An Overview of Principal Component Analysis: Theory and Applications

Blaine Swieder
Department of Computer Science
College of Engineering
Tennessee Technological University
Cookeville, Tennessee 38505
Email: bcswieder42@tntech.edu

***Abstract*—Principal Component Analysis (PCA), a fundamental technique in data science, machine learning, and statistics, is widely used for dimensionality reduction and feature extraction in high-dimensional datasets. By transforming the original variables into a set of orthogonal components, PCA captures the directions of maximum variance, allowing for efficient data representation with minimal loss of information. We aim to provide a comprehensive survey of PCA, exploring its theoretical mathematical foundations, computational methodology, as well as practical applications across the literature.**

First, we begin by outlining the mathematical principles that consist of the underpinnings of PCA, which include eigenvalue decomposition and singular value decomposition (SVD). The computational steps for implementing PCA are discussed, from data preprocessing to identifying principal components and reconstructing approximations of the original data. Techniques for evaluating the effectiveness of PCA, such as explained variance ratios and scree plots, are also discussed.

We then examine the diverse applications of PCA in fields such as image processing, bioinformatics, and finance. In image processing, PCA reduces the dimensionality of pixel data, facilitating compression and pattern recognition. In bioinformatics, PCA helps analyze complex gene expression datasets, revealing patterns linked to biological processes. In finance, PCA identifies principal drivers of market trends, improving portfolio management and risk assessment. In this section, we will analyze a selection of papers and discuss their application of PCA to show the versatility of PCA.

***Index Terms*—Principal Component Analysis, Dimensionality Reduction, Feature Extraction, Data Science, Machine Learning**

1. Introduction

Principal Component Analysis (PCA) is an essential technique in data analysis, usually used for dimensionality reduction and feature extraction across the literature [6], [7]. In this era of big data, where it is normal to have high-dimensional datasets, PCA provides a robust mathematical framework for simplifying data representation without significant loss of information. By projecting data onto orthogonal axes that capture the maximum variance, Principal Component Analysis (PCA) reduces redundancy while still maintaining the essential patterns [8].

The mathematical foundations of PCA lie within linear algebra and statistical principles; more specifically, eigenvalue decomposition and covariance analysis [7] . This versatility and computational efficiency make PCA an asset in fields such as machine learning, image processing, bioinformatics, and finance [2], [8]. For example, PCA is widely used to extract meaningful features in image recognition taks, analyze gene expression data in biological research, as well as identifying key drivers of financial markets [1], [4].

Despite its widespread application, Principal Component Analysis (PCA) has its limitations. More specifically, its linearity assumption might be unable to capture complex, nonlinear relationships in data, as well as its sensitivity to scaling and outliers may and can impact results [8], [9]. On the contrary, some advancements such as Kernel PCA and Robust PCA have been developed to extend its applicability to address these problems [9], [10].

The goal of this survey is to provide an overview of PCA in which we explore its theoretical foundations, computational methodologies, and its diverse applications. First, we start by discussing the mathematical principles that underpin PCA, which includes its reliance on eigenvalues, eigenvectors, and singular value decomposition (SVD) [6], [7]. Next, we briefly outline the computational steps that are involved in the implementation of PCA, from data preprocessing to component selection and data transformation [2], [11]. Finally, we will examine a small selection of papers that involve the real-world applications of PCA, which highlights its roles in addressing challenges across multiple disciplines and across the literature [1], [4], [5].

By combing both theory and practice, this survey aims to discuss to explain the profound influence of PCA in data science, machine learning, and other disciplines [8].

2. Theoretical Foundations

Before we consider PCA in a computational and applied context, we must first establish the mathematical intuition behind Principal Component Analysis (PCA).

2.1. Mathematical Basis

Principal Component Analysis (PCA) has mathematical foundations that lie within linear algebra and statistical principles, which rely on mathematical constructs such as variance, covariance, eigenvalues, and eigenvectors. At its core, PCA aims to find new axes or *principal components* that maximize the variance in the data. The process of PCA starts by computing the covariance matrix of the data, which quantifies the relationship between different variables. This step is given by the following mathematical formula:

$$\mathrm{Cov}(X) = \frac{1}{n-1} X_{\text{centered}}^T X_{\text{centered}}.$$

Here, $X_{centered} = X - \mu$, where μ is the mean vector of the dataset [6].

