Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 11, No. 4, 2023, 525-534.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

Autoencoder-PCA-based Online Supervised Feature Extraction-Selection Approach

Amir Mehrabinezhad[1], Mohammad Teshnehlab[2*] and Arash Sharifi[1]

1. Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.
2. Faculty of Electronic and Computer Engineering Department, K.N Toosi University of Technology, Tehran, Iran.

Article Info

Article History:
Received 20 November 2022
Revised 27 December 2022
Accepted 21 September 2023

DOI:10.22044/jadm.2023.12436.2390

Keywords:
Principal component analysis (PCA), Online PCA, Autoencoder, Stacked autoencoder, Semi-supervised learning.

**Corresponding author: Teshnehlab@eetd.kntu.ac.ir (M. Teshnehlab).*

Abstract

Due to the growing number of data-driven approaches, especially in artificial intelligence and machine learning, extracting appropriate information from the gathered data with the best performance is a remarkable challenge. The other important aspect of this issue is storage costs. The principal component analysis (PCA) and autoencoders (AEs) are samples of the typical feature extraction methods in data science and machine learning that are widely used in various approaches. The current work integrates the advantages of AEs and PCA for presenting an online supervised feature extraction selection method. Accordingly, the desired labels for the final model are involved in the feature extraction procedure and embedded in the PCA method as well. Also stacking the non-linear autoencoder layers with the PCA algorithm eliminated the kernel selection of the traditional kernel PCA methods. Besides the performance improvement proved by the experimental results, the main advantage of the proposed method is that, in contrast with the traditional PCA approaches, the model has no requirement for all samples to feature extraction. As regards the previous works, the proposed method can outperform the other state-of-the-art ones in terms of accuracy and authenticity for feature extraction.

1. Introduction

Nowadays, data science approaches have seen remarkable development. Accordingly, machine learning (ML)-based applications, various types of intelligent data streaming, and data mining-based applications are widely in service. In most of these services, the input data is employed for training or concluding from a machine learning model such as a regressor or classifier. High dimensionality is a significant problem when building a classification model since it may result in redundancy, feature noise, and computational complexity. Several benefits could be obtained by applying the dimensionality reduction techniques to a dataset [1, 2], some of them are as follows:

- Irrelevant, redundant, and noisy data could be removed.
- It takes less computation time.
- As the number of dimensions decreases, data storage space could be reduced.
- Data quality could be improved.
- Some algorithms could perform better on a more significant number of dimensions taken. Thus reducing these dimensions helps an algorithm work efficiently, and improves accuracy.
- It is challenging to visualize data in higher dimensions. Thus reducing the dimension may allow us to visualize patterns more clearly.
- It simplifies the process of classification and also improves efficiency. Generally, the dimensionality reduction is achieved through two different techniques: feature selection and feature extraction.

In feature selection, a subset of features is kept, while less relevant features are discarded. The feature subset is chosen to retain the essence of the original representation. Many feature selection methods exist including filters, wrappers, and embedded/hybrid methods [3, 4]. Feature extraction transforms the original feature space into a new lower-dimensional one. The initial features undergo various operations to produce new features, as the new features cannot be associated easily with their original components. Many state-of-the-art feature extraction techniques have been used to deal with high-dimensional datasets such as genetic algorithms (GAs) and partial least squares regression [5], ant colony optimization, k-means clustering [6], and PCA [7].

PCA is one of the oldest and most widely used techniques, an unsupervised linear dimension reduction technique [8]. The identification of Principal Components (PCs) is a set of uncorrelated features, which is the main aim of PCA. The first PC holds the most considerable variance in the dataset and in that order. Although it is a robust dimension reduction technique, it has some limitations. The PCA transformation, despite its widespread use, relies on second-order statistics. The principal components can be highly statistically dependent though uncorrelated, leading to PCA failing to find the most compact description of the data. PCA geometrically models the data as a hyperplane embedded in a space that is ambient space and requires a larger dimensional representation than would be found by a non-linear technique if the data components have non-linear dependencies. This has prompted the development of non-linear alternatives to PCA [9]. The PCA methods also fail to account for outliers common in realistic training sets because they employ least squares estimation techniques. The need to process all samples to calculate the covariance matrix, and consequently, eigenvalues is one of the considerable disadvantages of the PCA method. Although PCA is a strong and adaptable method, several restrictions should be known. A different selection or organization of your data might provide different findings since, for instance, PCA is susceptible to the sequence of observations and the selection of variables.

With the growing success of deep learning (DL) techniques, autoencoders (AEs) have been used for feature extractors as dimension reducers. AEs are unsupervised deep learning [10, 11] neural networks with back-propagation algorithms for learning. AEs represent the high-order input vector space to intermediate low-order vector space, and later, it reconstructs the output equivalent to the given input from intermediate low-order representation. This represents the dimensionality reduction characteristics like PCA [12], but PCA works only for linear transformation, and AEs work for both linear and non-linear data transformation. AEs are the common architectures and techniques for feature extraction with dimension reduction purposes. AE is a two-layer perception network with the same input and label vectors. After training, the output vector of the hidden layer in the AE is used as the extracted vector from the input. So far, various architectures have been developed to improve the performance of autoencoders. A significant number of these architectures are introduced to improve the hidden space distribution, i.e. adding sparsity to the hidden vectors so that the hidden vector of each input sample is unique and has a proper separability against the other vectors related to the other input vectors.

