Journal of Soft Computing and Computer Applications

Manuscript 1012

New Feature Selection Using Principal Component Analysis

Zaid Mundher Radeef

Soukaena Hassan Hashem

Ekhlas Khalaf Gbashi

JOURNAL OF SOFT COMPUTING AND COMPUTER APPLICATIONS 2024;1:1012

ORIGINAL STUDY

New Feature Selection Using Principal Component Analysis

Zaid Mundher Radeef *, Soukaena Hassan Hashem, Ekhlas Khalaf Gbashi

University of Technology Iraq, Department of Computer Science, Al-Sina'a St., Al-Wehda District, 10066 Baghdad, Iraq

ABSTRACT

Dimensionality reduction techniques streamline machine learning by reducing data complexity, improving model accuracy, and cutting computational costs. They remove noise and irrelevant features, making models faster and more efficient. These techniques also enhance data visualization and interpretation by condensing data into manageable, insightful dimensions. Ultimately, dimensionality reduction leads to simpler, more interpretable models without sacrificing critical information, making it a cornerstone of efficient data analysis and machine learning applications. Theoretically, feature extraction tends to create new features that encapsulate more information by combining multiple existing features, resulting in more concentrated and informative features. In contrast, feature selection involves choosing a subset of the original features without altering their content. In this paper, a feature selection method based on Principal Component Analysis (PCA) is proposed, along with a comparative study of PCA performance as a feature extraction technique and the newly proposed feature selection method. The proposed feature selection method utilizes the variance scored by the principal component to identify the features with the most effect on the principal component's variance and selects them as the best set of features.

Experimental results demonstrate that the proposed PCA-based feature selection achieves comparable or improved performance across various classifiers, while maintaining high accuracy and precision, even with fewer features. For instance, when using the proposed method with Network Security Laboratory - Knowledge Discovery in Databases (NSL-KDD) to select only one feature and employing six different classifiers (Decision Tree, Naive Bayes, Logistic Regression, K-Neighbors Classifier, XGBoost, and AdaBoost) to evaluate performance, the accuracy of 80.88%, 81.29%, 43.07%, 44.53%, 84.94%, and 82.87% were obtained using the listed classifiers in the same order. On the other hand, when using PCA for feature extraction the following accuracy values, listed in the same classifier order, were obtained: 76.64%, 76.10%, 43.07%, 47.40%, 80.57%, and 82.05%, demonstrating that the proposed method delivers higher accuracies. Similarly, for the mushroom dataset, the accuracies were 51.38%, 51.38%, 48.62%, 51.06%, 87.08%, and 86.22%, compared to 50.14%, 50.30%, 50.88%, 50.59%, 73.60%, and 71.51%.

Received 19 July 2024; accepted 22 December 2024.
Available online 31 December 2024

* Corresponding author.
E-mail addresses: cs.20.03@grad.uotechnology.edu.iq (Z. M. Radeef), soukaena.h.hashem@uotechnology.edu.iq (S. H. Hashem), ekhlas.k.gbashi@uotechnology.edu.iq (E. K. Gbashi).

https://doi.org/10.70403/3008-1084.1012

Keywords: Feature extraction using PCA, Feature selection using PCA, Feature contribution, Principal component analysis, PCA comparative study

1. Introduction

In the era of big data, extracting meaningful insights from vast datasets is essential for the success of machine learning applications in domains such as finance and healthcare. As datasets grow in complexity and size, the curse of dimensionality poses significant challenges for machine learning methods, resulting in overfitting, increased computational costs, and reduced model performance. Dimensionality reduction techniques, particularly Principal Component Analysis (PCA), are essential tools for addressing these challenges by reducing high-dimensional data into a lower-dimensional space while retaining key information. Dimensionality reduction can be achieved through two main approaches: feature extraction and feature selection. Many techniques are available for both tasks, and some can be adapted to serve one of these approaches [1].

PCA is a widely used technique that supports both feature selection and feature extraction [2]. By performing a linear transformation, PCA reduces the original dimensions to fewer components, called Principal Components (PCs). Each PC is a linear combination of the original features, with weights indicating each feature's contribution to the component. These weights identify the features with the greatest influence on the dataset [3]. Despite the advantages of PCA, a significant research gap exists in leveraging PCA effectively for feature selection instead of feature extraction [4]. Most studies emphasize PCA's role in feature extraction while overlooking its potential for feature selection.

This study introduces a novel method for applying PCA specifically to feature selection instead of feature extraction. The approach assesses the variance of each principal component and the contributions of each feature to determine its impact on overall data variance. This method enables the identification and retention of the most relevant features while maintaining the interpretability of the original data and includes a comparative study of PCA's use in feature extraction versus feature selection using the proposed method.

The main contributions of this study can be stated as follows:

- **New Feature Selection Approach:** Proposes a novel method for employing PCA as a feature selection technique instead of feature extraction.
- **Variance Analysis:** Utilizing the obtained variance of PCs to evaluate the significance of individual features.
- **Comparative Study:** Conducts a comprehensive evaluation of the performance differences between using PCA for feature selection and feature extraction, tested with various classifiers as listed in the results.

The choice of PCA to address the feature selection problem arises from its capacity to reduce data dimensionality efficiently by capturing the most critical information through variance analysis. Although PCA is traditionally used as a feature extraction technique, transforming original features into a new set of uncorrelated components referred to as principal components (PCs), it also holds potential for feature selection due to its ability to maximize variance. PCA efficiently reduces dimensions by simplifying the dataset into fewer components through a linear transformation. Using PCA for features selection reduces the number of variables without requiring extensive search methods, such as wrapper-based approaches.

This paper is organized as follows: Section two examines pertinent literature, providing a concise overview of PCA applications. Section three details the approaches used in the study, and section four presents a comprehensive description of the proposed method. Section five explains and analyzes the experimental results, and section six concludes with insights and implications of the findings.