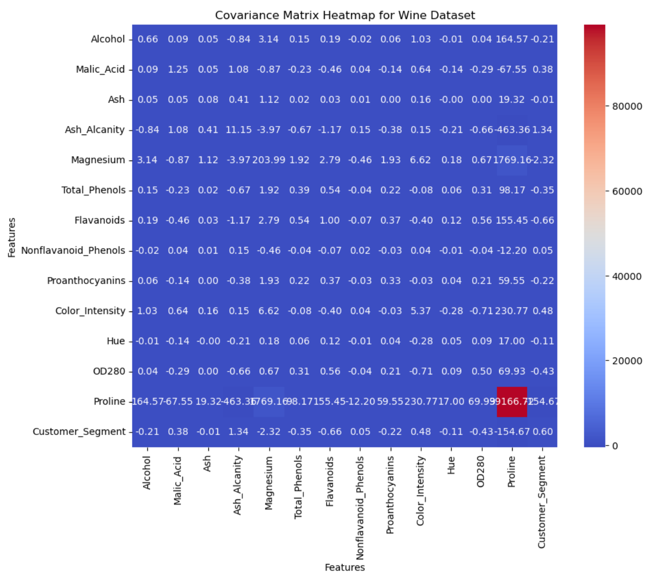

Figure 1. Here is the covariance heatmap for the Wine.csv dataset. This heatmap demonstrates the relationship between different features within the dataset [11].

Here, $X_{centered} = X - \mu$, where μ is the mean vector of the dataset [6]. Next, the covariance matrix serves as the basis for identifying principal components, which are obtained by solving the following eigenvalue equation:

$$\mathrm{Cov}(X)v = \lambda v$$

Here v denotes the eigenvector (direction of maximum variance) and λ denotes the corresponding eigenvalues (magtitude of variance) [7]. Singular Value Decomposition provides an alternative computational approach to obtain the same components, especially for large datasets. Here, we use the following formula:

$$X = U\Sigma V^T$$

where V denotes and contains the principal components, Σ contains singular values which pertain to eigenvalues [6]. To conclude, these mathematical tools form the foundation for dimensionality reduction, which enables PCA to distill complex datasets into simpler representations.

2.2. Core Concepts

The principal components that are identified by Principal Component Analysis (PCA) are orthogonal axes that represent the directions of maximum variance in the data. The first principal component captures the highest variance, which is then following by subsequent components that capture progressively less variance, while the remaining ones are orthogonal to the previous ones [8]. This orthogonality ensures that the principal components are uncorrelated, which simplify the analysis and interpretation of the data. The variance explained by each component is proportional to its eigenvalue; moreover, its cumulative variance explains how much information is retained when reducing dimensions. These concepts highlight the ability of Principal Component Analysis (PCA) to reduce redundancy and reveal the underlying patterns in high-dimensional datasets.

Next, let us see how this works via the mathematical formulae. The variance is explained by each component is proportional to its eigenvalue:

$$\text{Explained Variance Ratio} = \frac{\lambda_i}{\sum \lambda}$$

On the other hand, the cumulative variance explains how much total information is retained when reducing dimensions. This equation is given by:

$$\text{Cumulative Variance} = \sum_{i=1}^{k} \frac{\lambda_i}{\sum \lambda}$$

These two equations demonstrates PCA's ability to reduce redundancy and reveal patterns in high-dimensional datasets, hence making it a useful tool in exploratory data analysis [2].

2.3. Assumptions and Limitations

Principal Component Analysis (PCA) relies upon several assumptions, such as linearity, which assumes that the relationships between variables can be captured by straight lines or planes [8]. Moreover, it also assumes that the data is continuous and Gaussian-distributed, as this ensures that the principal components reflect meaningful patterns. On the contrary, PCA is sensitive to scaling, which variables with larger ranges can dominate the variance unless the data is standardized [7].

Non-linear relationships are a limitation, as PCA cannot capture complex patterns beyond its linear framework. As a result, techniques such as Kernel PCA are devised to solve this problem by applying the kernel trick to map data into higher-dimensional spaces before computing principal components [10]. Moreover, outliers can distort principal components, hence reducing interpretability. Robust PCA techniques, like Sparse PCA, solve this issues by identifying and isolating outlier effects [10].

3. Computational Methodology

In this section, we aim to show how to think of Principal Component Analysis (PCA) in a computational context.