The current work proposes a novel online supervised feature extraction-selection method combining the AEs and PCA benefits. The proposed method considers the requirements of the defined model in the feature selection and extraction process using the supervised learning method. The major benefits that distinguish the proposed method from the previous state-of-the-art ones are as follows:

- In addition to the utilization of orthogonality, i.e. the unsupervised approach in methods such as PCA, it considers the target, label, and values of the dataset samples in the feature selection process in a supervised manner.
- By manipulation of the batch-based approach in the method, the requirement for offline analysis of data is eliminated; this strategy, in addition to reducing the need for storage and computation resources for loading and processing all the data simultaneously, creates the ability to use the approach in online cases that data compression needed such as video streaming.
- Based on the experimental results [13, 14], the proposed method, by purging redundant features from data, outperforms the model trained by the original dataset [15, 16].
- In the proposed method, non-linear transformation is embedded in the

autoencoder layers, so there is no need to use non-linear approaches such as kernel-PCA, and surely, their problems, like suitable kernel function selection, no longer matter.

The rest of this paper is organized as what follows. The related studies are reviewed in the second section to specify the main gaps and shortcomings. Then the third section illustrates the methodology used in the work. The results are discussed in the fourth section, and more comparisons are made. The conclusions and suggestions for future work are given in the fifth section.

2. Related Works

PCA seeks a projection matrix such that the covariance matrix of the projected data is full rank, inherently making PCA sensitive to noise and outliers [12, 17]. In practice, there are many extensions of PCA to help address some of its challenges and improve the efficiency of algorithms [8, 18]. They can be broadly classified into two categories: ℓ1-norm-based approaches and ©nuclear-norm-based approaches. Nuclear-norm-based methods aim to find clean data with a low-rank structure [19]. Generally, this kind of method does not directly generate a lower-dimension representation. Some representative methods are robust PCA (RPCA) [20], graph-based RPCA [15, 18], and non-convex RPCA [17, 19], which are typically used for foreground-background separation. Moreover, they are transductive methods and cannot handle out-of-samples. However, Bao [21] proposed an inductive approach targeted for clean data. Malladi [22] developed a computationally simple paradigm for image denoising using a superpixel-based PCA approach. Zhu [23] integrated PCA with manifold learning to learn the hash functions and achieve efficient similarity search. Unlike the nuclear-norm-based methods, ℓ1-norm PCA adopted ℓ1-norm to replace the squared Frobenius norm as the distance metric. For instance, L1-PCA minimized the ℓ1-norm reconstruction error [24]. Though it improved the robustness of PCA, it did not have rotational invariance [25]. Some methods maximized ℓ1-norm covariance [26, 27]. CS-ℓ1-PCA, developed by Liu [28], calculated robust subspace components by explicitly maximizing ℓ1 projection to enable low-latency video surveillance.

Notably, the aforementioned ℓ1-norm PCA methods need to calculate the data mean in the least square sense, which is not optimal for the non-Frobenius norm. Therefore, optimal mean RPCA (RPCA-OM) optimizes both the projection matrix and the mean [29]. Nevertheless, it can achieve the global mean [30]. Luo in [31] maximizes the projected ℓ1 differences between each pair of points. Though it avoids the mean computation, it could be stuck into bad local minima.

Moreover, a technique was given in another work to consciously use channel-wise reconstruction errors as a characteristic to identify aberrant signals [32]. An ML anomaly detection model compiles the channelwise reconstruction mistakes into an anomaly score after a convolutional autoencoder generates them. Using simulated data and actual automotive data, we perform tests to show the efficacy and applicability of the proposed approach. The findings demonstrated that the suggested technique significantly improves the detectability compared to the straightforward average of the reconstruction errors. AI techniques were utilized in conjunction with EEG, structural MRI, and functional MRI to diagnose schizophrenia in another work [33] automatically. Also several schizophrenia datasets and methods and tools were used to pre-process MR and EEG pictures. Despite such interest, many gaps and limitations need to be dealt with in the research work. A procedure for feature selection is presented in [34] that includes a binary teaching-learning-based optimization algorithm with mutation (BMTLBO).

One excellent method for reducing dimensionality is PCA, which may greatly aid in the reduction of a model's feature count. It may appear like a strong tool for a data scientist, but certain issues prevent it from being used for supervised machine learning applications. This issue has been solved in the current work as an innovation as the proposed online supervised feature extraction-selection approach is introduced.

3. Methodology

3.1. Supervised PCA

The supervised PCA conception until now considered a metric learning kernel estimation (MLKE) for the mean squared error (MSE) loss function. In this case, the MSE cost function is defined as the Mahalanobis distance between the model output vector and target vector as Eq. 1:

$$MSE(\hat{y}_{(x)}, y_{(x)}) = d(\hat{y}_{(x)}, y_{(x)}) = (\hat{y}_{(x)} - y_{(x)})^T M (\hat{y}_{(x)} - y_{(x)}) \quad (1)$$

where $\hat{y}_{(x)}, y_{(x)}$ are the model output and the target vector, respectively. x is the input vector, and M is the semi-positive definite covariance matrix (kernel). The M matrix should satisfy the following

condition as a semi-positive definite matrix:

$$v^T M v \geq 0 \quad (2)$$

where v is an arbitrary vector. To satisfy Eq. 2 for any vector, we can define the M matrix equal to the covariance matrix of the $\hat{y}_{(x)} - y_{(x)}$ vector as Eq. 3:

$$M = (\hat{y}_{(x)} - y_{(x)})(\hat{y}_{(x)} - y_{(x)})^T \quad (3)$$

In this condition, if the M matrix is an identity matrix, the loss function is defined as the Eulerian distance between the output and target vector, a common loss function for regression models. Nevertheless, in other conditions, using Eq. 1 as a loss function will consider the dimension dependency on each other in the result, so the dependent dimensions will have less priority than the dimensions with large eigenvalues (independent dimensions) in calculating the loss function amount. This approach is called supervised PCA but is limited to a certain loss function and model type (regression) as its drawback. Nevertheless, nowadays, according to the emergence of deep learning models, we have dozens of different loss functions and models for classification, etc., and this approach cannot be implemented in these models. To tackle this issue, we redefine the supervised PCA approach as a flexible parameter embedded in the model

