Threshold Feature Selection PCA

Felipe de Melo Battisti, Tiago Buarque Assunção de Carvalho

Universidade Federal Do Agreste de Pernambuco (UFAPE), Brazil
felipebattist@gmail.com, tiago.buarque@ufape.edu.br

Abstract. Classification algorithms encounter learning difficulties when data has non-discriminant features. Dimensionality reduction techniques such as PCA are commonly applied. However, PCA has the disadvantage of being an unsupervised method, ignoring relevant class information on data. Therefore, this paper proposes the Threshold Feature Selector (TFS), a new supervised dimensionality reduction method that employs class thresholds to select more relevant features. We also present the Threshold PCA (TPCA), a combination of our supervised technique with standard PCA. During experiments, TFS achieved higher accuracy in 90% of the datasets compared with the original data. The second proposed technique, TPCA, outperformed the standard PCA in accuracy gain in 70% of the datasets.

Categories and Subject Descriptors: I.2.6 [**Artificial Intelligence**]: Learning; I.5.1 [**Pattern Recognition**]: Models; H.2.8 [**Database Management**]: Database Applications

Keywords: dimensionality reduction, machine learning, principal component analysis, feature selection, classification problems

1. INTRODUCTION

The advances of society in technology and the context of large-scale internet creates a reality where data sets with high dimensionality (data with a large number of features) are standard in various fields of technological knowledge [Eustáquio and Nogueira 2020]. Notably, such high-dimensional datasets contain an expressive number of noisy data and redundant features, which add little to the direct description of the data and can harm the execution of classification algorithms. Hence, there is a need to eliminate irrelevant data using dimensionality reduction algorithms. Such reduction not only can increase the final accuracy of several predictive models but also helps to reduce the training time of the machine learning algorithms. [Ganjei and Boostani 2022].

There are two standard ways for dimensionality reduction in the state of the art, feature extraction and feature selection. The **feature extraction** task uses techniques capable of mapping the original data int a smaller dimensional subspace, thus obtaining a new representation of the data with lower dimensionality [Woo and Lee 2018] [Huang et al. 2021] [Biagetti et al. 2021] [Mi et al. 2021] [Priyanka and Kumar 2020]. In contrast, **feature selection** aims excluding redundant features and thus generating an accurate representation of the data [Arun Kumar et al. 2022] [Gárate-Escamila et al. 2020] [Beiranvand et al. 2022] [Tang et al. 2018]. When comparing the result of the two forms of dimensionality reduction, the feature selection has the advantage of keeping the orginal meaning of the data – making the interpretation of prediction models simpler [Wang et al. 2022] . Therefore, feature selection occupies a prominent position regarding high-dimensional data reduction and has been the topic of important studies in the field of artificial intelligence [Zhou et al. 2022].

Principal component analysis (PCA) [Jolliffe 2022] is one of the most used techniques for feature extraction and has applications in the most diverse fields of data science [Zhu et al. 2022], machine learning [Weiwei 2022] [Liang et al. 2022], and deep learning [Vinodhini and Chandrasekaran 2014] [Ali

and Erçelebi 2021] [Song et al. 2022] [Liu and Durlofsky 2021]. The PCA method defines an orthogonal linear transformation that converts the data into a new coordinate system. The first coordinate after projection (first principal component) has the most significant variance, and the second principal component has the second largest variance, and so on. [Maćkiewicz and Ratajczak 1993]. However, PCA is a unsupervised method, meaning that PCA, when applied to classification problems, does not consider class information in the dataset.

The Minimum Classification Error PCA (MCEPCA) [de Carvalho et al. 2017] is a supervised technique based on the standard PCA. MCEPCA proposes a new way of selecting features after the data projection in the new subspace. Instead of simply selecting the components with the highest variance, MCEPCA selects the features that minimize the Bayes error rate for classification [Tumer and Ghosh 1996]. The main limitation of MCEPCA is that it is defined only for datasets with binary output.

This article presents the Threshold Feature Selection method (TFS) based on the concept of class thresholds. We also propose the Threshold PCA (TPCA), analogous to MCEPCA; this method keeps PCA data projection and selects features using TFS. The strength of the methods lies in using the class information to make a selection of features. The rest of this article is organized as follows: Section 2 contains background information, Section 3 presents the proposed methods, Section 4 has the experiments and results, and Section 5 presents the conclusion.