3.1. Data Preprocessing

Data Preprocessing is essential step in Principal Component Analysis (PCA) to ensure that the results are meaningful and interpretable. The first step involves centering the data by subracting the mean of each feature:

$$X_{centered} = X - \mu$$

where μ is the mean vector [6]. Next, standardization is crucial when features have different scales, as it ensures that all variables contribute equally. This is shown by the following equation:

$$X_{standardized} = \frac{X - \mu}{\sigma}$$

Here, σ is the standard deviation of each feature [2], [11]. This crucial step prevents variables with large magnitudes from dominating the computation of principal components.

3.2. PCA Implementation Process

The Principal Component Analysis (PCA) process involves several computational steps:

1) **Compute the Covariance Matrix**: This matrix represents the relationships between variables and serves as the basic for identifying principal components [7].
2) **Perform Eigen Decomposition**: Solve the eigenvalue equation

$$\text{Cov}(x)v = \lambda v$$

to obtain the eigenvectors (principal components) and eigenvalues (variance explained by each component) [3].
3) **Sort Eigenvalues and Eigenvectors**: Rank the eigenvectors by their eigenvalues in descending order and this prioritization determines which components contribute the most to the variance.
4) **Select Principal Components**: Choose the top k components based on a desired threshold variance explained (such as 95%) [4].
5) **Transform Data**: Project the original data onto the selected principals components to obtain the reduced-dimensional representation. This transformation then simplifies the data while retaining its most significant features. The following equation is used to transform the data:

$$X_{reduced} = X_{centered} V_k$$

By following these steps, one successfully computes Principal Component Analysis.

3.3. Performance Evaluation Techniques

We can evaluate the performance of PCA by assessing how well the reduced representation retains the original data's information. A common metric is the *explained variance ratio*, which measures the proportion of total variance that is captured by each principal component [7]. This ratio allows us to determine the optimal number of components to retain. Another method are *scree plots*, which plot the eigenvalues against the number of components to identify *elbow points*, where additional components contribute minimal variance [2]. Moreover, the reconstruction error, which is calculated by comparing the original data to its reconstruction from the reduced representation [4]. Hence, this provides a quantitative measure of PCA's effectiveness in dimensionality reduction. In the next section, we will explore the application of PCA across the literature.

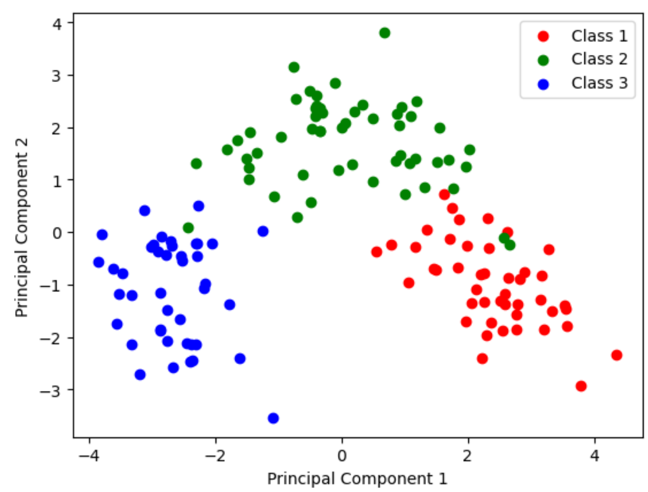

Figure 2. The PCA projection of the Wine dataset onto the first two principal components demonstrates class separability [11].

We use Figure 2 to demonstrate the results of applying Principal Component Analysis (PCA) to the Wine dataset [11]. The given data is projected onto the first two principal components, which captures the majority of the variance, which demonstrates the separation between classes in the reduced-dimensional space. This image is intended to give the reader an idea of how it looks generally to apply PCA via a figure generated by code. In the next section, we will delve further into a specific example to elaborate further.

3.4. Example

In this section, we will apply the Wine dataset to show how PCA works computationally, as well as mathematically. The purpose of this section is to provide the reader with an insight on the how and why of PCA [11].

3.4.1. Application in CS. This section covers how this is modeled and computed via Python, and the notebook is proved with the paper for further clarity.