2. Related work

In related work, several studies have used PCA for feature selection or feature extraction. For instance, Rahul et al. [5] combined PCA with the Big Bang-Big Crunch (BBBC) optimization algorithm for dimensionality reduction and for optimizing feature selection. The PCA-BBBC method was tested using various plant disease image datasets. This integration was followed by an artificial neural network classifier, achieving a classification accuracy of 99.12%. Song et al. [6] applied PCA for feature selection, proposing a method that considers several eigenvectors, and uses a systematic scheme to perform feature selection. Experimental results on face recognition showed that their method significantly reduced the dimensionality of the original samples, without compromising recognition accuracy. PCA was used for feature extraction in the fruit recognition algorithm presented by Nareen et al. [7]. The system used support vector machines as a classifier, achieving a classification accuracy of around 75%.

Astha et al. [8] proposed a method to reduce emotional confusion and enhance emotion rates across multilingual datasets using PCA, support vector machines, and random forests for feature selection and classification. They evaluated this speech emotion recognition system on databases: Subcontinental Emotional Speech Corpus (SUBESCO), Emotional Voices (EMOVO), and Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS). The results of the recognition system in emotion categorization achieved an accuracy of around 88% for SUBESCO, 80% for EMOVO, and 74% for RAVDESS datasets. Priyanka et al. [9] investigated preprocessing, feature extraction, and selection phases for developing a classification model for ultrasound images of the kidney. They used PCA to reduce the number of features to an optimal subset. The accuracy of the image classification using artificial neural networks was around 78%. Usha and Neera [10] combined feature extraction and feature selection, the process, starting by extracting reliable features from normalized data, and evaluating the covariance matrix using rows and columns. The covariance matrix was further analyzed using eigen matrices. To reduce the feature sets, the stochastic gradient descent optimization method was employed, utilizing a fitness function that calculates the best score for these sets.

Potharlanka et al. [11] proposed a metaheuristic feature selection method that combines Particle Swarm Optimization (PSO), Firefly Algorithm (FA), and Whale Optimization Algorithm (WOA) within a Deep Q-Learning framework. Their approach adjusts feature significance dynamically based on feedback, making the selection process both adaptive and resilient over time. The collection of metaheuristic algorithms improves feature selection by integrating the advantages of PSO, FA, and WOA, each aiding in the identification of the most pertinent features. PSO manages the equilibrium between exploration and exploitation, FA maintains diversity in the search space, and WOA improves the exploration of feature landscapes. Deep Q-Learning optimizes the selection process by progressively refining feature significance according to model efficacy. This allows the framework to identify complex feature interactions that traditional static methods might miss, leading to more efficient and accurate feature selection across datasets. Their approach demonstrates substantial improvements in precision, accuracy, and recall over traditional feature selection methods, emphasizing the value of adaptive mechanisms in feature selection.

Deyanara and Wibowo [12] proposed a phishing protection technique that uses backpropagation algorithm based on the neural network method and PCA based on feature selection to reduce high-dimensional attributes into smaller set of attributes. They compared the performance of their techniques with and without using PCA, observing a marginal accuracy improvement when using PCA. Pournima and Pragnyaban [13] developed a feature reduction and selection approach for network threat detection, utilizing PCA for multivariate analysis. After the eigenvectors are calculated, they are sorted in descending order and the feature class mean value is calculated. This mean value is used to determine the threshold for feature selection.

Felipe and Tiago [14], proposed a Threshold Feature Selector (TFS), a new supervised dimensionality reduction method that uses class thresholds to identify the most relevant features. They also propose the Threshold PCA (TPCA), a combination of their supervised technique with standard PCA. In experiments across 10 datasets, TFS achieved higher accuracy in 90% of the reduced datasets compared to the original datasets. The second proposed technique, TPCA, outperformed the standard PCA in accuracy gain in 70% of the datasets. Osama et al. [15] propose a hybrid filter-wrapper method combining PCA as a filter to select an appropriate and informative subset of features and grey wolf optimizer as a wrapper approach to select further informative features. Logistic regression was used to evaluate the classification accuracy of the selected features. This method was used to classify Arabic news articles.

Jovanovic et al. [16] proposed a hybrid two-level framework where feature selection is critical in the initial stage. Their method integrates an improved version of the FA, named the Diversity Oriented Firefly Algorithm (DOFA), designed to overcome the original algorithm's limitations by improving population diversity and enhancing the search process in feature selection. The feature selection phase of their framework decreases the dimensionality of the phishing dataset by identifying the most relevant features prior to the hyperparameter tuning of the eXtreme Gradient Boosting (XGBoost) model. This two-tier approach improves both the accuracy and computational efficiency of the phishing detection process. The performance of DOFA-based feature selection was compared to other state-of-the-art metaheuristics, demonstrating superior results across multiple phishing datasets emphasizing its robustness in identifying key features while minimizing error.

3. Theoretical background

In this section, a brief description of the subjects and methods used in this study is provided.

3.1. Dimensionality reduction

The number of variables measured on each observation is called data dimensionality. Dimensionality reduction is defined as the mapping of data to a lower-dimensional space such that uninformative variance in the data is discarded, or such that a subspace in which the data lives is detected [17]. It is mainly used in data analysis, compression, and visualization [18]. Dimensionality reduction techniques have been performed either using linear methods or nonlinear methods, linear methods tend to be inadequate to complex nonlinear data, while nonlinear methods are better at handling such data. At the same time, nonlinear methods are weaker than linear methods when used with natural data

[17]. PCA is considered a linear method. Dimensionality reduction can be viewed as two primary categories: feature extraction and feature selection.

