which are linear combinations of the original variables. Applications of PCA span various fields, including image [12] [13] and signal processing [14] [15], genetics [16] [17]. It is particularly useful in scenarios where the dataset is high-dimensional and the interpretation and visualization of the data are challenging.

Standardization and Interpolation: With simple interpolation, the dataset size can be easily increased. To ensure that the new produced data would not lower the correctness and precsion of the dataset, standardization to the origin dataset is required.[**?**] has used simple interpolation to achieve data augmentation with great outcome. With standardization and interpolation, we earn enough data for the latter machine learning.

Classification algorithm

k-nearest neighbors (KNN) classification algorithm: The KNN classification algorithm is a supervised learning method which is first developed by Fix and Hodges [18]. The idea of KNN is based on the idiom, "birds of a feather flock together". By picking the k-nearest neighbors of a data point, the unkonwn class label can be determined. Lots of works [19] [20] [21] show the fact that KNN performs well for prediction of diabetes disease.

support vector machine (SVM): Given a set of training datas, where each data is labeled as a binary class, such as 0 and 1, SVM training algorithm creates a model that assigns new examples to the binay labels by making it a non-probabilistic binary linear classifier. In addition, accroding to [22] [23], SVM can also use a method called kernel trick to effectively perform non-linear classification by implicitly mapping its inputs into a high-dimensional feature space. The idea of kernel trick is transform the data into higher dimension. In ideal situation, the dataset which is not linear separable becomes linear separable. Fig 1 illurates this case.

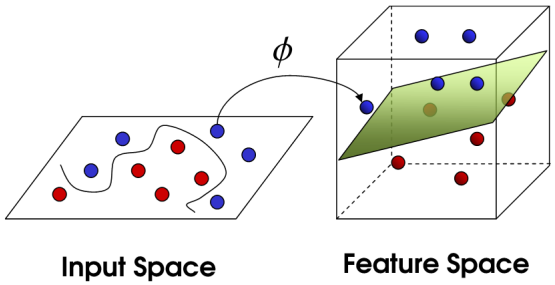

Figure 1: An example of SVM kernel trick

neural network (NN): Neural network have been used in many fields to deal with intricate datas. With input layer, hidden layer and output layer constructed by neurons, each data in dataset is processed while passing through neurons, layer by layer. After the processing, the outcome can be used to predict. Using back propogation, the accuracy of predictions increase in each training. In order to construct the hidden layer more efficient, NAS(Neural Arcitecture Searching) is used to search suitable structure for hidden layer, increasing the accuracy. According to [24][25], many neural network have been constructed and trained already, with high efficiency and accuracy in prediction of diabetes.

naive bayes classifier: naive Bayes classifier as its name suggest, is a machine learning method base on Bayesian theorem, this model will statistic each data and

find all conditional probability of each event occurring if each outcome holds. And finally when it is asked to predict the result, the model will evaluate the data which requist provide and find the most likely outcome according to the conditional probability it just recorded.

Clustering algorithm

k-means: K-means is a popular and easily implemented clustering method in machine learning and data analysis. With n data points in a dataset, k-means algorithm partition them into k distinct clusters. The algorithm works iteratively and converges to a solution by minimizing the sum of squared distance between the data points and the center of their cluster. K-means is a useful algorithm for clustering data, which is widely applied on many researches, e.g. [26], [27], [28].

Density-based spatial clustering of applications with noise (DBSCAN): DBSCAN is a data clustering algorithm proposed by Ester et al. [29] DBSCAN is particularly effective in discovering clusters of arbitrary shapes and handling noise in the data. Unlike k-means need user specify how many clusters, DBSCAN determine the clusters and noise automatically by the density of data points. This characteristic makes it particularly useful when addressing the datasets where the number of cluster is unkonwn. DBSCAN is the major clustering algorithm in machine learning field, since the ability to discover the number of clusters automatically. And the variants of DBSCAN [30] is also developed widely.