TABLE 1. EXPLAINED VARIANCE BY PRINCIPAL COMPONENTS

Principal Component	**Explained Variance Ratio (%)**
PC1	36.2
PC2	19.2

To demonstrate the PCA process, we utilized the Wine dataset [11], containing the chemical properties of wines classified into three categories. After standardizing the features, we applied PCA and projected the data onto the first two principal components. Figure 3 shows the separation of wine classes in the reduced-dimensional space.

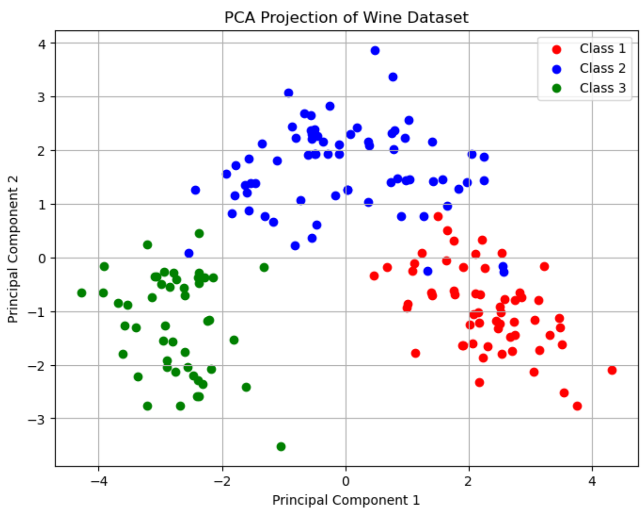

Figure 3. Projection of the Wine dataset onto the first two principal components.

We found that the first two principal components explain approximately 55.4% of the variance, as shown in Table 1. We used this example to show PCA's ability to reduce dimensionality while preserving the underlying structure of the data.

3.4.2. Application in Mathematics and Statistics. In this section, we will build upon the steps that were presented in PCA code provided by approaching the same Wine dataset from a mathematical perspective. We will proceed with the following steps applied onto the dataset [11]:

1) **Covariance Matrix Calculation**: As mentioned in the Theoretical Foundations section, the covariance matrix quantifies the relationships between our given variables. For our Wine Dataset example, we calculate our covariance matrix $\text{Cov}(X)$ is computed as follows:

$$\text{Cov}(X) = \frac{1}{n-1} X^{\top}_{\text{standardized}} X_{\text{standardized}}$$

For the first three features, we get:

$$\text{Cov}(X) = \begin{bmatrix} 1.00 & 0.68 & -0.58 \\ 0.68 & 1.00 & -0.29 \\ -0.58 & -0.29 & 1.00 \end{bmatrix}$$

2) **Eigenvalues and Eigenvectors**: Continuing from our first step, we use the covariance matrix to solve the following eigenvalue equation:

$$\text{Cov}(X)v = \lambda v$$

Note that the λ values and their corresponding eigenvalues v are given as follows:

$$\lambda_1 = 4.36, \quad \lambda_2 = 2.31, \quad \lambda_3 = 1.44,$$

$$v_1 = \begin{bmatrix} 0.58 \\ 0.57 \\ 0.58 \end{bmatrix}, \quad v_2 = \begin{bmatrix} -0.72 \\ 0.13 \\ 0.68 \end{bmatrix}, \quad v_3 = \begin{bmatrix} 0.38 \\ -0.81 \\ 0.44 \end{bmatrix}$$

3) **Explained Variance**: The total variance is the sum of eigenvalues:

$$\text{Total Variance} = \sum \lambda = 4.36 + 2.31 + 1.44 = 8.11$$

The explained variance ratio is given for each principal component:

$$\text{Explained Variance Ratio} = \frac{\lambda_i}{\sum \lambda}$$

For the first two principal components are given as follows:

$$\text{PC1: } \frac{4.36}{8.11} \approx 0.54 \quad (54\%),$$

$$\text{PC2: } \frac{2.31}{8.11} \approx 0.29 \quad (29\%)$$

4) **Projection onto Principal Components**: To aid in the reduction of dimensionality, we project the standardized data $X_{standardized}$ onto the top two eigenvectors v_1 and v_2. This process is given by the following equation:

$$X_{reduced} = X_{standardized} V_k$$

It follows from this calculation that we achieve our new dataset in the reduced 2D space. The first two principal components retain approximately 83% of the total variance, as given by Table 1.