3.2. Feature extraction

Feature extraction methods reduce the number of features in a dataset by creating new features that capture most of the information from the original feature set. The reviewed studies identified two types of feature extraction techniques: statistical and optimization-based techniques [19]. This approach is particularly vital for high-dimensional datasets, as it mitigates the curse of dimensionality and enhances model efficiency and accuracy [2]. This process is instrumental in enhancing model performance, reducing computational complexity, and facilitating a deeper understanding of complex datasets.

3.3. Feature selection

Feature selection is a crucial step in machine learning. It involves selecting features from a set of features to improve model performance, interpretability, and efficiency. Essentially feature selection is about picking out and keeping the distinctive features while eliminating irrelevant or repetitive ones [20]. Feature selection methods are categorized into four types, filter, wrapper, embedded, and hybrid methods.

1. Filter methods assess feature relevance using measures or scoring functions independently of the machine learning algorithm. Examples include techniques like correlation analysis, squared tests, and information gain.
2. Wrapper methods evaluate sets of features by repeatedly training and testing the model [21]. These methods decide on including or excluding features based on their impact on model performance. Recursive Feature Elimination (RFE) [22] and forward/backward feature selection are examples.
3. Embedded Methods: involve feature selection embedded with the model training process. Techniques such as least absolute shrinkage and selection operator (LASSO) penalize features by giving them weights, effectively removing them during training [23].
4. Hybrid Method combines elements of filter, wrapper, and embedded methods to utilize their strengths effectively.

These techniques strive to find a balance, between being computationally efficient and delivering optimal model results.

3.4. Principal component analysis

PCA is a dimensionality reduction technique that projects the data onto a plane where each coordinate represents a data feature, then it transfers this data onto a new dimension where the variation is maximized [2].The transformation identifies the optimal component, represented as the best orthogonal line that aligns with the data. When referring to the “best line,” it implies the line that effectively minimizes the distance between the data points and itself.

In essence, this involves selecting the line that most effectively captures the variance in the data, resulting in a more meaningful representation. The goal is to minimize the distance between the observed data points and the chosen line, thereby enhancing the effectiveness of the transformation process [24].

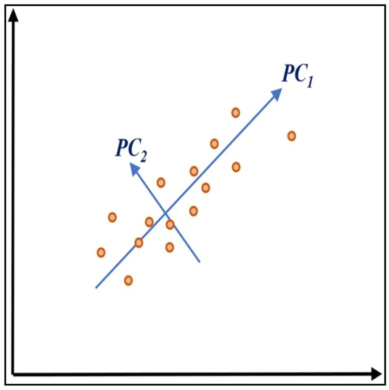

Fig. 1. PCA components [2].

The first optimal line is referred to as the first Principal Component (PC1). PC1 captures the maximum variation in the data. This variation is quantified by calculating the singular vector or eigenvector associated with PC1, divided by the sample size. Subsequently, the second-best component (PC2), is introduced. PC2 is positioned perpendicular to PC1, as illustrated in Fig. 1. Importantly, PC2 exhibits a lower variation ratio compared to PC1. The same pattern continues for subsequent components—PC3 is orthogonal to both PC1 and PC2 and so forth. This sequential progression results in principal components that are successively perpendicular to those preceding them, each capturing a diminishing level of variation in the data [25]. Every dataset can have several components up to the number of features in that dataset, with the first component representing the maximum variation component and the last component representing the least. The optimal number of components can be chosen to retain a high proportion of the variation in the data (more than 90% as an example) and transform the data using these components to a new dataset that contains only several features equal to the number of components that he chooses.

Some limitations and considerations have existed in PCA, and they can be listed as:

- PCA works best for datasets where the features are highly correlated.
- PCA is sensitive to the feature ranges, so it is preferable to normalize or standardize the dataset before applying PCA.
- Noisy features can negatively affect PCA as it tends to be biased toward noisy features, incorporating them into the first principal component.
- PCA components represent the proportional contribution of each feature that exists in the dataset; therefore, it is not preferable to use it as a feature selection technique.
- PCA can work with categorical and binary features after being coded but it is not ideal since these variables are not mapped to a coordinate plane.

Despite these limitations, PCA remains a valuable dimensionality reduction technique. To mitigate its drawbacks, it is recommended to complement PCA by integrating it with other techniques or preprocessing the dataset. This approach helps minimize the adverse effects of these limitations can be minimized, and a more robust and effective dimensionality reduction process can be achieved.

3.5. Evaluation metrics

The most effective way to assess a classification technique is through a confusion matrix, which consist of four values: True Positive (TP), representing the number of correctly predicted instances of the positive class; True Negative (TN), indicating the number of correctly predicted instances of the negative class; False Positive (FP), representing the number of instances incorrectly classified as positive when they are negative [22]; and False Negative (FN), reflecting the number of instances incorrectly classified as negative when they are actually positive. By determining these four numbers, one can compute the model's accuracy, which represents the percentage of correctly classified samples [26]. These metrics are calculated as follows:

$$Acc = \frac{TP + TN}{n} \times 100\% \tag{1}$$

Precision, which is how precisely a positive class was predicted by the model.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

The F1 score is performance measure used to evaluate the effectiveness of a classification algorithm, particularly in scenarios with imbalanced class distribution. The harmonic mean of precision and recall produces a single score that balances both measures [26].

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{3}$$

4. Proposed work

To identify the optimal set of features using the PCA technique, principal components are calculated by computing the covariance matrix of the dataset. Eigenvalues and eigenvectors are then divided from this matrix. PC1 is the eigenvector associated with the largest eigenvalue. These values represent the contributions of the features to this component, and the features that hold the higher values are the most influential in this component and can be considered the most significant features affecting this variance. Subsequent PCs contribute less to the variance but can still be effective in some cases.