3.2. Feature extractor

According to Figure 1, a three-layer stacked autoencoder (SAE) was used in the first section of the proposed model. This model reduces the input dimension to 10; this parameter has been set according to the dataset and experimental results. The learning process of each layer in the SAE was defined in an unsupervised manner, using stochastic gradient descent (SGD) [35] concerning the mean squared error (MSE) loss function. To improve the model discrimination property and sparsity of the hidden vector (output vector of the encoder layer), we added the L1 norm [36] of the hidden vector to the loss function, as shown in Eq. (4).

$$E_{(x_i,\hat{x}_i)} = \frac{1}{2} e_i^2 = \frac{1}{2}(x_i - \hat{x}_i)^2 \quad (4)$$

where E is the MSE loss function value for each input vector, e is the i-th error vector, and x_i $\hat{x}_i$ the i-th input and output vector, respectively. By assuming an input vector size of 500 and a hidden vector size of 30 for an autoencoder, the feed-forward equations for single-layer unsupervised learning are as follows:

$$net^{EN}_{30\times1} = W^{EN}_{30\times500} \times x^T_{1\times500} + b^{EN}_{30\times1} \quad (5)$$

$$h_{30\times1} = o^{EN}_{30\times1} = f(net^{EN}_{30\times1}) \quad (6)$$

$$net^{DE}_{500\times1} = W^{DE}_{500\times30} h_{30\times1} + b^{DE}_{500\times1} \quad (7)$$

$$\hat{x}_{500\times1} = o^{DE}_{500\times1} = f(net^{DE}_{500\times1}) \quad (8)$$

where W^{EN} and W^{DE} are the encoder layer and decoder layer weights, $f(.)$ is the hyperbolic tangent activation function, and $x, \hat{x}, h$ are the input vector, output vector, and the hidden vector, respectively. The decoder layer weights are updated per input sample using Eq. (9):

$$W^{DE}_{(k+1)} = W^{DE}_{(k)} - \eta \frac{\partial E}{\partial W^{DE}}(k) \quad (9)$$

where η is the autoencoder learning rate, and k is the sample index. The $\frac{\partial E}{\partial W^{DE}}$ term derivative chain rules are as follows Eq. (10):

$$\frac{\partial E}{\partial W^{DE}} = \underbrace{\frac{\partial E}{\partial e}}_{e} \underbrace{\frac{\partial e}{\partial \hat{x}}}_{-1} \underbrace{\frac{\partial \hat{x}}{\partial net^{DE}}}_{f'_{(net^{DE})}} \underbrace{\frac{\partial net^{DE}}{\partial W^{DE}}}_{h} \quad (10)$$

where E is the MSE loss function value for the corresponding input vector, e is the error vector, and x_i $\hat{x}_i$ are the input and output vector, respectively. The encoder layer weights are updated based on the decoder weights transpose, shown in Eq. (11):

$$W^{EN}_{(k+1)} = (W^{DE}_{(k+1)})^T \quad (11)$$

Also it is possible to have algorithms of the learning process for the encoder layer weights by using back-propagation and gradient descend optimization method as Eq. (12):

$$W^{DE}_{(k+1)} = W^{DE}_{(k)} - \eta \frac{\partial E}{\partial W^{DE}}(k) \quad (12)$$

A pair of the encoder and decoder layers is configured as a two-layer perception network to create each layer in the SAE, shown in Figure 1. For this perception network, the input and target vectors are the same. Hyperbolic tangent and linear functions are the assigned activation functions for the encoder and decoder layer, respectively. The weights update for the encoder and decoder layers is performed by leveraging the stochastic gradient descent (SGD) per sample. Updating weights for each sample increases the convergence rate but, on

the other hand, may cause instability in the learning process that the learning rate can control; this issue is considered in the implementation. After completion of the training process, the decoder layer will be eliminated, and the encoder layer will be placed in the SAE architecture.

In the SAE, each layer output vector (hidden vector) is considered the input vector of the subsequent layer. The output vector of the last encoder layer is the SAE final output vector. Figure 2 shows the final architecture of the SAE model.

3.3. Online PCA

In this section, the batch normalization [37] is applied to the output vector of the SAE at first. Then to perform the online PCA, the batch samples calculate the ten-dimensional vector's eigenvalues. In the unsupervised mode, these eigenvalues can be used for feature selection from extracted vectors from the SAE. However, besides these values, a supervised method in the next section was defined to improve the feature selection process.

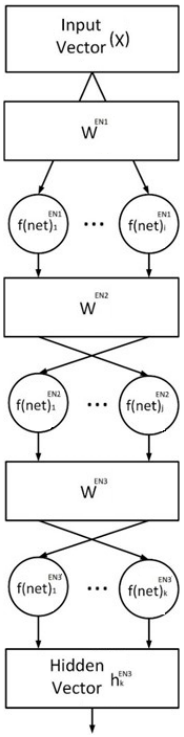

Figure 1. Autoencoder architecture.