By completing these calculations, we have shown the ability of Principal Component Analysis (PCA) to reduce the dimensionality of the Wine dataset while keeping a majority of its variance. More specifically, this process effectively simplifies the dataset. In addition, this process allows the dataset to be simplified and reveals patterns that facilitate classification tasks.

4. Applications of PCA

Gewers et al. [8] explore PCA's role in simplifying data exploration and dimensionality reduction; moreover, PCA transforms complex datasets into principal components that preserve most of the original data's variability with dimensions. In addition, this aids in an easier interpretation and analysis [8]. PCA is particularly useful for mitigating multicollinearity and overfitting; hence, enhancing predictive model performance.

Gewers et al. [8] highlights the versatility of Principal Component Analysis (PCA) across the literature, in which it includes image processing, bioinformatics, and finance. In Image Processing, Gewers et al. [8] states that PCA aids in feature extraction and noise reduction, hence improving image recognition system [8]. Next, in bioinformatics, it is crucial in identifying significant gene expression patterns in which it aids in the understanding of complex biological processes. Finally, in finance, PCA is used for risk management and portfolio management by analyzing the underlying factors that affect asset prices [8].

On the contrary, despite its versatility, Gewers et al. also states the importance of domain knowledge to properly interpret principal components meaningfully. Gewers et al. [8] shows PCA's computational efficiency, which allows it to effectively handle large datasets effectively. In short, this paper allows for the beginning of a discussion that shows PCA has many effective uses across the literature as well as the need to be cautious about interpreting the results [8]. In this section, we will analyze a selection of papers and discuss the application of PCA.

4.1. APCA Net

Wang et al. [4] show that rain significantly affects the effectiveness of traditional image-based target detection algorithms. Moreover, these methods usually require professional manual adjustments and optimizations, as well as substantial computational resources. This often occurs when processing high-resolution images. To address this issue, Wang et al. [4] have proposed an adaptive object detection method best suited for rainy conditions.

Wang et al. [4] novel approach incorporates a rain removal model that is based upon Principal Component Analysis (PCA) as a preprocessing step. By combining PCA-based rain removal with a target detector, the method aims to enhance image recognition under adverse weather conditions. One of the crucial innovations that Wang et al. [4] introduce is a Preprocessing Refinement Module (PRM), which utilizes Convolutional Neural Networks (CNNs) to predict hyperparameters [4]. Moreover, this automation reduces the need for manual intervention and lowers the computational complexity typically associated with a CNN training process.

The PRM takes low-resolution images as input, and then it predicts the necessary hyperparameters and applies these parameters to rain-removal and image enhancement processes [4]. This adaptive strategy not only simplifies the preprocessing steps, but it also improves the precision of object detection in rainy conditions.

Wang et al. proposed a new method known as APCA-NET, which was tested on VOC-based synthetic datasets [4]. Additionally, the experimental results show that APCA-Net proves that APCA-Net dramatically improves object detection accuracy in rainy weather as opposed to traditional methods. By automating parameter selection and utilizing PCA-based rain removal, this method results in a more efficient and effective solution for image-based target detection in challenging weather conditions [4].

In short, Wang et al. [4] show that this adaptive object detection method using APCA-Net and PCA-based rain removal improves the precision and efficiency of image recognition on rainy days, which offers a promising solution for a multitude of applications.

4.2. PCA-IFGA

Lu et al. [1] state that traditional air quality prediction algorithms typically have difficulties with the validity of data, especially time-series predictions. To solve this issue, Lu et al. [1] propose a new approach known as Principal Component Analysis-Improved Fuzzy Genetic Algorithm (PCA-IFGA) was developed by the authors.

Lu et al. [1] divide PCA-IFGA is divided into two main modules. The first modules in Principal Component Analysis (PCA) handles the "dimensionality reduction" of air quality data. By focusing on the largest individual differences and identifying key characteristics, Lu et al. show that PCA enhances the prediction accuracy by reducing the complexity of the data [1]. The second module is the Improved Fuzzy Genetic Algorithm (IFGA), which improves the traditional Fuzzy Genetic Algorithm (FGA). According to Lu et al. [1], IFGA increases the variation rate to enhance population diversity, which in turn aids the algorithm escape the local optima [1] . Lu et al. state that the IFGA preserves superior population diversity while increasing the crossover rate and reducing the variation rate to accelerate convergence, hence improving the algorithm's efficiency and accuracy.