2. BACKGROUND

2.1 Principal Component Analysis (PCA)

PCA is a dimensionality reduction technique capable of extracting new features from linear combinations of the original features [Jolliffe 2002]. Assuming the dataset is represented as a matrix, PCA works as follows. The dataset has the format of a $D_{n\times f}$ matrix where n is the number of rows representing points in the data, and f is the number of columns representing the attributes of the problem. Also, any point belonging to the D matrix has the representation, $\mathbf{d}_i = [d_{i1} \dots d_{if}]$, and the vector $\bar{\mathbf{d}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{d}_i$ is the average of all points contained in D. In this way, the centered matrix $M_{n\times f}$ is calculated by subtracting $\bar{\mathbf{d}}$ from all points of the original matrix.

$$M = \begin{bmatrix} \mathbf{d}_1 - \bar{\mathbf{d}} \\ \mathbf{d}_2 - \bar{\mathbf{d}} \\ \mathbf{d}_3 - \bar{\mathbf{d}} \\ \dots \\ \mathbf{d}_n - \bar{\mathbf{d}} \end{bmatrix}$$

Then it is possible to find the covariance matrix of the data as $C = n^{-1}M^T M$. Next, we build the new dataset by following the steps:

(1) Extract the eigenvalues and eigenvectors from the covariance matrix $C_{f\times f}$.
(2) Sort the eigenvectors, so those with the highest eigenvalues get priority to maximize the variance.
(3) Finally, design the $W_{n\times k}$ matrix of new data. $W = ME$, where $E_{f\times k}$ are the selecteded eigenvectors of C and k is the number of features you want to keep in the data set, ensuring that $k < f$.

2.2 Minimum Classification Error PCA (MCEPCA)

MCEPCA is a supervised implementation of the PCA. It extracts features similar to the PCA but selects the features by minimizing the error rate of Bayes for classification [de Carvalho et al. 2017].

The method is limited to problems with only two classes. The algorithm uses a score for sorting eingenvectors instead of the eigenvalues in the original PCA. The score calculation works as follows:

(1) For each feature, one average per class must be calculated. The mean value of the i-th feature for the class $c \in \{1, 2\}$ is given by:
$$\bar{w}_{ci} = \frac{\sum_{j=1}^{n} w_{ij}\delta_{jc}}{\sum_{j=1}^{n} \delta_{jc}} ,$$
where w_{ij} is the value of the i-th feature for the j-th point in W, and δ_{jc} is the Dirac delta function $\delta_{jc} = 1$ if point j belongs to class c, and $\delta_{jc} = 0$ otherwise.

(2) The score is then calculated for each feature extracted with the PCA. s_i is the score for the i-th feature:
$$s_i = \begin{cases} \frac{(\bar{w}_{1i} - \bar{w}_{2i})^2}{\lambda_i} & , if \lambda_i \neq 0 \\ 0 & , if \lambda_i = 0 \end{cases} ,$$
where λ_i is the eigenvalue of the i-th feature.

Once the score is calculated for all features, the algorithm selects k features with the highest score and projects the data with the k eigenvectors related to the chosen feature.

3. CLASS THRESHOLD FEATURE SELECTION METHODS

In this section we propose two methods: a new feature selector based on class thresholds (TFS) and a combination of this technique with the already known PCA (TPCA). The TPCA is inspired by how MCEPCA changes PCA feature selection.

3.1 Threshold Feature Selector (TFS)

The objective of the selection technique is to generate a score to order the features of a dataset by relevance to the classification problem. In this case, we define this score by the number of class thresholds within a feature. Thus, features presenting fewer class thresholds tend to have more relevance for the classification problem. Extracting class thresholds from a dataset is a supervised task and require a set of output classes Y for each point in the dataset. This extraction works as follows. Given a dataset $D_{n \times f}$, for each point $\mathbf{d}_i$ there is a output value y_i:

$$D = \begin{bmatrix} d_{11} & d_{12} & d_{13} & ... & d_{1f} \\ d_{21} & d_{22} & d_{23} & ... & d_{2f} \\ d_{31} & d_{32} & d_{33} & ... & d_{3f} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ d_{n1} & d_{n2} & d_{n3} & ... & d_{nf} \end{bmatrix} , Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} .$$

Each column represents a feature. In this way, each feature is a vector. Next, each feature vector is sorted in ascending order so that it is possible to observe intervals where every element within the interval belongs to the same class. Finally, the upper and lower boundaries of the interval are stored and defined as **class thresholds** for the current feature.