3.4. Supervised learning for feature selection

In this step, an α_i scalar coefficient for each eigenvalue obtained from the previous section is defined and shown in Eq. (13).

$$\alpha_i' = \alpha_i \lambda_i \tag{13}$$

After multiplying the alpha values by the eigenvalues according to Eq. (13), a Soft-Max function [38] is defined to the resulting values, shown in Eq. (14).

$$\beta_i = Soft - Max(\alpha_i') = \frac{e^{\alpha_i'}}{\sum_j e^{\alpha_j'}} \tag{14}$$

The vector obtained by the Soft-Max function is multiplied by the 10-dimensional vector, the output vector of the SAE, and the resulting vector is considered the input vector for a 3-three-layer perception network, shown in 12. We train the α_i coefficients beside the network weights. By this approach, we define an essential coefficient for each input dimension.

After completing the training process, we can choose the top K features from the 10-dimensional feature space, which have more significant values than others.

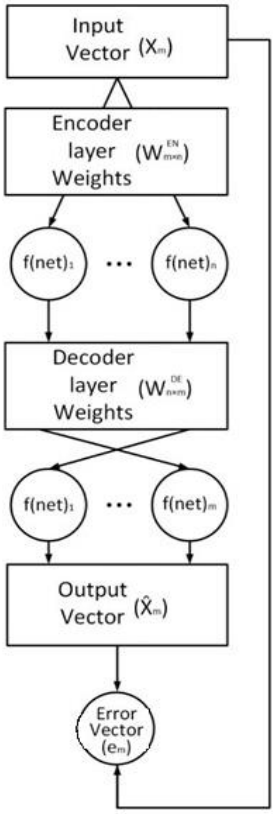

Figure 2. Stacked autoencoder architecture.

In the output of the Soft-Max function, more significant β_i values.

$$x_i' = x_i \beta_i \tag{15}$$

The chain rule for training the α vector is shown in Eq. (16).

$$\frac{\partial E}{\partial \alpha}=\frac{\partial E}{\ldots}\cdots\frac{\ldots}{\partial o^{net}}\frac{\partial o^{net}}{\partial x'}\underbrace{\frac{\partial x'}{\partial \beta}}_{x}\underbrace{\frac{\partial \beta}{\partial \alpha'}}_{SoftMax'(\alpha')}\underbrace{\frac{\partial \alpha'}{\partial \alpha}}_{\lambda} \quad (16)$$

where E, o^{net}, and $\acute{x}$ are loss functions, the perceptron network output vector and perceptron network input vector are obtained from 12, respectively. The loss function in the proposed method is the mean squared error as 5. Where $\boldsymbol{y}, \hat{\boldsymbol{y}}$ are labels and model outputs in a batch, N is the number of samples per batch, and $\boldsymbol{y}_i, \hat{\boldsymbol{y}}_i$ are i-th label and i-th model output in the batch.

$$E_{(y,\hat{y})}=\frac{1}{2N}\sum_{i=1}^{N}(y_i-\hat{y}_i)^2 \quad (17)$$

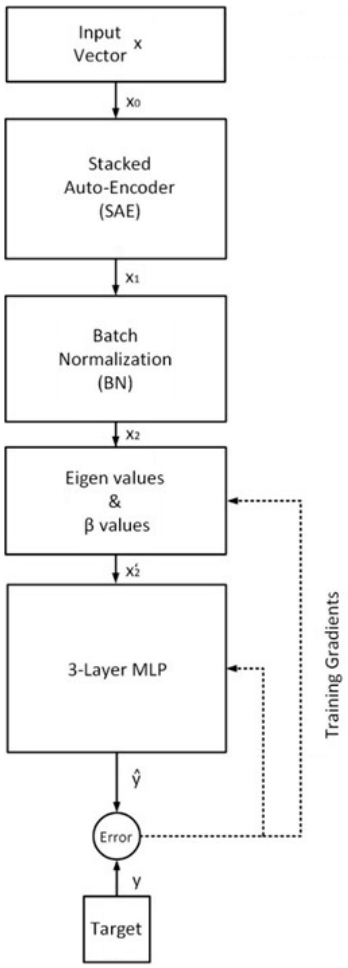

Figure 3. Supervised online PCA diagram.

4. Experiments and Results

4.1. Dataset

Considering the issues related to feature selection in datasets whose components are non-linearly related, first, we selected the Madelon dataset. The Madelon dataset is an artificial dataset made for evaluating feature selection approaches. The samples of this dataset have input vectors with 500 dimensions, and binary labels are also defined for each sample in the dataset. In addition to the Madelon dataset, we also used the Spambase dataset to improve the evaluation of the performance of the proposed method. The Spambase dataset has 57 input attributes in each of its samples; the labels of this dataset are also binary.

4.1. Implementation and results

In the training process of the three-layer perceptron network, the batch size is set to 16. It was found that the three-layer for SAE has outperformed. We applied PCA to the data after and before applying the SAE; the results showed that the correlation between different feature space dimensions was preserved along with the SAE layers, so using PCA after SAE is reasonable. Also by comparing the beta values and eigenvalues, it was observed that the importance of the same feature is different in unsupervised PCA and supervised learned β_i values. After these results, we trained a 3-layer MLP three times: without feature selection, using vanilla online PCA, and finally, using supervised online PCA, the proposed method. The test dataset confusion matrices for both Madelon and Spam-based datasets are shown in Figures 2 and 3, respectively. As shown in the figures, using supervised online PCA outperforms vanilla online PCA; in the case of the Madelon dataset, supervised online PCA is even better than using a dataset without feature selection. Based on these figures, the model reduces biasing to a certain class in general, consequently tackling overfitting phenomena in the learning process.

To better introspect the proposed approach, we compare the trained Beta values and Eigenvalues for Madelon and Spam-based datasets in Figures 6 and 7, respectively. Based on these figures, the eigenvalues represent a better concept of relative interdependence of the features but in an unsupervised manner. The beta values due to the SoftMax function focus on a few features. Therefore, this approach suits models with small-dimension input vectors, especially single-input models.