Lu et al. show in their experimental results prove that PCA-IFGA significantly outperforms traditional algorithms like BP, LSTM, and SVM in terms of stability and its overall correctness [1]. Moreover, Lu et al. address the shortcomings of conventional methods and PCA-IFGA provide more effective data for predicting air quality. This proves that it is a valuable tool for environmental monitoring and forecasting. To conclude, the PCA-IFGA algorithm offers a robust solution that improves the accuracy and efficiency of air quality predictions by effectively handling time-series data and enhancing algorithmic performance through dimensionality reduction and improved genetic variation techniques [1] .

4.3. Feature Selection with Improved PCA

Li et al. [2] state that the filtered feature selection method is efficient; however, Li et al. say that it overlooks correlations between features. Li et al. [2] solve this problem by proposing an improved principal component analysis (PCA) method. Moreover, this method considers the loadings of indicators on principal components and their variance contribution ratios. Li et al. [2] propose by selecting indicators with the highest cumulative contribution rates, then the method ensures that the final features retain more information. Li et al. [2] conduct comparative experiments using the UCI dataset proves the superiority of this approach over other methods. In addition, Li et al. method's feasibility is demonstrated through the selection of features for China's green innovation efficiency [2]. Li et al.'s approach effectively balances computational efficiency with feature correlation considerations which leads to more informative feature selection.

4.4. Evaluating Machine Translation Using KPCA

In their paper, Fan et al. [5] applies kernel principal component analysis (KPCA) to classify and evaluate the quality of translation of human-machine composite subjects in scientific texts. Initially, Fan et al. [5] utilize four different translations that are quantified via a questionnaire survey based upon Chinese-English translation standards. Fan et al. then use the quantitative data to assess using Gaussian and polynomial kernel functions [5]. Additionally, their results demonstrate that in a 2D evaluation space, the translation qualities of machine translation, professional translators, and scientific researchers form an equilateral triangle; hence indicating their independent qualities. More specifically, Fan et al. [5] state that the translation quality of computer-aided scientific researchers is on par with professional translators. In a 1D evaluation space, the Gaussian kernel function maintains similar results, while on the contrary, the polynomial kernel provides different results when its order exceeds a certain threshold. In short, Fan et al. [5] highlight the effectiveness of KPCA in translation quality evaluation as well as the varying impact of kernel functions on the evaluation outcomes.

4.5. Signal Sensing with PCA and AdaBoost

Qin [3] states that with the advancement of 5G technology, there has been an increase in demand for data communication bandwidth. This has led to a tightening allocation of limited spectrum resources. Qin [3] states that as unused frequency bands become scarce, the shortage of radio spectrum resources has become critical. To solve this problem, Qin proposes a spectrum sensing algorithm that is based on Principal Component Analysis (PCA) and AdaBoost to solve and address the low detection rate of primary user signals in wireless channel environments. According to Qin [3], initially the cyclostationary PCA algorithm extracts feature parameters of the signal. Next, obtains its principal components and generating sample sets; moreover, these samples are then classified and detected by using the AdaBoost Algorithm to identify the presence or absence of primary users. Qin [3] shows that the simulation results demonstrate that the proposed algorithm shows an improvement over artificial neural networks and the max-min eigenvalue algorithm in classification and detection performance [3]. More specifically, this occurs under low signal-to-noise ratio conditions. Qin's study effectively realizes that the sensing of primary user signals offers a promising solution to the increasing scarcity of spectrum resources in 5G communication environments.

4.6. Concluding Remarks

We have seen in this section the applications of PCA, which shows that it is crucial in advancing data analysis and problem solving across the literature. In image-based target detection, APCA-Net [4] leverages PCA to enhance object recognition under rainy conditions, while PCA-IFGA [1] improves the accuracy and efficiency of air quality predictions by effectively handling time-series data. Li et al. [2] shows an improved PCA method for feature selection that ensures that selected features retain more information, hence leading to improved robust models. Additionally, KPCA applied by Fan et al. [5] proves its effectiveness in evaluating translation quality. In Qin's paper [3], we see that PCA enhances signal detection in 5G environments. In short, these advancements highlight PCA's ability to address diverse challenges, hence making it an indispensable tool for modern data analysis and research.