Support Vector Data Description(SVDD): SVDD is an unsupervised learning algorithm used for one-class classification or outlier detection. It defines a hypersphere in the feature space to include some specific data to classify one class from other classes. With Deep SVDD, which is the variant of the SVDD which utilize the deep learning, more precise classification can be achieved providing only one class data.[**?**] give us the precise explanation how to train a model to classify data with only one class.

Nearest neighbor chain(NN-Chain): NN-Chain is data clustering algorithm developed by Jean-Paul Benzécri and J. Juan[**?**]. This method cluster the data by merge mutual nearest neighbors nodes to one node, and repeat the same action until the goal clustering number is achieved. While the distance can be earned by the normal Euclidean distance, some distance approach is used in the algorithm as well to suit some specific data.[**?**][**?**]. NN-Chain is one of the hierarchical clustering algorithm which is likely to be presented in tree-like data structure.

3 Main Process

The pesudo code of the proposed algorithm to process the dataset is shown below and Fig 2 illurates the process of this algorithm:

Algorithm 1 The Main Algorithm

Require: Dataset

```
Data, Labels ← Dataset
Data′ ← preprocess(Data)
Konwns, Unkonwns ← classify(Data′)
Clusters ← clustering(Unkonwns)
Accuracy ← calacc((Konwns, Clusters), Labels)
```

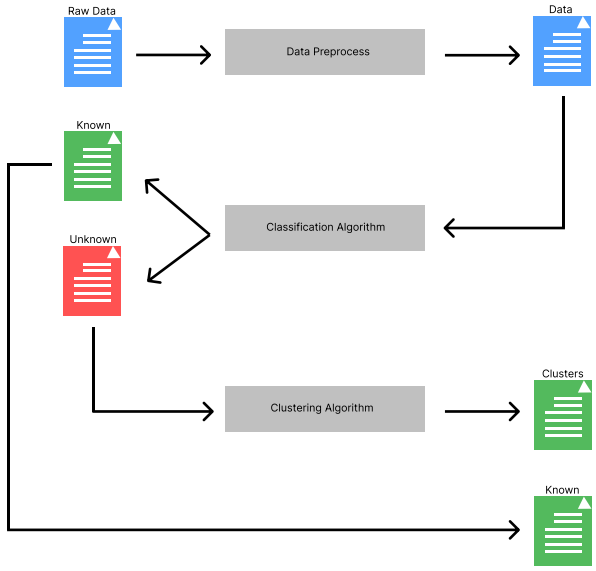

Figure 2: A figure about the main algorithm

First, preprocess the dataset, e.g. normalize, apply

PCA, apply AE. Second, classify the data after preprocessing. In the experiment, we use PCA and AE to reduce the dimension down to 32 and 30 in arrhythmia dataset and gene expression cancer RNA-Seq dataset, respectively. With this step, the konwn types and the unknown types will be separated by a certain criterion. This criterions will be varied depending on the classification algorithm. For KNN, the criterion is distance and probability. More specifically, assume a data point x, and the k-nearest data points are $p_1, ..., p_k$. Only those points whose distance between x smaller than L will be marked valid. Denote the valid data points as $p'_1, ...p'_n$, and the corresponding labels are $l_1, ..., l_n$. If this data point satisifies Eq 1, mark it as unkown type.

$$\frac{\arg\max_{t \in \text{types}\{N(t)\}}}{n} < P \tag{1}$$

where P is a manunally set probability and $N(t)$ is the number of type t in those n valid data points.

As for SVM, the criterion born in the designed. Since SVM is meant to be a binary classifier, the SVM classification algorithm must be redesigned. As shown in Algo 2.