Due to embedding autoencoder layers in the proposed model, the non-linear transform implementation, critical in most cases of facing real datasets, was also implemented. Hence, there is no need to be concerned about general challenges in similar methods such as transformer kernel selection and kernel function input features. The

practical result shows that the model outperforms even the original dataset in some cases because of the elimination of redundant features.

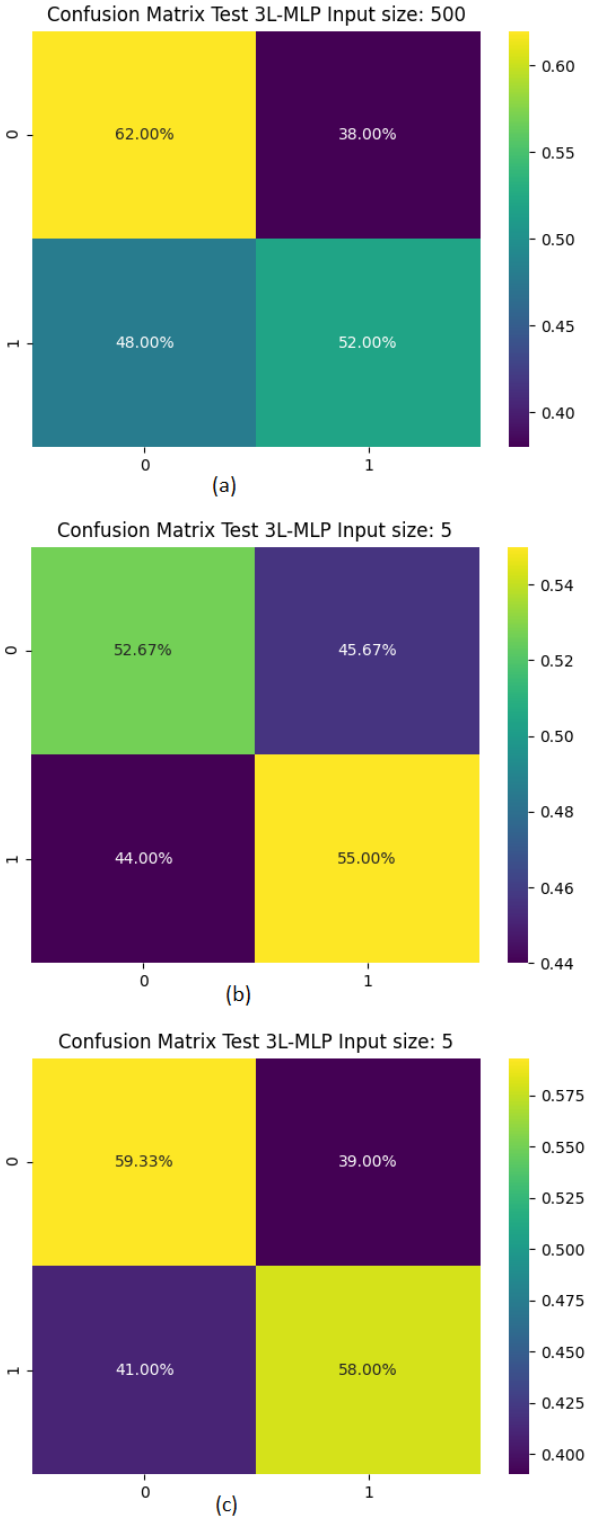

Figure 4. Madelon test set confusion matrices, a) Without feature selection b) Online-PCA c) Supervised online-PCA.

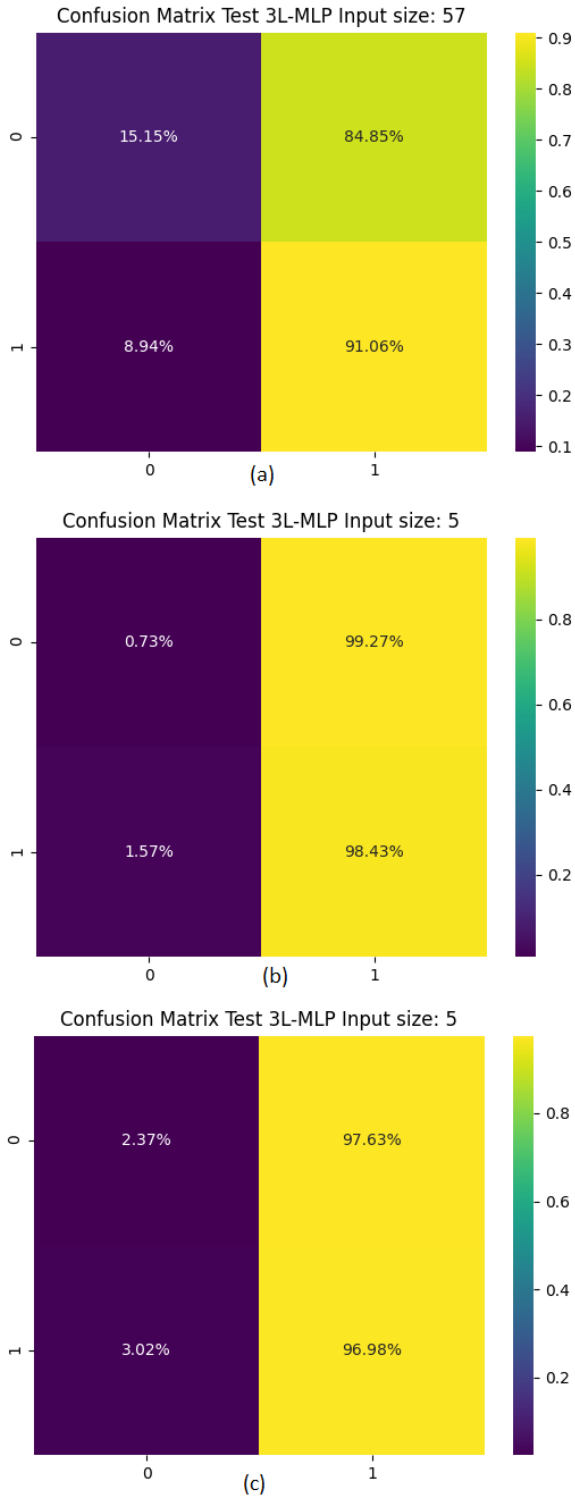

Figure 5. Spam-based test set confusion matrices, a) Without feature selection b) Online-PCA c) Supervised online-PCA.