Illustrating this process, we have a matrix of arbitrary data and a vector with three output classes as examples. To find class thresholds for features $\mathbf{x_1}$, $\mathbf{x_2}$ and $\mathbf{x_3}$, first is necessary to analyze them individually. Each feature passes through sorting in ascending order, and the output vector Y follows this ordering for each feature separately to analyze class intervals. In the case of the example above, the data threshold analyses occur as described in Fig. 1.

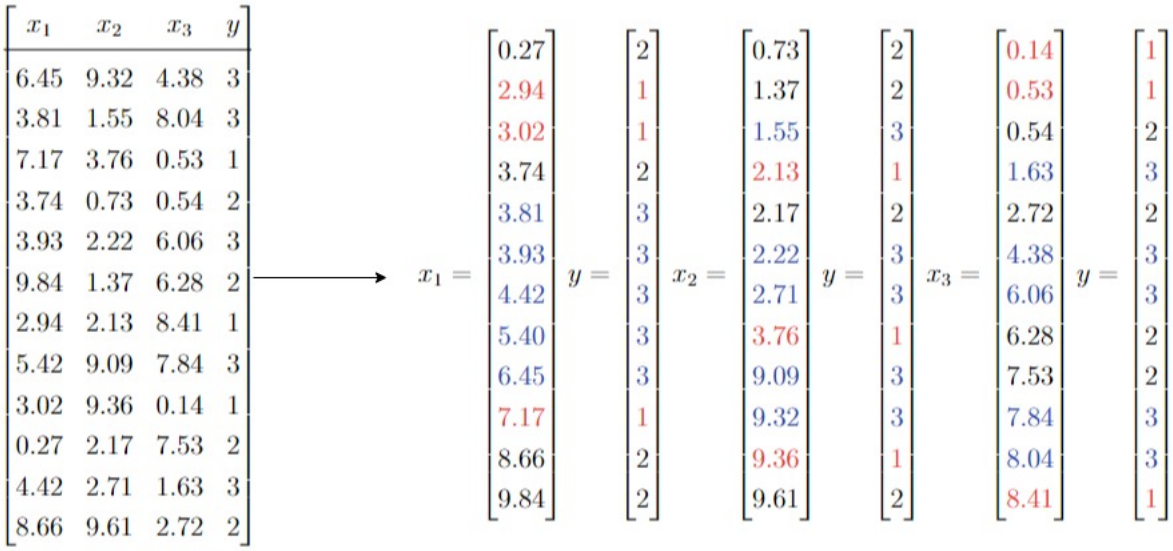

Fig. 1. Class-Thresholds extraction example

After analyzing the scope of the features, the class thresholds for each feature become observable. Interval exchange values assume the role of class thresholds for the attribute, and the number of thresholds works as a metric to assess the relevance of the characteristic. In the example above $\mathbf{x_1}$ has **5** class thresholds, while $\mathbf{x_2}$ has **8** and $\mathbf{x_3}$ presents **7** thresholds. This examination indicates that the attribute $\mathbf{x_1}$, having fewer thresholds, has greater relevance in class decision. The TFS is summarized in the following algorithm:

Algorithm

(1) Sort each feature of the dataset along with the class outputs.
(2) Check the class intervals of each feature.
(3) Collect the number of class thresholds for each feature.
(4) Select only the desired k features with the lowest number of class thresholds.

3.2 Threshold PCA (TPCA)

The **TPCA** is a technique derived from the MCEPCA, which maintains the same data extraction as PCA but after data projection uses the number of class thresholds as a metric for feature selection. In this way, the method receives a set of data and performs the dimensionality reduction seeking to keep only the features with fewer class thresholds, as described in Fig. 2 and the following algorithm.

Fig. 2. Threshold PCA algorithm data flow

Algorithm

(1) Apply the PCA algorithm to the original data without reducing its dimensionality. In order to create a new representation of the data.
(2) Apply the TFS on the extracted data.