To determine the best features, the initial set of components with a variance greater than 95% is selected. The variance percentage of each PC is multiplied by the Singular Value Decomposition (SVD) of every feature for that component. Finally, the results of these multiplication are sorted, and the features with highest values are identified as the best features to be selected by the system.

The diagrammatic representation of the proposed algorithm is shown in Fig. 2.

Algorithm 1 describes the proposed method step by step.

Algorithm 1. Proposed feature selection using PCA.

Input: Dataset, number of selected features (n).
Output: n selected best features.

Begin:
Step 1: Standardize the dataset to ensure a mean of 0 and standard deviation of 1.
Step 2: Calculate the covariance matrix of the standardized dataset.
Step 3: Perform eigen decomposition to derive eigenvalues and eigenvectors.
Step 4: Select the PCs that account for at least 95% of the variance.
Step 5: Compute the contributions of each feature to the selected PCs, weighted by the calculated variance.
Step 6: Aggregate the contributions of each feature across all selected PCs.
Step 7: Sort the features based on their aggregated contributions and select the top n-unique features.
End.

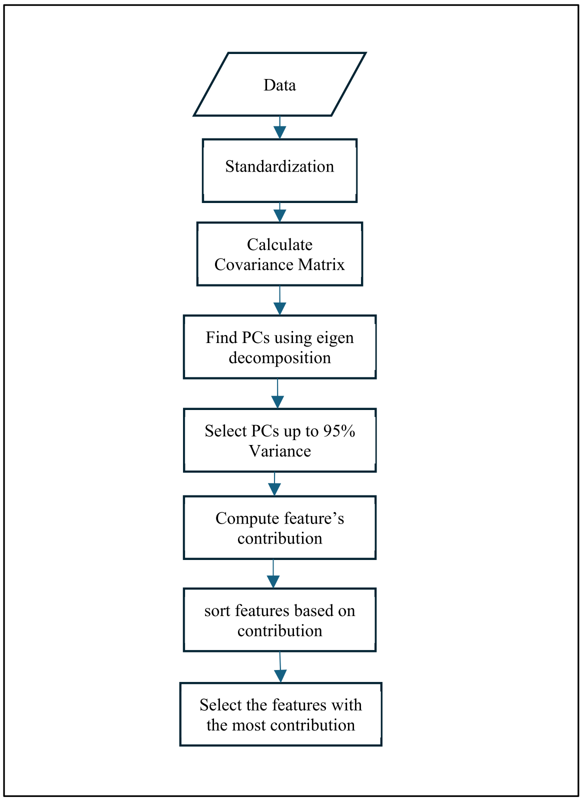

Fig. 2. Diagrammatic representation of the proposed work.

The initial step of the work is to standardize the data since PCA is sensitive to the variances of the original variables, Standardization involves computing the mean and the standard deviation for each feature individually. Then, each value of the feature is standardized by subtracting the mean of that feature and dividing the result by the standard deviation of that feature. Subsequently, the covariance matrix is computed using the standardized values.

After computing the covariance matrix, eigenvalues and eigenvectors are computed from the covariance matrix. Eigenvalues represent the amount of variance explained by each principal component, while eigenvectors indicate the direction of the PCs in the original feature space. After computing the eigenvalues and eigenvectors, these values are sorted in descending order of the eigenvalues. After sorting the eigenvalues and their corresponding eigenvectors, the first set of components with variance higher than 95% are selected.

The final step involves multiplying the SVD of each feature by the variance ratio of each PC and sorting the results to determine the order of the contribution for the features. Then, the required number of features is selected from the sorted list.

5. Results and discussion

For the experimental setup, a standard laptop with an Intel Core i5 processor, 8 GB RAM, Windows 11 64-bit, and Jupyter Notebook running Python 3.11 was used, utilizing the SKLearn libraries for the required modules. This work was applied to and tested on two datasets: NSL-KDD [27, 28], and Mushroom [29], and six classifiers: Decision Tree (DT) [30], Naive Bayes (NB) [31], Logistic Regression (LR) [32], K-Neighbors Classifier

(KN) [33], XGBoost [16], and Adaptive Boosting (AdaBoost). First, PCA was used as the standard feature extraction technique to generate different sets of features, which were then compared with the performance of the proposed method that selected the best set of features from the original feature pool.

The NSL-KDD dataset is an improved version of the Knowledge Discovery and Data Mining (KDD'99) dataset for network-based intrusion detection systems. It addresses some of inherent problems of the KDD'99 dataset, such as redundant records, that skew the learning and evaluation of intrusion detection models. NSL-KDD provides a subset of network connections that similar to real-world data, making it a more effective benchmark for researchers and practitioners in the field of cybersecurity. Additionally, NSL-KDD includes a variety of labeled attack types along with normal connections, facilitating the development and evaluation of machine learning models for anomaly and misuse detection in network security. The dataset is widely used in academic research and practical applications to enhance the performance and accuracy of intrusion detection systems [27]. The proposed feature selection method was tested on the first dataset, NSL-KDD, and the results in Table 1 were obtained.

In Table 2, PCA was used as a feature extraction technique on NSL-KDD.

The results presented in Tables 1 and 2 illustrate that the accuracy score of the two methods is very close implying that PCA performed well in both cases. When applying these methods to the NSL-KDD dataset, it becomes evident that neither approach demonstrates a clear advantage over the other, the comparison is clearly shown in Fig. 3.

This observation supports the notion that PCA can be effectively and reliably used as a feature selection technique.

Feature selection was tested on the second dataset, Mushroom, and the following results presented in Table 3 were obtained.

In Table 4, PCA is used as a feature extraction technique on the Mushroom dataset.

The Mushroom dataset is a classic example of a binary classification problem with a clear real-world implication, distinguishing between edible and poisonous mushrooms. This practical aspect, combined with its complexity and categorical nature, makes it a popular choice for educational and demonstration purposes in machine learning.