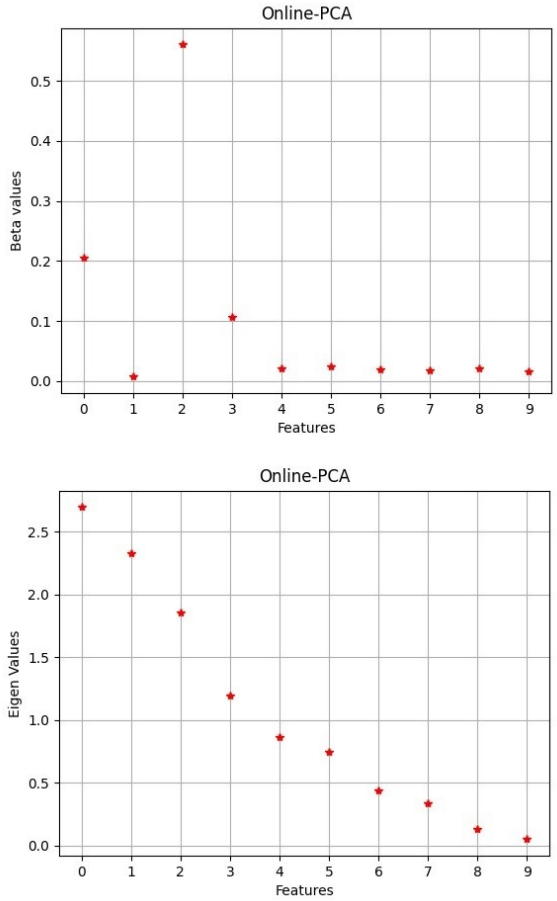

Figure 6. Madelon dataset Beta values and eigenvalues comparison.

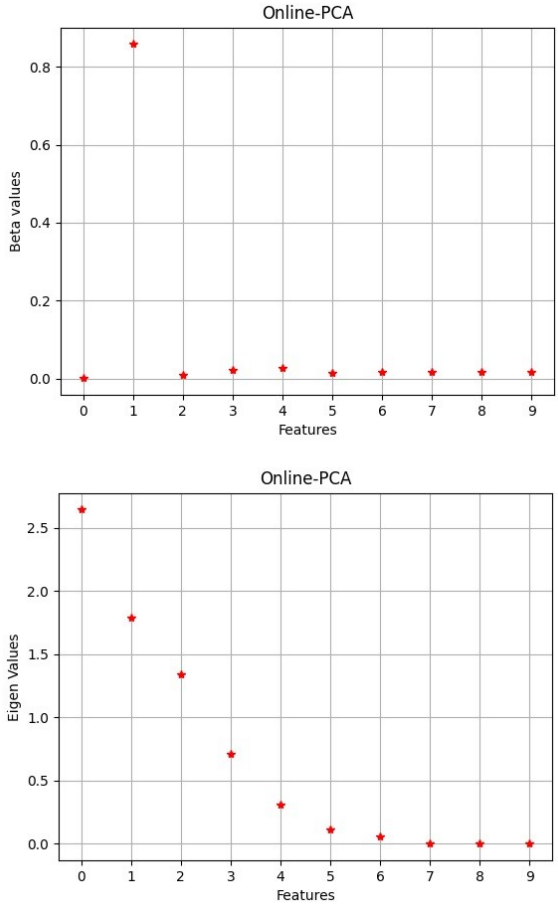

Figure 7. Spam-based dataset Beta values and eigenvalues comparison.

The main outperforming aspects of the proposed model are the enhancement of separability and precision and reduction bias to the classes with a massive number of samples in the dataset; these aspects appear to trace the desired values in regression problems better. Figures 4 and 5 compare the proposed model with the traditional PCA and original dataset classification results using confusion matrixes. Due to the experimental results, the approach tends to assign greater score values to a single feature, while PCA-based methods provide relative importance (relative independence) for each feature. From this viewpoint, the method is suitable for single input models such as auto-regressive moving average (ARMA) but does not provide analytical information about the independence of features. Figures 6 and 7 compare the proposed model feature selection scores with the traditional PCA scores. Although this difference exists between the proposed approach and PCA-based approaches, in most cases, the selected feature set is the same in both methods.

Finally, we compared the overall results of the proposed method with traditional PCA and original dataset results using common metrics: accuracy, sensitivity, and specificity in Tables 1 and 2 for both datasets.

Furthermore, the proposed method can outperform those given in the past reports. When employing the statistical feature selection techniques, each input variable's connection to the target variable is assessed, and the input variables with the strongest relationships are chosen. Although the choice of statistical measures relies on the data type of both the input and output variables, these techniques may be quick and efficient. The main issue neglected in the previous works is that by eliminating unnecessary information, feature selection enables reducing the number of dimensions. However, by using fake sets to convert the data into fewer dimensions, PCA preserved the same information.