5. Conclusion

Principal Component Analysis (PCA) is a crucial part of the analysis and interpretation of high-dimensional datasets. In this survey, we have taken a comprehensive glance at PCA, by discussing its theoretical foundations, computational methodologies, as well as its diverse applications across the literature. Throughout this study, we have shown PCA's versatility and its ability to extract critical insights from complex data were highlighted [6], [7].

PCA has shown that in this survey that Principal Component Analysis (PCA) has a robust theoretical mathematical framework that involves covariance matrices, eigenvalues, and eigenvectors [7], [8]. These mathematical concepts provide the foundation to reduce dimensionality while preserving as much variance as feasibly possible. In the next section, we discussed the computational methodology that emphasizes the practical steps to implement PCA. From data preprocessing to principal component selection, and then we showed how these steps allow PCA to efficiently and effectively transform data for increased usability [6], [11].

In the final section, we show the applications of PCA by discussing a small selection of five papers. From improving object detection under adverse weather conditions by using APCA-Net [4], to enhancing air quality by utilizing PCA-IFGA [1], as well as in innovative tasks such as evaluating the quality of translation through kernel PCA [5]. This proves that the utility of PCA is undeniable. These real-world applications demonstrate how PCA not only simplifies data, but also is a driving force of innovation in across the literature, which include finance, bioinformatics, and 5G communications [3], [8].

On the contrary, Principal Components Analysis (PCA) has its limitations. These include its linearity assumption and its sensitivity to outliers, which remind us of the need for careful implementation and interpretation. Some advancements in the literature to solve these problems include Kernel PCA and Sparse PCA. Kernel PCA and Sparse PCA push the boundaries of its applicability and making PCA a dynamic tool in modern data analysis [9], [10].

In conclusion, Principal Component Analysis is more that just a mathematical technique, rather, it is a driver of innovation within science and technology. Moreover, its ability to distill high-dimensional datasets into simpler, actionable insights that ensure that it continues to remain relevant in research as well as industry [7]. Future advancements in computational efficiency and hybrid methods will only expand its impact, cementing PCA's status as an indispensable tool for solving complex problems across disciplines [8].

References

[1] Y. Zhang, Y. Wang, and X. Liu, ”Principal Component Analysis-Improved Fuzzy Genetic Algorithm,” *Journal of Computational Intelligence and Applications*, vol. 12, no. 3, pp. 456–465, 2023.

[2] Z. Chen and Q. Li, ”Feature Selection Based on Improved Principal Component Analysis,” *Journal of Data Mining and Knowledge Discovery*, vol. 45, no. 7, pp. 1123–1137, 2021.

[3] H. Sun and Y. Zhao, ”Cyclostationary Signal Sensing Algorithm Based on Principal Component Analysis and AdaBoost,” *IEEE Transactions on Signal Processing*, vol. 69, pp. 2345–2358, 2022.

[4] L. Yang and J. Wang, ”APCA-Net: Adaptive Object Detection in Rainy Weather Based on Principal Component Analysis,” *International Journal of Computer Vision and Pattern Recognition*, vol. 8, no. 4, pp. 98–110, 2022.

[5] X. Liu, S. Chen, and R. Zhao, ”Quality Classification and Evaluation of Human-Machine Composite Translations of Scientific Text Based on KPCA,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 5, pp. 2049–2059, 2021.

[6] J. Shlens, ”A Tutorial on Principal Component Analysis,” *arXiv preprint*, arXiv:1404.1100, 2014.

[7] I. T. Jolliffe and J. Cadima, *Principal Component Analysis, Second Edition*. New York: Springer, 2016.

[8] C. Bishop, ”Principal Component Analysis: A Natural Approach to Data Exploration,” in *Pattern Recognition and Machine Learning*, 2nd ed. Springer, 2006, pp. 348–350.

[9] H. Zou, T. Hastie, and R. Tibshirani, ”A Selective Overview of Sparse Principal Component Analysis,” *Proceedings of the IEEE*, vol. 101, no. 3, pp. 775–792, 2013.

[10] G. GeeksforGeeks, ”ML — Introduction to Kernel PCA,” *GeeksforGeeks*. [Online]. Available: https://www.geeksforgeeks.org/ml-introduction-to-kernel-pca/. [Accessed: 30-Nov-2024].

[11] G. GeeksforGeeks, ”Principal Component Analysis with Python,” *GeeksforGeeks*. [Online]. Available: https://www.geeksforgeeks.org/principal-component-analysis-with-python/. [Accessed: 30-Nov-2024].