As illustrated in Tables 3 and 4, feature selection using PCA performed as well as using PCA as feature extraction, and in some cases, it outperformed feature extraction, demonstrating that this method provides better results than those obtained when using PCA as feature extraction techniques. The comparison is clear as shown in Fig. 4.

Despite the low classification score, this outcome can be attributed to the dataset not undergoing preprocessing techniques and the use of basic classifiers in the field.

Key observations include the method's effectiveness in reducing dimensionality without a substantial loss in accuracy, making it suitable for real-world applications where computational efficiency and model simplicity are critical. Additionally, the method offers flexibility in selecting the optimal number of features, balancing performance and complexity. However, the limitations of this method can be summarized as follows:

1. PCA requires the dataset to be standardized or normalized as it is sensitive to the scales of the features and the results can be skewed if not well processed.
2. PCA cannot work with binary or categorical features as these types of features do not map well onto a coordinate plane, which requires a method to convert these values into numerical ones.
3. PCA works best on datasets where features are correlated, and it will not be effective if the features are independent.
4. Noisy features can distort the feature selection process as PCA tends to give more weight to noisy features, thus, noise removal is required before using PCA.

Table 1. Classification results using PCA as feature selection and NSL-KDD dataset.

Selected features	Classifier	Accuracy	Precision	F1 score
1 feature	DT	80.88%	82.57%	0.8088
	KN	81.29%	83.28%	0.8129
	NB	43.07%	42.96%	0.3012
	LR	44.53%	39.06%	0.4028
	XGBoost	84.94%	95.53%	0.8537
	AdaBoost	82.87%	95.52%	0.8298
2 features	DT	76.35%	80.22%	0.7626
	KN	81.45%	83.57%	0.8145
	NB	43.02%	31.52%	0.3011
	LR	48.78%	40.54%	0.4265
	XGBoost	80.76%	95.96%	0.8035
	AdaBoost	76.89%	96.48%	0.7524
5 features	DT	78.17%	81.38%	0.7812
	KN	79.49%	82.35%	0.7946
	NB	45.03%	68.79%	0.3403
	LR	66.85%	48.47%	0.5286
	XGBoost	80.32%	96.01%	0. 7980
	AdaBoost	79.27%	95.21%	0.7862
10 features	DT	79.49%	82.71%	0.7946
	KN	77.65%	81.27%	0.7758
	NB	45.03%	68.79%	0.3403
	LR	61.95%	49.39%	0.5188
	XGBoost	78.08%	96.97%	0.7673
	AdaBoost	77.52%	96.52%	0.7606
15 features	DT	79.41%	82.66%	0.7937
	KN	77.67%	81.28%	0.7760
	NB	45.03%	68.79%	0.3403
	LR	61.99%	49.42%	0.5198
	XGBoost	76.86%	96.78%	0.7513
	AdaBoost	77.52%	96.52%	0.7606
20 features	DT	79.81%	82.88%	0.7978
	KN	77.67%	81.28%	0.7760
	NB	45.03%	68.79%	0.3403
	LR	71.95%	49.39%	0.5188
	XGBoost	77.06%	96.80%	0.7540
	AdaBoost	77.40%	96.37%	0.7595
30 features	DT	81.55%	83.79%	0.8155
	KN	78.55%	81.84%	0.7851
	NB	45.03%	68.79%	0.3403
	LR	70.77%	63.87%	0.6206
	XGBoost	80.17%	96.66%	0.7949
	AdaBoost	77.75%	96.50%	0.7639

Table 2. Classification results using PCA as feature extraction and NSL-KDD dataset.

Extracted features	Classifier	Accuracy	Precision	F1 score
1 feature	DT	76.64%	80.57%	0.7655
	KN	76.10%	80.29%	0.7598
	NB	43.07%	42.96%	0.3012
	LR	47.40%	71.29%	0.3823
	XGBoost	80.57%	95.42%	0.8022
	AdaBoost	82.05%	95.45%	0.8201
2 features	DT	78.11%	81.29%	0.7806
	KN	80.67%	83.11%	0.8066
	NB	43.02%	31.52%	0.3011
	LR	44.55%	63.86%	0.3341
	XGBoost	80.32%	96.02%	0.7980
	AdaBoost	76.91%	96.44%	0.7527
5 features	DT	77.08%	81.10%	0.7699
	KN	77.08%	81.25%	0.7698
	NB	45.04%	68.88%	0.3403
	LR	72.21%	75.09%	0.7216
	XGBoost	78.86%	96.75%	0.7780
	AdaBoost	77.13%	96.40%	0.7558
10 features	DT	80.45%	83.22%	0.8043
	KN	77.26%	81.24%	0.7717
	NB	45.04%	68.88%	0.3403
	LR	73.23%	75.01%	0.7322
	XGBoost	81.89%	96.57%	0.8164
	AdaBoost	77.47%	96.70%	0.7597
15 features	DT	83.85%	85.38%	0.8385
	KN	77.25%	81.24%	0.7716
	NB	45.04%	68.88%	0.3403
	LR	73.23%	75.02%	0.7323
	XGBoost	81.71%	96.31%	0.8145
	AdaBoost	81.08%	97.05%	0.8056
20 features	DT	79.93%	82.87%	0.7991
	KN	77.26%	81.24%	0.7717
	NB	45.04%	68.88%	0.3403
	LR	73.25%	75.03%	0.7325
	XGBoost	80.00%	96.69%	0.7927
	AdaBoost	80.96%	97.16%	0.8039
30 features	DT	79.32%	82.51%	0.7928
	KN	77.28%	81.25%	0.7719
	NB	45.04%	68.88%	0.3403
	LR	73.18%	74.98%	0.7318
	XGBoost	80.86%	96.77%	0.8033
	AdaBoost	80.99%	97.09%	0.8044