Table 1. Madelon dataset metrics.

Method	*Accuracy (%)*	*Sensitivity (%)*	*Specificity (%)*
Original dataset	57	56	57
Online-PCA	54	54	54
Supervised online PCA (Proposed method)	59	59	54

Table 2. Spam-based dataset metrics.

Method	*Accuracy (%)*	*Sensitivity (%)*	*Specificity (%)*
Original dataset	53	62	57
Online-PCA	49	31	49
Supervised online PCA (Proposed method)	49	43	49

5. Conclusion

In summary, a feature extraction-selection approach was proposed in the study, emphasizing dimension reduction and online data processing. The feature extraction procedure considered both dimensional independence and target (desired) values of the configured model. In addition, the proposed approach eliminated the requirement to process and load all dataset samples simultaneously, thanks to this quiddity. Notably, the method could be implemented in stream processing models. As reported in the literature, the proposed method can outperform its counterparts in accuracy and authenticity. Future investigations are necessary to validate the kinds of conclusions that can be drawn from this study.

References

[1] R. Zebari,A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, "A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction", *Journal of Applied Science and Technology Trends*, vol. 1 ,pp. 56-70, 2020.

[2] Juvonen A, Sipola T, and Hämäläinen T, Online anomaly detection using dimensionality reduction techniques for HTTP log analysis, *Computer Networks*, vol. 91, pp. 46-56 ,2015.

[3] S.A. Alsenan, I.M. Al-Turaiki, and A.M. Hafez, "Auto-KPCA: A Two-Step Hybrid Feature Extraction Technique for Quantitative Structure–Activity Relationship Modeling", *IEEE Access* 9, pp. 2466-2477, 2020.

[4] A.U. Khan, "Descriptors and their selection methods in QSAR analysis: paradigm for drug design", *Drug discovery today*,vol 21, pp. 1291-1302, 2016.

[5] N. Sukumar,G. Prabhu, and P. Saha, "Applications of genetic algorithms in QSAR/QSPR modeling, in: Applications of metaheuristics in process engineering" *Springer*, pp. 315-324, 2014.

[6] E. Bonabeau,M. Dorigo,G. Theraulaz, "Inspiration for optimization from social insect behaviour", *Nature*, vol. 406, pp. 39-42, 2000.

[7] C. Yoo, M. Shahlaei, "The applications of PCA in QSAR studies: A case study on CCR5 antagonists", *Chemical biology & drug design*, vol. 91, pp. 137-152, 2018.

[8] S. Nanga, A.T. Bawah, B.A. Acquaye, M-I Billa, F.D. Baeta, N.A. Odai, S.K. Obeng, and A.D. Nsiah, "Review of dimension reduction methods", *Journal of Data Analysis and Information Processing*, vol 9, pp. 189-231, 2021.

[9] N. Kambhatla and T.K. Leen, "Dimension reduction by local principal component analysis", *Neural computation*,vol 9, pp. 1493-1516, 1997.

[10] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks", *Ieee Access* 5, pp. 21954-21961, 2017.

[11] N. Shone, T.N. Ngoc, V.D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection", *IEEE transactions on emerging topics in computational intelligence*, vol 2, pp. 41-50, 2018.

[12] K. Singh, L. Kaur, and R. Maini, "Comparison of principle component analysis and stacked autoencoder on NSL-KDD dataset", *Computational Methods and Data Engineering: Proceedings of ICMDE 2020, Volume 1*, pp. 223-241, *Springer*, 2021.

[13] J. Oh, N. Kwak, "Generalized mean for robust principal component analysis", *Pattern Recognition*, vol. 54, pp.116-127, 2016.

[14] Q. Wang, Q. Gao, X. Gao, and F. Nie, "Optimal mean two-dimensional principal component analysis with F-norm minimization", *Pattern recognition*, vol. 68, pp. 286-294, 2017.

[15] F. Farahnakian and J. Heikkonen, "A deep auto-encoder based approach for intrusion detection system", *20th International Conference on Advanced Communication Technology (ICACT): IEEE*, pp. 178-183, 2018.

[16] B. Lee, S. Amaresh, C. Green, and D. Engels, "Comparative study of deep learning models for network intrusion detection", *SMU Data Science Review*, 1 p. 8, 2018.

[17] N. Shahid, V. Kalofolias, X. Bresson, M. Bronstein, and P. Vandergheynst, "Robust principal component analysis on graphs", *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2812-2820, 2015.

[18] Z. Kang, H. Liu, J. Li, X. Zhu, and L. Tian, "Self-paced principal component analysis", *Pattern Recognition*, vol. 142, p. 109692, 2023.

[19] C. Peng, C. Chen, Z. Kang, J. Li, and Q. Cheng, "RES-PCA: A scalable approach to recovering low-rank matrices", *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7317-7325, 2019.

[20] E.J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?", *Journal of the ACM (JACM)*, vol. 58, pp. 1-37, 2011.

[21] B.K. Bao, G. Liu, C. Xu, and A. Yan, "Inductive robust principal component analysis", *IEEE transactions on image processing*, vol. 21, pp. 3794-3800, 2012.

[22] S.R.S. Malladi, S. Ram, and J.J. Rodríguez, "Image denoising using superpixel-based PCA", *IEEE Transactions on Multimedia*, vol. 23, pp. 2297-2309, 2020.

[23] Z. Zhu, X. Li, S. Zhang, Z. Xu, L. Yu, and C. Wang, "Graph PCA hashing for similarity search", *IEEE Transactions on Multimedia*, vol. 19, pp. 2033-2044, 2017.