On the other hands, neural network classifier has another criterion. First, train a neural network that classify the "konwn" and "unknown". Here "kown" means the types in train data, and "unkown" means the reverse. Then train the other neural network to classify the "konwn" data. The detail is in Algo 3. During the known-unknown classification, the SVDD is used. First define a model , input every data into it to get the outcome, then use the mean of the outcome as the center of the hypersphere. Then, train the model to make every data in the dataset approach to the center of the hypersphere as close as possible. After training, we can get the radius of the hypersphere, which can be define according to the outcome of the model. Using the radius, we can classify the data to "known", if the distance between the center and the data is less than or equal to the radius, or to "unknown" otherwise.

As for Naive Bayes classifier, first assume there are N known types. Denotes the N Bayes classifiers recognizing each type of known types as $B_1, B_2, ..., B_N$. For each test data, let B_1 determine whether it belongs to type 1, if it does, mark it as type 1 and try classifying next test data, otherwise, B_2 will determine whether it belongs to type 2 or not, and so on. If all Bayes classifiers results false, namely, the data doesn't belong to any known types, marks it as unknown type.

Algorithm 2 The SVM Multi-classification Algorithm

Require: Train Data, Labels, Test Data
Assumption: Length(Train Data) = Length(Labels) = N, Length(Test Data) = M
for t in Known Types **do**
  for i in $[1, N]$ **do**
    if Labels$[i] \neq t$ **then**
      Labels$'[i] = +1$
    else
      Labels$'[i] = -1$
    end if
  end for
  SVM$[t]$.fit(Train Data, Labels$'$)
end for
for t in Known Types **do**
  for i in $[1, M]$ **do**
    if SVM$[t]$.predict() $= +1$ **then**
      Test Labels$[i] = t$
    else
      Test Labels$[i]$ = UNKNOWN
    end if
  end for
end for

After classifying, the konwn types and unkonwn types are separated. The data marked as unkonwn types will be thrown into clustering algorithm, e.g., DBSCAN, k-means. Subsequently, the process is completed.

4 Experiment result

The experiment result of arrhythmia dataset is shown in Table 1. And the result of gene expression cancer RNA-Seq data set is in Table 2.

From the result, the fact that SVM performs worse than KNN can be seen. The main reason is because the mult-classification SVMs method designed here can't separate well in most of types. Fig 3 illurates the situation. For type 1 SVM, the negative/positive classification makes the data hard to separate, and also for type 2, 3, etc. This happens even when the kernel trick is applied, though the

Algorithm 3 Neural Network Classification

Require: Train Data, Labels, Test Data
Assumption: Length(Train Data) = Length(Labels) = N, Length(Test Data) = M
1 Binary NN.train(Train Data, Labels)
2 Mutli-Classification NN.train(Train Data, Labels)
3 Knowns, Unkonwns = Binary NN.predict(Test Data)
4 Knowns = Mutli-Classification NN.predict(Knowns)

kernel trick sometimes ease the pain.

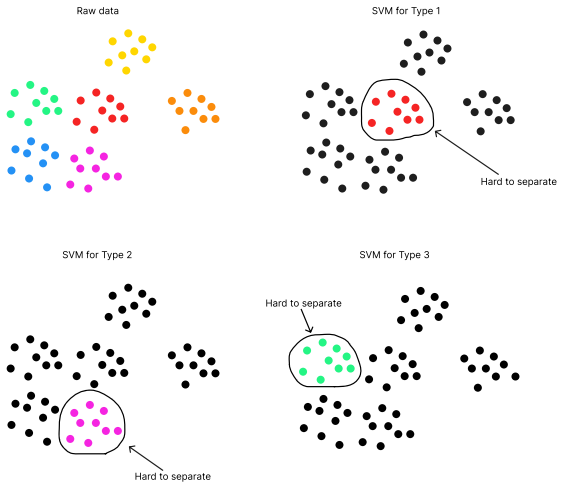

Figure 3: An example that mult-classification SVMs fail

On the other hand, KNN performs extraordinarily. In arrhythmia dataset, KNN with k-means in raw data gets 53.1646% accuracy, which is the highest accuracy comparing to others methods. And in gene expression cancer RNA-Seq dataset, KNN with k-means in PCA data gets 94.8795%. And KNN with DBSCAN performs well too. This method shows similar performance in these two datasets, which also gets 94.5783% in gene expression cancer RNA-Seq dataset PCA data.