4. EXPERIMENTS AND RESULTS

This section aims to demonstrate the effectiveness of the proposed feature selection method. The first experiment evaluates the TFS and the second compares TPCA with standard PCA. The experiment uses real datasets. Table I displays information about all of the datasets utilized in the experiments. Each dataset is stratified split in train and test sets (50%/50%). The accuracy is an average of 30 holdout repetitions for each classifier. The following classifiers were used during the tests: *(a)* k-NN (1-NN), *(b)* Naive Bayes (NB), *(c)* Logistic Regression (LR), *(d)* Decision Tree (DT).

4.1 Experiment 1: Maximum mean accuracy with Threshold Feature Selection

In this experiment, the goal is to evaluate the TFS accuracy gain when applied to dimensionality reduction by following the steps: *(1)* The algorithm is applied to a dataset reducing its dimensions to maintain only one feature. Then we store the accuracy obtained with this reduction for all classifiers. *(2)* The exact process happens, reducing the dataset to keep only two features. *(3)* This process gets repeated, increasing the number of features until the entire dataset is utilized for prediction. The accuracy for each number of selected features is stored, as described in Fig. 3.

Results. The analysis of the figure shows that, for the Wine dataset with the Decision Tree classifier, the TFS did not achieve significantly higher accuracy than the complete data classification. Still, it was able to diminish the number of features needed to get the same level of accuracy. Furthermore, in the Dermatology dataset with the 1-NN classifier, TFS increased accuracy and used fewer features, expanding the classification performance.

Table II displays all the results for the first experiment. The table contains the maximum accuracy achieved using the TFS algorithm for each dataset, the values between braces stand for the number of data features employed for achieving such accuracies. And for the comparison metric, the accuracy without any dimensionality reduction is also displayed (ALL). Each line represents a dataset, and the columns represent the mean accuracy for a classifier with or without the TFS algorithm. For these results, the evaluation of the TFS data reduction consists of verifying if the data reduction can achieve higher accuracy or similar results with fewer features.

The TFS dimensionality reduction achieved higher accuracy in seven of the ten datasets for the four classifiers. The other three remaining datasets contained different results. TFS had higher accuracy

Dataset	Points	Features	Nº of Classes
Banknote	1372	4	2
Bupa	345	7	2
Climate	540	18	2
Column	310	6	2
Dermatology	366	34	2
HillValley	606	100	2
Immunotherapy	90	7	2
Leaf	340	15	36
Wine	178	13	3
Wine Quality	6497	11	10

Table I. Datasets used during the experiments.

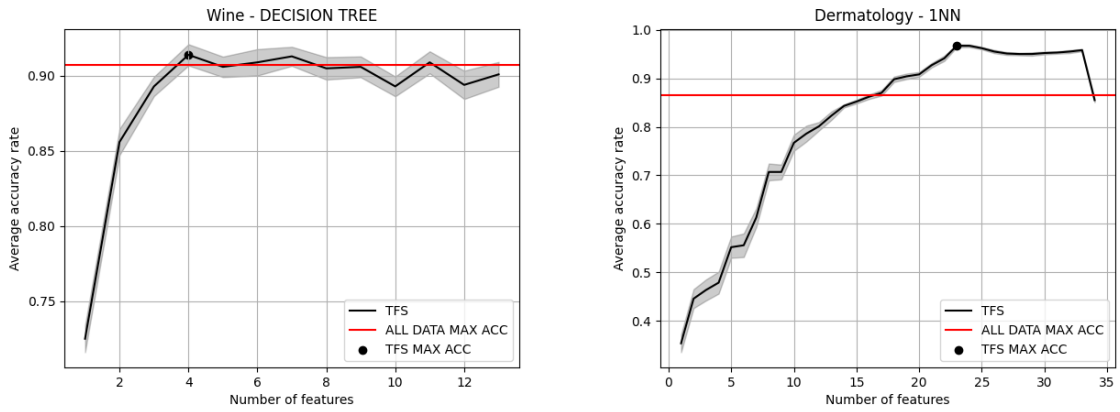

Fig. 3. Mean accuracy (y-axis) per number of features (x-axis). TFS mean accuracy for each feature for Wine (Decision Tree) and Dermatology (1-NN) datasets. The red line represents the accuracy using all the features. The black bullet represents the maximum mean accuracy with TFS.