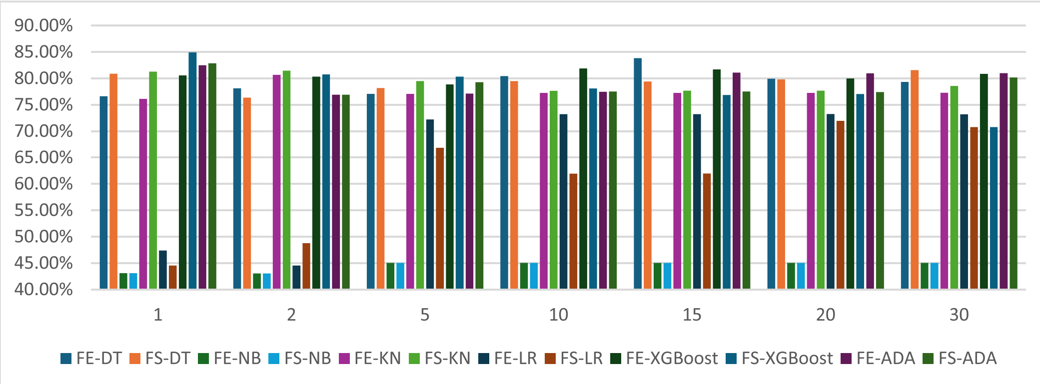

Fig. 3. Accuracy comparison between feature extraction and feature selection using the proposed method and NSL-KDD dataset.

Table 3. Classification results using PCA as feature selection and Mushroom dataset.

Selected features	Classifier	Accuracy	Precision	F1 score
1 feature	DT	51.38%	54.82%	0.4548
	KN	51.38%	54.82%	0.4548
	NB	48.62%	71.03%	0.5249
	LR	51.06%	78.83%	0.6779
	XGBoost	87.08%	85.57%	0.8676
	AdaBoost	86.22%	84.11%	0.8600
2 features	DT	49.72%	29.31%	0.2926
	KN	49.88%	28.46%	0.2842
	NB	48.59%	68.71%	0.5042
	LR	51.20%	78.97%	0.6429
	XGBoost	88.98%	89.11%	0.8847
	AdaBoost	86.22%	84.11%	0.8600
5 features	DT	50.37%	55.09%	0.5463
	KN	48.49%	35.69%	0.3370
	NB	48.13%	24.03%	0.3243
	LR	49.14%	61.30%	0.5617
	XGBoost	100.00%	100.00%	1
	AdaBoost	99.75%	100.00%	0.9974
10 features	DT	50.56%	56.65%	0.5535
	KN	48.18%	32.51%	0.3247
	NB	49.80%	51.42%	0.5114
	LR	49.36%	47.97%	0.4633
	XGBoost	100.00%	100.00%	1
	AdaBoost	99.75%	100.00%	0.9974
15 features	DT	50.53%	61.95%	0.6054
	KN	48.22%	28.26%	0.3191
	NB	51.86%	25.85%	0.3400
	LR	50.01%	38.76%	0.3874
	XGBoost	100.00%	100.00%	1
	AdaBoost	99.88%	100.00%	0.9987
20 features	DT	50.34%	64.24%	0.6370
	KN	48.22%	28.53%	0.3197
	NB	51.87%	25.86%	0.3402
	LR	49.83%	36.43%	0.3632
	XGBoost	100.00%	100.00%	1
	AdaBoost	99.88%	100.00%	0.9987

Table 4. Classification results using PCA as feature extraction and Mushroom dataset.

Extracted features	Classifier	Accuracy	Precision	F1 score
1 feature	DT	50.14%	67.65%	0.6759
	KN	50.30%	72.59%	0.7203
	NB	50.88%	69.78%	0.6405
	LR	50.59%	61.68%	0.5986
	XGBoost	73.60%	81.80%	0.6791
	AdaBoost	71.51%	97.05%	0.5870
2 features	DT	50.04%	95.00%	0.9501
	KN	50.02%	96.04%	0.9606
	NB	50.18%	77.30%	0.7713
	LR	50.03%	75.30%	0.7531
	XGBoost	94.34%	94.12%	0.9412
	AdaBoost	84.92%	87.34%	0.8368
5 features	DT	50.05%	98.45%	0.9846
	KN	50.04%	99.12%	0.9914
	NB	50.26%	83.00%	0.8251
	LR	50.20%	83.37%	0.8311
	XGBoost	99.32%	99.49%	0.9930
	AdaBoost	91.14%	93.58%	0.9049
10 features	DT	50.06%	98.45%	0.9846
	KN	50.07%	99.13%	0.9914
	NB	50.35%	85.36%	0.8430
	LR	50.21%	83.23%	0.8292
	XGBoost	99.75%	99.49%	0.9974
	AdaBoost	92.62%	94.85%	0.9211
15 features	DT	50.08%	99.64%	0.9963
	KN	50.05%	99.49%	0.9951
	NB	50.20%	91.14%	0.9085
	LR	50.13%	87.50%	0.8741
	XGBoost	100.00%	100.00%	1
	AdaBoost	98.03%	98.96%	0.9793
20 features	DT	50.07%	99.69%	0.9969
	KN	50.05%	99.49%	0.9951
	NB	50.02%	86.70%	0.8670
	LR	50.03%	95.18%	0.9520
	XGBoost	100.00%	100.00%	1
	AdaBoost	98.58%	98.97%	0.9852