For the SVDD + NN with NN-chain, the performance is not as good as KNN, since SVDD cannot fit well with data augmented by interpolation. The hypersphere that defined by SVDD includes all data points in the test dataset, which makes clustering impossible. Therefore, though NN classifying achieves 100% correctness on the known types, the accuracy of SVDD + NN with NN-chain can only reach 64.45% in gene expression cancer RNA-Seq dataset.

5 Conclusion

In this study, we present a framework of algorithm to deal with the dataset containing out-of-knowledge data. And with in this algorithm, multiple major machine learning techniques are used, e.g., for preprocess, PCA, AE, normalization; for classification, KNN, SVM, SVDD, NN, as for clustering, DBSCAN, k-means, NN-chain. Among the various combinations, PCA + KNN + k-means performs outstanding in gene expression cancer RNA-Seq dataset, and raw data + KNN + k-means performs the best in arrhythmia dataset. Accidently, SVM performs bad in these two dataset. The main cause is the defect of the multi-classification SVMs design, which make some SVMs hard to separate the target types. However, by customize the kernel for each SVM, the problem may ease. In the same time, the increasing number of parameters can be fine-tuned by metaheuristic algorithms, e.g., simulated annealing, genetic algorithm, etc. We will develope a refinement design on multi-classification SVMs, to enhance the accuracy on these datasets. Another issue is SVDD, in 32 dimension PCA data, it can not separate the known and unkonwn types well. The reason behind is that in a lower dimension, the distance between known and unknown types are so close that SVDD can not separate them. If reduce the dataset to a relatively larger dimension, the problem may be solved.

6 Contribution

- Hsieh Cheng-Han: KNN, SVM, DBSCAN, paper, ppt
- Hsu Ting-Hao: NN, SVDD, NN-Chain, paper, ppt
- Sun Shih-Yu: SVM, k-means, paper
- Lu Che-Yuan: naive Bayes classifier
- Huang Chia-Yen: naive Bayes classifier, paper, poster, ppt

Table 1: COMPARISON OF CLASSIFICATION TECHNIQUES. (ARRHYTHMIA DATASET)