[24] Q. Ke and T. Kanade, "Robust l/sub 1/norm factorization in the presence of outliers and missing data by alternative convex programming", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*: IEEE, pp. 739-746, 2005.

[25] C. Ding, D. Zhou, X. He, and H. Zha, "R 1-pca: rotational invariant l 1-norm principal component analysis for robust subspace factorization", *Proceedings of the 23rd international conference on Machine learning*, pp. 281-288, 2006.

[26] R. Wang, F. Nie, X. Yang, F. Gao, and M. Yao, "Robust 2DPCA With Non-greedy $\ell _ {1} $-Norm Maximization for Image Analysis", *IEEE transactions on cybernetics*, vol. 45, pp. 1108-1112, 2014.

[27] F. Ju, Y. Sun, J. Gao, Y. Hu, and B. Yin, "Image outlier detection and feature extraction via L1-norm-based 2D probabilistic PCA", *IEEE Transactions on Image Processing*, vol. 24, pp. 4834-4846, 2015.

[28] Y. Liu and D.A. Pados, "Compressed-sensed-domain l 1-pca video surveillance", *IEEE Transactions on Multimedia*, vol. 18, pp. 351-363, 2016.

[29] F. Nie, J. Yuan, and H. Huang, "Optimal mean robust principal component analysis", *International conference on machine learning: PMLR*, pp. 1062-1070, 2014.

[30] Z. Song, D.P. Woodruff, and P. Zhong, "Low rank approximation with entrywise l1-norm error", *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 688-701, 2017.

[31] M. Luo, F. Nie, X. Chang, Y. Yang, A. Hauptmann, and Q. Zheng, "Avoiding optimal mean robust PCA/2DPCA with non-greedy l1-norm maximization", *Proceedings of International Joint Conference on Artificial Intelligence*, pp. 1802-1808, 2016.

[32] M. Kwak and S.B. Kim, "Unsupervised Abnormal Sensor Signal Detection With Channelwise Reconstruction Errors", *IEEE Access*, vol. 9, pp. 39995-40007, 2021.

[33] A. Tyagi, V.P. Singh, and M.M. Gore, "Towards artificial intelligence in mental health: a comprehensive survey on the detection of schizophrenia", *Multimedia Tools and Applications*, vol. 82, pp. 20343-20405, 2023.

[34] S. Hosseini, M. Khorashadizadeh, "Efficeint feature selection method using binary teaching-learning-based optimizatin algorithm", *Journal of artificial intelligence and data mining (JAIDM)*, vol. 11, No.1, pp. 29-37, 2023.

[35] J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function", *The Annals of Mathematical Statistics*, pp. 462-466, 1952.

[36] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, "Optimization with sparsity-inducing penalties", *Foundations and Trends® in Machine Learning*, vol. 4, pp. 1-106, 2012.

[37] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", *International conference on machine learning: pmlr*, pp. 448-456, 2015.

[38] J. Bridle, "Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters", *Advances in neural information processing systems*, vol. 2, 1989.

رویکردی برای انتخاب یا استخراج ویژگی با نظارت برخط مبتنی بر تحلیل مولفه اصلی و خودرمزگذار

امیر محرابی نژاد[۱]، محمد تشنه لب[۲،*] و آرش شریفی [۲]

[۱] دانشکده مهندسی کامپیوتر، واحد علوم و تحقیات، دانشگاه آزاد اسلامی، تهران، ایران .

[۲] دانشکده مهندسی کامپیوتر و الترونیک، دانشگاه خواجه نصیر الدین طوسی، تهران، ایران.

ارسال ۲۰۲۲/۱۱/۲۰؛ بازنگری ۲۰۲۲/۱۲/۲۷؛ پذیرش ۲۰۲۳/۰۹/۲۱

چکیده:

با توجه به رشد روزافزون رویکردهای داده محور، به ویژه در هوش مصنوعی و یادگیری ماشین، استخراج اطلاعات مناسب از داده های جمع آوری شده با بهترین عملکرد چالشی قابل ملاحظه است. جنبه مهم دیگر این موضوع هزینه های ذخیره سازی است. تحلیل مؤلفه اصلی (PCA) و خودرمزگذارها (AEs) نمونه‌هایی از روش‌های استخراج ویژگی در علم داده و یادگیری ماشین هستند که به طور گسترده در رویکردهای مختلف استفاده می شوند. مقاله ارایه شده، از مزایای خودرمزگذارها و تحلیل مولفه اصلی برای ارائه روش انتخاب و استخراج ویژگی تحت نظارت برخط بهره گرفته است. بر این اساس، برچسب‌های مورد نظر برای مدل نهایی در فرآیند استخراج ویژگی نقش دارند و در روش تحلیل مولفه اصلی نیز تعبیه می‌شوند. همچنین انباشتن لایه‌های رمزگذار خودکار غیرخطی با الگوریتم تحلیل مولفه اصلی، انتخاب هسته در روش‌های تحلیل مولفه اصلی مبتنی بر هسته قدیمی را حذف می کند. علاوه بر این، بهبود عملکرد توسط نتایج تجربی ارایه شده است. مزیت اصلی روش پیشنهادی این است که، برخلاف رویکردهای سنتی تحلیل مولفه اصلی، مدل ارایه شده، هیچ نیازی برای همه نمونه‌ها برای استخراج ویژگی ندارد. با توجه به کارهای قبلی، روش پیشنهادی می‌تواند از نظر دقت و اعتبار برای استخراج ویژگی از دیگر روش‌های پیشرفته برتر باشد .

کلمات کلیدی: تحلیل مولفه اصلی، تحلیل مولفه اصلی برخط، خودرمزگذار، خودرمزگذار پشته‌ای، یادگیری نیمه نظارتی.