Classifier	**LR**		**DT**		**NB**		**1-NN**	
DATA	**TFS**	**ALL**	**TFS**	**ALL**	**TFS**	**ALL**	**TFS**	**ALL**
Banknote	**0.992[3]**	0.989[4]	**0.982[3]**	0.979[4]	**0.873[2]**	0.838[4]	**1.000[3]**	0.999[4]
Bupa	0.670[5]	**0.676[6]**	0.605[5]	**0.622[6]**	**0.578[1]**	0.552[6]	**0.600[5]**	0.597[6]
Climate	0.922[17]	**0.923[18]**	**0.901[12]**	0.893[18]	**0.940[14]**	0.938[18]	**0.898[7]**	0.889[18]
Column	0.845[5]	**0.854[6]**	0.789[5]	**0.794[6]**	**0.786[1]**	0.783[6]	0.810[5]	**0.817[6]**
Dermatology	**0.977[23]**	0.969[34]	**0.952[23]**	0.944[34]	**0.885[20]**	0.891[34]	**0.967[23]**	0.866[34]
HillValley	**0.993[83]**	0.992[100]	**0.546[54]**	0.536[100]	**0.503[47]**	0.497[100]	**0.555[98]**	0.539[100]
Immunotherapy	**0.797[4]**	0.779[7]	**0.790[1]**	0.773[7]	**0.785[1]**	0.739[7]	**0.763[4]**	0.636[7]
Leaf	**0.486[14]**	0.377[15]	**0.578[14]**	0.572[15]	**0.681[12]**	0.673[15]	**0.588[10]**	0.132[15]
Wine	**0.955[6]**	0.954[13]	**0.914[4]**	0.907[13]	**0.969[11]**	0.970[13]	**0.782[2]**	0.711[13]
WineQuality	***0.591[10]***	0.591[11]	**0.571[10]**	0.570[11]	**0.544[5]**	0.530[11]	**0.553[7]**	0.536[11]

Table II. Maximum accuracy with Threshold Feature Selector against compared with full data accuracy.

for two of the four classifiers for the Bupa dataset. For the Climate dataset, TFS also performed better in three of the classifiers. For the Column dataset, TFS had higher accuracy for the Naive Bayes classifier reducing the number of features from six to a single feature. TFS achieved overall better classification results, the technique presented higher mean accuracy in three of four classifiers for eight of the ten datasets (80% of the datasets).

4.2 Experiment 2: Comparison between maximum mean accuracy achieved by PCA and TPCA

This experiment compares the mean classification accuracy achieved using the Threshold PCA over the standard PCA. For all datasets and classifiers, the experiment had the same setup.

Results. Fig. 4 shows the mean accuracy per number of features for TPCA and PCA using two datasets. For the Banknote dataset, selecting a single feature, the mean accuracy of TPCA and PCA are, respectively, 0.85 and 0.67. For the Dermatology dataset, selecting a single feature, the mean accuracy of TPCA and PCA are 0.71 and 0.42. The TPCA presented substantially higher mean accuracy for selecting a single feature. When selecting two features (for both datasets), the TPCA presented a mean accuracy close to the maximum and much higher than PCA.

Table III shows each dataset maximum mean accuracies (selecting up to 50% of the features) using

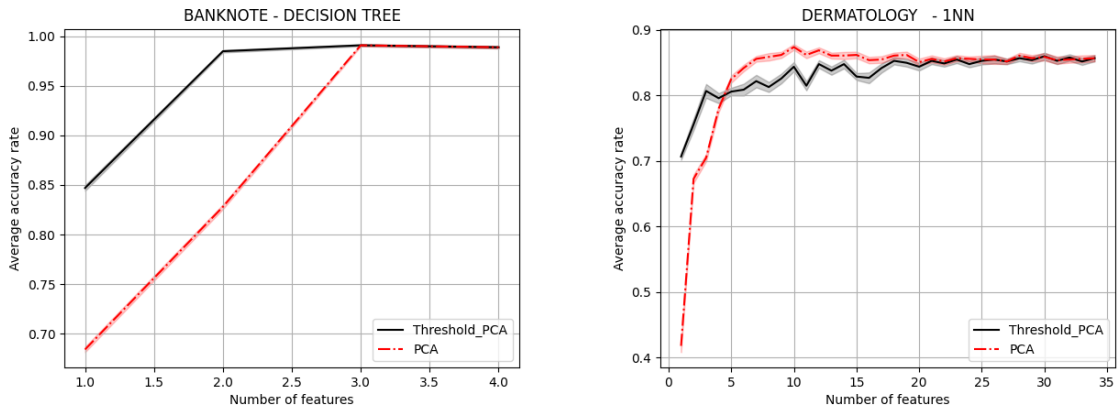

Fig. 4. Mean accuracy (y-axis) per number of features (x-axis) for TPCA and PCA for Banknote (Decision Tree) and Dermatology (1-NN) datasets.