Method	Raw		Normalized		PCA-32		PCA-32 Normalized		AE-32		AE-32 Normalized	
	Search (s)	acc	Search (s)	acc	Search (s)	acc	Search (s)	acc	Search (s)	acc	Search (s)	acc
KNN-Brute-Force + DBSCAN	0.0285	47.4684 ± 0	0.0254	46.8354 ± 0	0.0070	38.6076 ± 0	0.0071	42.4051 ± 0	0.0071	$\mathbf{37.9747 \pm 0}$	0.0071	$\mathbf{44.3038 \pm 0}$
SVM (linear) + DBSCAN	1.7269	25.9494 ± 0	8.9285	35.4430 ± 0	0.2186	32.2785 ± 0	0.8162	29.7468 ± 0	0.2169	31.6456 ± 0	0.1552	27.8481 ± 0
SVM (polynomial) + DBSCAN	0.9993	25.9494 ± 0	1.6100	25.3165 ± 0	0.1997	1.2658 ± 0	0.4577	33.5443 ± 0	0.2402	30.3797 ± 0	0.4924	30.2785 ± 0
SVM (RBF) + DBSCAN	0.9915	25.9494 ± 0	1.4681	22.7848 ± 0	1.4328	0.6329 ± 0	11.9417	30.3797 ± 0	13.5147	22.1519 ± 0	24.7346	34.1772 ± 0
SVM (sigmoid) + DBSCAN	3.5272	28.4810 ± 0	6.4731	20.2532 ± 0	1.5779	30.3797 ± 0	0.9797	36.0759 ± 0	0.5284	22.1519 ± 0	0.8430	24.6835 ± 0
KNN-Brute-Force + kmeans	0.0236	$\mathbf{53.1646 \pm 0}$	0.0343	$\mathbf{50.6329 \pm 0}$	0.0068	$\mathbf{43.0380 \pm 0}$	0.0069	$\mathbf{43.0380 \pm 0}$	0.0069	36.0759 ± 0	0.0080	41.7722 ± 0
SVM (linear) + kmeans	1.7719	25.3165 ± 0	37.9747	37.9747 ± 0	0.2317	25.3165 ± 0	0.8932	27.8481 ± 0	0.2331	24.0506 ± 0	0.1596	20.2532 ± 0
SVM (polynomial) + kmeans	1.0456	25.3165 ± 0	1.4942	26.5823 ± 0	0.2326	14.5570 ± 0	0.4094	28.4810 ± 0	0.2033	24.0506 ± 0	0.4884	25.3165 ± 0
SVM (RBF) + kmeans	34.5382	8.8608 ± 0	43.0936	29.7468 ± 0	110.589	0.6329 ± 0	112.945	1.2658 ± 0	38.7497	22.7848 ± 0	14.1561	27.2152 ± 0
SVM (sigmoid) + kmeans	4.0277	22.6582 ± 0	8.0918	17.7848 ± 0	0.7529	25.3165 ± 0	0.7967	12.0253 ± 0	0.6921	17.7215 ± 0	0.4013	28.4810 ± 0
Bayes-GA + kmeans	0.1342	13.2911 ± 0	X	X	0.0294	14.5570 ± 0	0.0400	31.6456 ± 0	0.0297	15.8228 ± 0	0.0399	31.6456 ± 0
NN + NN-Chain	0.6338	31.6456 ± 0	0.633	29.7468 ± 0	0.9358	31.6456 ± 0	0.8370	30.3797 ± 0	1.0540	29.7468 ± 0	1.0540	29.7468 ± 0

Table 2: COMPARISON OF CLASSIFICATION TECHNIQUES. (GENE EXPRESSION CANCER RNA-SEQ DATASET)

Method	Raw		Normalized		PCA-30		PCA-30 Normalized		AE-30		AE-30 Normalized	
	Search (s)	acc	Search (s)	acc	Search (s)	acc	Search (s)	acc	Search (s)	acc	Search (s)	acc
KNN-Brute-Force + DBSCAN	14.5101	59.0361 ± 0	10.3013	$\mathbf{66.5663 \pm 0}$	0.0224	94.5783 ± 0	0.0237	78.3133 ± 0	0.0258	$\mathbf{64.4578 \pm 0}$	0.0235	59.3373 ± 0
SVM (linear) + DBSCAN	124.593	61.7470 ± 0	610.6730	9.9398 ± 0	0.6466	9.9398 ± 0	0.1193	10.5422 ± 0	0.1407	63.2530 ± 0	0.1122	13.5542 ± 0
SVM (polynomial) + DBSCAN	81.6495	61.7470 ± 0	86.6406	1.2048 ± 0	0.2585	64.4578 ± 0	0.3012	61.1446 ± 0	0.2013	63.2530 ± 0	0.2056	50.3012 ± 0
SVM (RBF) + DBSCAN	X	not converge	X	not converge	12.5077	9.9398 ± 0	3.8838	63.2530 ± 0	5.0461	40.3614 ± 0	11.4194	49.6988 ± 0
SVM (sigmoid) + DBSCAN	X	not converge	X	not converge	0.6372	71.6867 ± 0	0.8222	65.9639 ± 0	0.6236	44.2771 ± 0	0.2273	38.2530 ± 0
KNN-Brute-Force + kmeans	22.754	$\mathbf{74.0964 \pm 0}$	6.36577	53.6145 ± 0	0.0212	$\mathbf{94.8795 \pm 0}$	0.0214	$\mathbf{84.9398 \pm 0}$	0.0214	42.1687 ± 0	0.0220	$\mathbf{65.9639 \pm 0}$
SVM (linear) + kmeans	136.408	61.7470 ± 0	664.2860	9.9398 ± 0	0.5914	9.9398 ± 0	0.1077	10.5422 ± 0	0.1274	63.2530 ± 0	0.1084	13.2530 ± 0
SVM (polynomial) + kmeans	81.4276	61.7470 ± 0	81.1669	63.8554 ± 0	0.2003	64.4578 ± 0	0.2167	25.3012 ± 0	0.1780	63.2530 ± 0	0.1930	50.3012 ± 0
SVM (RBF) + kmeans	1772.32	9.9398 ± 0	1800.6	9.9398 ± 0	1.3625	9.9398 ± 0	1.3848	9.9398 ± 0	11.1729	39.7590 ± 0	4.7882	28.6145 ± 0
SVM (sigmoid) + kmeans	190.866	26.5060 ± 0	81.7487	63.8554 ± 0	0.8287	65.9639 ± 0	0.4269	65.9639 ± 0	0.9908	28.9157 ± 0	0.2360	39.4578 ± 0
Bayes-GA + kmeans	2.231	43.6746 ± 0	X	X	0.0327	34.0361 ± 0	0.0364	16.8675 ± 0	0.0338	27.7946 ± 0	0.0427	16.8674 ± 0
NN + NN-Chain	163.359	40.9600 ± 0	169.031	41.2651 ± 0	1.1779	35.5421 ± 0	1.1170	40.96 ± 0	0.9979	16.4634 ± 0	1.0220	41.2651 ± 0