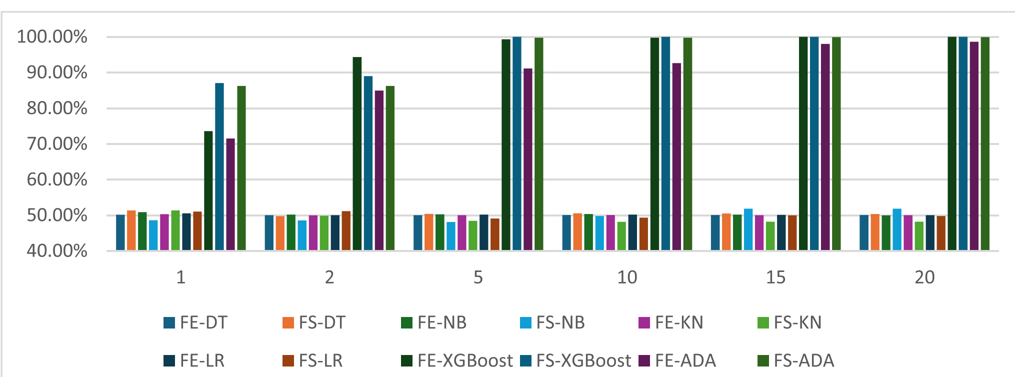

Fig. 4. Accuracy comparison between feature extraction and feature selection using the proposed method and mushroom dataset.

6. Conclusion

PCA is a powerful dimensionality reduction technique widely used in many fields. Traditionally, PCA has been applied for feature extraction, transforming original features into uncorrelated principal components. However, this work demonstrates that PCA can also be effectively employed as a feature selection method, providing significant benefits in some scenarios. The proposed method leverages PCA to identify and select the most influential features based on their contribution to the PC variance. By focusing on feature selection rather than extraction, the approach preserves the interpretability of the original data while enhancing model performance. Experimental results show that the proposed PCA-based feature selection method achieves comparable or superior accuracy and precision compared to using PCA for feature extraction, particularly when tested on different classifiers across multiple datasets. In conclusion, PCA not only remains a valuable tool for feature extraction but can also serve as a robust and interpretable feature selection method. This work demonstrates that in certain cases, feature selection using PCA can outperform feature extraction, offering a practical alternative for dimensionality reduction in machine learning tasks.

Future work could explore combining PCA with other methods to mitigate its limitations and further enhance its performance on more complex datasets. A hybrid approach could be employed where PCA is used for feature extraction on a subset of features, also selected using PCA, to ensure that weaker features do not negatively impact the extraction process. The listed drawbacks suggest that the proposed method can be further enhanced by integrating additional techniques, such as noise filtering or hybrid feature selection methods.

Acknowledgment

The authors wish to express their appreciation for the opportunities and resources that contributed to the completion of this research.

Conflicts of interest

The authors declare no conflicts of interest.

Authors' contribution

All authors contributed equally to the preparation of this article.

Data availability

The datasets used in this study are publicly available.

References

1. R. E. Nogales and M. E. Benalcázar, "Analysis and evaluation of feature selection and feature extraction methods," *International Journal of Computational Intelligence Systems*, vol. 16, no. 1, Art. no. 153, Sep. 2023, doi: 10.1007/s44196-023-00319-1.