CLASSIFIER	**LR**		**DT**		**NB**		**1-NN**	
DATA	**TPCA**	**PCA**	**TPCA**	**PCA**	**TPCA**	**PCA**	**TPCA**	**PCA**
Banknote	**0.913[2]**	0.731[2]	**0.983[2]**	0.822[2]	**0.922[2]**	0.722[2]	**0.989[2]**	0.852[2]
Bupa	**0.628[3]**	0.598[3]	**0.585[3]**	0.585[3]	**0.601[2]**	0.562[3]	0.574[3]	**0.586[3]**
Climate	**0.917[9]**	0.916[8]	0.871[3]	**0.884[9]**	**0.918[9]**	0.915[4]	**0.881[9]**	0.871[7]
Column	0.784[2]	**0.808[3]**	**0.771[3]**	0.756[3]	**0.799[3]**	0.798[3]	**0.778[3]**	0.763[3]
Dermatology	0.956[16]	**0.970[16]**	0.915[17]	**0.929[17]**	0.929[16]	**0.962[14]**	0.858[15]	**0.873[12]**
HillValley	0.936[49]	**0.978[50]**	0.831[38]	**0.886[11]**	**0.543[1]**	0.523[11]	**0.900[41]**	0.558[42]
Immunotherapy	**0.796[1]**	0.776[1]	**0.733[2]**	0.672[3]	**0.795[1]**	0.731[1]	**0.723[1]**	0.643[1]
Leaf	0.360[5]	**0.369[7]**	**0.512[7]**	0.487[7]	**0.594[7]**	0.550[7]	**0.444[7]**	0.132[7]
Wine	0.927[6]	**0.946[6]**	**0.854[6]**	0.867[6]	0.913[5]	**0.933[5]**	**0.722[6]**	0.720[6]
WineQuality	**0.571[5]**	0.569[5]	**0.550[5]**	0.546[5]	**0.557[4]**	0.550[5]	**0.539[5]**	0.533[5]

Table III. Comparison between TPCA and PCA accuracy rates with up to 50% of the data features.

the four classifiers with PCA and TPCA applied to dimensionality reduction. For the HillValley dataset (1-NN) the mean accuracy is 0.90 for TCPA (selecting 41 features) and 0.56 for PCA (42 features). Direct analysis of every dataset prediction accuracy in the TPCA and standard PCA demonstrated that TPCA performed a reduction capable of achieving higher accuracy for three of the four classifiers for 70% datasets. If we add the datasets where TPCA reached higher accuracy for two of the four classifiers, TPCA had satisfactory performance in 90% of the datasets.

5. CONCLUSION

We proposed a class-threshold feature selection technique focused on classification problems during this research. By running tests on different classifiers and datasets, we highlighted the efficiency of the feature selector. Beyond that, we combined our selector with the already known Principal Component Analysis algorithm and, through tests, validated that such a combination (TPCA) can surpass the original method in a wide range of datasets. For future works, we want to create new varieties of the TFS with different feature extraction methods such as Supervised PCA, MCEPCA and Linear-Optimal-Low-Rank Projections (LOL).

Acknowledgement. This work was partially supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq (Proc. 401832/2021-0).

REFERENCES

Ali, A. K. and Erçelebi, E. Automatic modulation classification using different neural network and pca combinations. *Expert Systems with Applications* vol. 178, pp. 114931, 2021.

Arun Kumar, R., Vijay Franklin, J., and Koppula, N. A comprehensive survey on metaheuristic algorithm for feature selection techniques. *Materials Today: Proceedings*, 2022.

Beiranvand, F., Mehrdad, V., and Dowlatshahi, M. B. Unsupervised feature selection for image classification: A bipartite matching-based principal component analysis approach. *Knowledge-Based Systems*, 2022.