References

[1] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis, "Machine learning applications in cancer prognosis and prediction," *Computational and structural biotechnology journal*, vol. 13, pp. 8–17, 2015.

[2] J. A. Cruz and D. S. Wishart, "Applications of machine learning in cancer prediction and prognosis," *Cancer informatics*, vol. 2, p. 117693510600200030, 2006.

[3] I. Kavakiotis, O. Tsave, A. Salifoglou, N. Maglaveras, I. Vlahavas, and I. Chouvarda, "Machine learning and data mining methods in diabetes research," *Computational and structural biotechnology journal*, vol. 15, pp. 104–116, 2017.

[4] M. Learning, "Heart disease diagnosis and prediction using machine learning and data mining techniques: a review," *Adv. Comput. Sci. Technol*, vol. 10, no. 7, pp. 2137–2159, 2017.

[5] L. Huang, L. Wang, X. Hu, S. Chen, Y. Tao, H. Su, J. Yang, W. Xu, V. Vedarethinam, S. Wu *et al.*, "Machine learning of serum metabolic patterns encodes early-stage lung adenocarcinoma," *Nature communications*, vol. 11, no. 1, p. 3556, 2020.

[6] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder." in *Interspeech*, vol. 2013, 2013, pp. 436–440.

[7] Y. Zhang, "A better autoencoder for image: Convolutional autoencoder," in *ICONIP17-DCEC. Available online: http://users. cecs. anu. edu. au/Tom. Gedeon/conf/ABCs2018/paper/ABCs2018_paper_58. pdf (accessed on 23 March 2017)*, 2018.

[8] K. G. Lore, A. Akintayo, and S. Sarkar, "Llnet: A deep autoencoder approach to natural low-light image enhancement," *Pattern Recognition*, vol. 61, pp. 650–662, 2017.

[9] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Deep convolutional autoencoder-based lossy image compression," in *2018 Picture Coding Symposium (PCS)*. IEEE, 2018, pp. 253–257.

[10] W. Xu, H. Sun, C. Deng, and Y. Tan, "Variational autoencoder for semi-supervised text classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[11] W. Xu and Y. Tan, "Semisupervised text classification by variational autoencoder," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 1, pp. 295–308, 2019.

[12] J. Ma and Y. Yuan, "Dimension reduction of image deep feature using pca," *Journal of Visual Communication and Image Representation*, vol. 63, p. 102578, 2019.