2. M. Greenacre, P. J. F. Groenen, T. Hastie, A. I. D'Enza, A. Markos, and E. Tuzhilina, "Principal component analysis," *Nature Reviews Methods Primers*, vol. 2, Art. no. 100, Dec. 2022, doi: 10.1038/s43586-022-00184-w.
3. T. Kurita, "Principal component analysis (PCA)," in *Computer Vision*, Cham: Springer, pp. 1013–1016, 2021.
4. Z. M. Hira and D. F. Gillies, "A review of feature selection and feature extraction methods applied on microarray data," *Advances in Bioinformatics*, vol. 2015, Art. no. 198363, June 2015, doi: 0.1155/2015/198363.
5. R. Sharma and A. Singh, "VGG16 feature selection using PCA-big bang big algorithm," *Journal of Intelligent & Fuzzy Systems*, vol. 45, no. 1, pp. 1437–1451, July 2023, doi: 10.3233/JIFS-222517.
6. F. Song, Z. Guo, and D. Mei, "Feature selection using principal component analysis," in Proc. IEEE 2010 Int. Conf. on System Science, Engineering Design and Manufacturing Informatization, Yichang, China, pp. 27–30, doi: 10.1109/ICSEM.2010.14.
7. N. O. M. Salim and A. K. Mohammed, "Fruit recognition using Statistical and Features extraction by PCA," *Academic Journal of Nawroz University*, vol. 12, no. 3, pp. 566–574, 2023, doi: 10.25007/ajnu.v12n3a1687.
8. A. Tripathi and P. Rani, "PCA-based feature selection and hybrid classification model for speech emotion recognition," in Proc. 6th Int. Conf. on Innovative Computing and Communications (ICICC), Delhi, India, pp. 347–353, 17–18 Feb. 2023, doi: 10.1007/978-981-99-3315-0_26.
9. P. and D. Kumar, "Feature extraction and selection of kidney ultrasound images using GLCM and PCA," *Procedia Computer Science*, vol. 167, pp. 1722–1731, 2020, doi: 10.1016/j.procs.2020.03.382.
10. U. Devi and N. Batra, "An optimized stochastic PCA feature selection method to enhance the prediction of credit risk," in *Proc. 6th Int. Conf. on Innovative Computing and Communications (ICICC)*, Delhi, India, pp. 391–399, 17–18 Feb. 2023, doi: 10.1007/978-981-99-3315-0_30.
11. J. L. Potharlanka and B. M. N., "Feature importance feedback with deep Q process in ensemble-based metaheuristic feature selection algorithms," *Scientific Reports*, vol. 14, no. 1, Art. no. 2923, Feb. 2024, doi: 10.1038/s41598-024-53141-w.
12. D. Tuapattinaya and A. Wibowo, "Phishing Website detection using neural network and PCA based on feature selection," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, no. 6, pp. 1150–1152, March 2020, doi: 10.35940/ijrte.D4532.038620.
13. P. More and P. Mishra, "Enhanced-PCA based dimensionality reduction and feature selection for real-time network threat detection," Engineering, *Technology & Applied Science Research*, vol. 10, no. 5, pp. 6270–6275, October 2020, doi: 10.48084/etasr.3801.
14. F. D. M. Battisti and T. B. A. D. Carvalho, "Threshold feature selection PCA," in *Proc. 10th Symp. on Knowledge Discovery, Mining and Learning (KDMiLe 2022)*, Campinas, SP, Brazil, pp. 50–57, Nov. 2022, doi: 10.5753/kdmile.2022.227718.
15. O. A. Alomari *et al.*, "Hybrid feature selection based on principal component analysis and grey wolf optimizer algorithm for arabic news article classification," *IEEE Access*, vol. 10, pp. 121816–121830, 2022, doi: 10.1109/ACCESS.2022.3222516.
16. L. Jovanovic *et al.*, "Improving phishing website detection using a hybrid two-level framework for feature selection and XGBoost tuning," *Journal of Web Engineering*, vol. 22, no. 3, pp. 543–574, May 2023, doi: 10.13052/jwe1540-9589.2237.
17. V. S. Sumithra and S. Surendran, "A review of various linear and non linear dimensionality reduction techniques," *International Journal of Computer Science and Information Technologies*, vol. 6, no. 3, pp. 2354–2360, 2015.
18. A. L. Grace and M. Thenmozhi, "Survey of popular linear dimensionality reduction techniques," in Proc. of Int. Conf. on Deep Learning, Computing and Intelligence, Singapore, pp. 593–603, April 2022, doi: 10.1007/978-981-16-5652-1_53.
19. H. H. Htun, M. Biehl, and N. Petkov, "Survey of feature selection and extraction techniques for stock market prediction," *Financial Innovation*, vol. 9, no. 1, Art. no. 26, Jan 2023, doi: 10.1186/s40854-022-00441-7.
20. A. Al-zubidi, N. F. AL-Bakri, R. K. Hasoun, S. H. Hashim, and H. T. S. Alrikabi, "Mobile application to detect Covid-19 pandemic by using classification techniques: Proposed system," *International Journal of Interactive Mobile Technologies*, vol. 15, no. 16, pp. 34–51, 2021, doi: 10.3991/ijim.v15i16.24195.
21. Y. Li, T. Li, and H. Liu, "Recent advances in feature selection and its applications," *Knowledge and Information Systems*, vol. 53, no. 3, pp. 551–577, May 2017, doi: 10.1007/s10115-017-1059-8.
22. D. N. Mhawi, A. Aldallal, and S. Hassan, "Advanced feature-selection-based hybrid ensemble learning algorithms for network intrusion detection systems," *Symmetry*, vol. 14, no. 7, Art. no. 1461, July 2022, doi: 10.3390/sym14071461.
23. D. N. Mhawi and S. H. Hashem, "Proposed hybrid correlation feature selection forest panalized attribute approach to advance IDSs," *Karbala International Journal of Modern Science*, vol. 7, no. 4, Art. no. 15, 2021, doi: 10.33640/2405-609X.3166.

24. M. J. Kelain and A. Hussein, “Enhancement principal component analysis for features extract in image processing,” *Journal of Advanced Research in Dynamical & Control Systems*, vol. 10, no. 11, pp. 455–460, Oct. 2018.
25. H. M. Radha, A. K. AbdulHassan, and A. H. Al-Timemy, “Classification of different shoulder girdle motions for prosthesis control using a time-domain feature extraction technique,” *ARO-The Scientific Journal of Koya University*, vol. 10, no. 2, pp. 73–81, 2022, doi: 10.14500/aro.11064.
26. T. S. Othman and S. M. Abdullah, “An intelligent intrusion detection system for internet of things attack detection and identification using machine learning,” *ARO-The Scientific Journal of Koya University*, vol. 11, no. 1, pp. 126–137, 2023, doi: 10.14500/aro.11124.
27. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in *Proc. IEEE 2009 Symp. on Computational Intelligence for Security and Defense Applications*, Ottawa, ON, Canada, pp. 1–6, doi: 10.1109/CISDA.2009.5356528.
28. B. Mohammed and E. K. Gbashi, “Intrusion detection system for NSL-KDD dataset based on deep learning and recursive feature elimination,” *Engineering and Technology Journal*, vol. 39, no. 7, pp. 1069–1079, 2021, doi: 10.30684/etj.v39i7.1695.
29. “Mushroom,” *UCI Machine Learning Repository*, 1981. [Online]. Available: https://doi.org/10.24432/C5959T.
30. J. Xu, “Systematic analysis and application prospect of decision tree,” *Highlights in Science, Engineering and Technology*, vol. 71, pp. 163–170, Nov. 2023, doi: 10.54097/hset.v71i.12687.
31. M. Schonlau, “The Naive Bayes classifier,” in *Applied Statistical Learning*, Cham: Springer, pp. 143–160, 2023.
32. S. Chalichalamala, N. Govindan, and R. Kasarapu, “Logistic regression ensemble classifier for intrusion detection system in internet of things,” *Sensors*, vol. 23, no. 23, Art. no. 9583, Dec. 2023, doi: 10.3390/s23239583.
33. G. N. Ahmad, H. Fatima, N. L. Shafiullah, and H. Akhter, “Optimal medical diagnosis of human heart disease by k-nearest neighbors and decision trees classifiers algorithms,” In: *Artificial Intelligence and Communication Technologies*, Saroj Hiranwal and Garima Mathur, Eds, India, SCRS, pp. 1031–1037, 2023. https://doi.org/10.52458/978-81-955020-5-9-98.