Biagetti, G., Crippa, P., Falaschetti, L., Luzzi, S., and Turchetti, C. Classification of alzheimer's disease from eeg signal using robust-pca feature extraction. *Procedia Computer Science* vol. 192, pp. 3114–3122, 2021. Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 25th International Conference KES2021.

de Carvalho, T. B. A., Sibaldo, M. A. A., and Tsang, I. R. Principal component analysis for supervised learning: a minimum classification error approach. *JOURNAL OF INFORMATION AND DATA MANAGEMENT* 8 (2), 2017.

Eustáquio, F. and Nogueira, T. Evaluating the numerical instability in fuzzy clustering validation of high-dimensional data. *Theoretical Computer Science* vol. 805, pp. 19–36, 2020.

Ganjei, M. A. and Boostani, R. A hybrid feature selection scheme for high-dimensional data. *Engineering Applications of Artificial Intelligence* vol. 113, pp. 104894, 2022.

Gárate-Escamila, A. K., Hajjam El Hassani, A., and Andrès, E. Classification models for heart disease prediction using feature selection and pca. *Informatics in Medicine Unlocked* vol. 19, pp. 100330, 2020.

Huang, P., Ye, Q., Zhang, F., Yang, G., Zhu, W., and Yang, Z. Double l2,p-norm based pca for feature extraction. *Information Sciences* vol. 573, pp. 345–359, 2021.

Jolliffe, I. pp. 10–28. In , *Mathematical and Statistical Properties of Population Principal Components*. Springer New York, New York, NY, pp. 10–28, 2002.

Jolliffe, I. A 50-year personal journey through time with principal component analysis. *Journal of Multivariate Analysis* vol. 188, pp. 104820, 2022. 50th Anniversary Jubilee Edition.

Liang, N., Tuo, Y., Deng, Y., and He, T. Pca-based svm classification for simulated ice floes in front of sluice gates. *Polar Science*, 2022.

Liu, Y. and Durlofsky, L. J. 3d cnn-pca: A deep-learning-based parameterization for complex geomodels. *Computers Geosciences* vol. 148, pp. 104676, 2021.

Maćkiewicz, A. and Ratajczak, W. Principal components analysis (pca). *Computers Geosciences* 19 (3): 303–342, 1993.

Mi, J.-X., Yang, L.-J., Zhou, L.-F., Sun, Y.-R., and Heng, K. Symmetrical feature extraction via novel mirror pca. *Neurocomputing* vol. 452, pp. 690–704, 2021.

Priyanka and Kumar, D. Feature extraction and selection of kidney ultrasound images using glcm and pca. *Procedia Computer Science* vol. 167, pp. 1722–1731, 2020. International Conference on Computational Intelligence and Data Science.

Song, P., Zhao, C., and Huang, B. Sfnet: A slow feature extraction network for parallel linear and nonlinear dynamic process monitoring. *Neurocomputing* vol. 488, pp. 359–380, 2022.

Tang, C., Liu, X., Li, M., Wang, P., Chen, J., Wang, L., and Li, W. Robust unsupervised feature selection via dual self-representation and manifold regularization. *Knowledge-Based Systems* vol. 145, pp. 109–120, 2018.

Tumer, K. and Ghosh, J. Estimating the bayes error rate through classifier combining. In *Proceedings of 13th International Conference on Pattern Recognition*. Vol. 2. pp. 695–699 vol.2, 1996.

Vinodhini, G. and Chandrasekaran, R. M. Sentiment classification using principal component analysis based neural network model. In *International Conference on Information Communication and Embedded Systems (ICICES2014)*. pp. 1–6, 2014.

Wang, Y., Liu, W., and Liu, X. Explainable ai techniques with application to nba gameplay prediction. *Neurocomputing* vol. 483, pp. 59–71, 2022.

Weiwei, H. Classification of sport actions using principal component analysis and random forest based on three-dimensional data. *Displays* vol. 72, pp. 102135, 2022.

Woo, S. and Lee, C. Incremental feature extraction based on decision boundaries. *Pattern Recognition* vol. 77, pp. 65–74, 2018.

Zhou, R., Gao, W., Ding, D., and Liu, W. Supervised dimensionality reduction technology of generalized discriminant component analysis and its kernelization forms. *Pattern Recognition* vol. 124, pp. 108450, 2022.

Zhu, T., Cheng, X., Cheng, W., Tian, Z., and Li, Y. Principal component analysis based data collection for sustainable internet of things enabled cyber–physical systems. *Microprocessors and Microsystems* vol. 88, pp. 104032, 2022.