[13] J. Yang and J. yu Yang, "From image vector to matrix: a straightforward image projection technique—impca vs. pca," *Pattern Recognition*, vol. 35, no. 9, pp. 1997–1999, 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320302000407

[14] R. P. Desale and S. V. Verma, "Study and analysis of pca, dct and dwt based image fusion techniques," in *2013 International Conference on Signal Processing , Image Processing and Pattern Recognition*, 2013, pp. 66–69.

[15] F. Castells, P. Laguna, L. Sornmo, A. Bollmann, and J. M. Roig, "Principal component analysis in ecg signal processing," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. 1–21, 2007.

[16] D. Reich, A. L. Price, and N. Patterson, "Principal component analysis of genetic data," *Nature genetics*, vol. 40, no. 5, pp. 491–492, 2008.

[17] J. Novembre and M. Stephens, "Interpreting principal component analyses of spatial population genetic variation," *Nature genetics*, vol. 40, no. 5, pp. 646–649, 2008.

[18] E. Fix and J. L. Hodges, "Discriminatory analysis. nonparametric discrimination: Consistency properties," *International Statistical Review / Revue Internationale de Statistique*, vol. 57, no. 3, pp. 238–247, 1989. [Online]. Available: http://www.jstor.org/stable/1403797

[19] M. NirmalaDevi, S. A. alias Balamurugan, and U. V. Swathi, "An amalgam knn to predict diabetes mellitus," in *2013 IEEE International Conference ON Emerging Trends in Computing, Communication and Nanotechnology (ICECCN)*, 2013, pp. 691–695.

[20] D. Shetty, K. Rit, S. Shaikh, and N. Patil, "Diabetes disease prediction using data mining," in *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, 2017, pp. 1–5.

[21] V. Vijayan and A. Ravikumar, "Study of data mining algorithms for prediction and diagnosis of diabetes mellitus," *International journal of computer applications*, vol. 95, no. 17, 2014.

[22] S.-i. Amari and S. Wu, "Improving support vector machine classifiers by modifying kernel functions," *Neural Networks*, vol. 12, no. 6, pp. 783–789, 1999.

[23] M. Hofmann, "Support vector machines-kernels and the kernel trick," *Notes*, vol. 26, no. 3, pp. 1–16, 2006.

[24] T. R. Gadekallu, N. Khare, S. Bhattacharya, S. Singh, P. K. R. Maddikunta, and G. Srivastava, "Deep neural networks to predict diabetic retinopathy," *Journal of Ambient Intelligence and Humanized Computing*, Apr 2020. [Online]. Available: https://doi.org/10.1007/s12652-020-01963-7

[25] T. Beghriche, M. Djerioui, Y. Brik, B. Attallah, and S. B. Belhaouari, "An efficient prediction system for diabetes disease based on deep neural network," *Complexity*, vol. 2021, p. 6053824, Dec 2021. [Online]. Available: https://doi.org/10.1155/2021/6053824

[26] O. J. Oyelade, O. O. Oladipupo, and I. C. Obagbuwa, "Application of k means clustering algorithm for prediction of students academic performance," 2010.

[27] N. Nidheesh, K. Abdul Nazeer, and P. Ameer, "An enhanced deterministic k-means clustering algorithm for cancer subtype prediction from gene expression data," *Computers in Biology and Medicine*, vol. 91, pp. 213–221, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010482517303402

[28] M. S. Kadhm, I. W. Ghindawi, and D. E. Mhawi, "An accurate diabetes prediction system based on k-means clustering and proposed classification approach," *International Journal of Applied Engineering Research*, vol. 13, no. 6, pp. 4038–4041, 2018.

[29] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96. AAAI Press, 1996, p. 226–231.

[30] K. Khan, S. U. Rehman, K. Aziz, S. Fong, and S. Sarasvady, "Dbscan: Past, present and future," in *The Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014)*, 2014, pp. 232–238.