Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 163 (2019) 292–299

Procedia Computer Science

www.elsevier.com/locate/procedia

16th International Learning & Technology Conference 2019

Data dimensional reduction and principal components analysis

Nema Salem[a*], Sahar Hussein[b†]

[a]*Assistant Professor, Electrical and Computer Engineering Department, Effat University, Jeddah 21478, KSA*
[b]*Assistant Professor, National Science Mathematics and Technology Unit, Effat University, Jeddah 21478, KSA*

Abstract

Research in the fields of machine learning and intelligent systems addresses essential problem of developing computer algorithms that can deal with huge amounts of data and then utilize this data in an intellectual way to solve a variety of real-world problems. In many applications, to interpret data with a large number of variables in a meaningful way, it is essential to reduce the number of variables and interpret linear combinations of the data. Principal Component Analysis (PCA) is an unsupervised learning technique that uses sophisticated mathematical principles to reduce the dimensionality of large datasets. The goal of this paper is to provide a complete understanding of the sophisticated PCA in the fields of machine learning and data dimensional reduction. It explains its mathematical aspect and describes its relationship with Singular Value Decomposition (SVD) when PCA is calculated using the covariance matrix. In addition, with the use of MATLAB, the paper shows the usefulness of PCA in representing and visualizing Iris dataset using a smaller number of variables.

Keywords: PCA; Data dimension reduction; Iris dataset; SVD

1. **Introduction**

It is easy plotting one-dimensional data on the number line, two-dimensional data set in $xy-plane$, and three-dimensional data in space. It is difficult to plot and to visualize a data set with a dimension higher than three.

* Corresponding author. Tel.: +966920003331; fax: +966126377447.
E-mail address: nsalem@effatuniversity.edu.sa
† Corresponding author. Tel.: +966920003331; fax: +966126377447.
E-mail address: shussein@effatuniversity.edu.sa

10.1016/j.procs.2019.12.111

Nowadays, there are large sets of data in many fields as science and engineering with multi-dimension and there is an urgent need to analyze them and extract the dominant features. It is difficult to extract information from a dataset with a number of variables, n, greater than three. Conventionally, researchers use a series of bivariate plots to analyze the dataset trying to find any relationships between variables. However, the number of such plots required for such a task is $n!/2!(n-2)!$ that is not feasible for large datasets. Thus, there is a need for a powerful analytical method.

Therefore, to interpret data with a large number of variables in a meaningful form, it is essential to reduce the number of variables and interpret linear combinations of the data. PCA is a technique that uses mathematical principles to transform a number of possibly correlated variables into a smaller number of variables called principal components. The new basis assures filtering the noise out and reveals the hidden structure of the original dataset. It has a wide range of applications based on large datasets. PCA uses a vector space transform to reduce the dimensionality of large datasets. It finds the directions of maximum variance in high-dimensional data that is equivalent to the least squares line of best fit through the plotted data, and projects it onto a smaller dimensional subspace while retaining most of the information. PCA method informs the contributions of each principal component, to the total variance, and the eigenvectors associated with non-zero eigenvalues, of the coordinates. In practice, it is sufficient to include enough principal components that cover about (70 − 80%) of the data variation. The reduced dimension dataset allows users to interpret, analyze, and process data in an easy way.

The aim of this research is to give a clear understanding of PCA and explains its mathematical side. In addition, it discusses the SVD method and its relation to the PCA technique. From classical matrix theory, SVD plays a fundamental role in matrix computation and analysis such as matrix Polar decomposition, and Least squares. Moreover, the paper presents the implementation approach of PCA based covariance and SVD for a real-world Iris dataset.

We organized this paper as follows. Section 2 overviews the importance of SVD in reducing the data dimension in terms of some related work. Section 3 explains the principal components analysis method while section 4 explains the SVD and explains its utilization in the PCA calculations. Section 5 shows the implementation of PCA and the use of SVD on the real Iris dataset. Section 6 shows and explains the obtained results. Lastly, section 7 concludes the article's topic.

2. Related Work

In 2015, Qiao could choose the suitable rank of the factorization and provide a good initialization for Non-Negative Matrix Factorization (NMF) algorithms by using Singular Value Decomposition (SVD) [1].

In 2015, Kumar et al. proposed an ECG signal compression algorithm for the huge data of the ambulatory system. Their algorithm is based on the SVD and wavelet difference reduction techniques. The algorithm gave a compression rate of up to 21.4:1 with excellent quality of signal reconstruction [2].

In 2016, Houari et al. proposed Copulas and the LU-decomposition technique for the dimensionality reduction. They applied their technique on real-world datasets as Diabetes, Waveform, Human Activity Recognition based on Smartphone, and Thyroid Datasets and they got promising results [3].

In 2016, Menon et al. proposed Hadamard-based random projection with fast SVD (FSVD) algorithm. Their experimental results proved that their proposed algorithm is better than Gaussian-based FSVD for dimensionality reduction in hyperspectral classification [4].

In 2017, Kumar and Manoj presented a compression technique for encrypted images by the use of Discrete Wavelet Transform, Singular Value Decomposition and Huffman coding [5].

In 2017, Olive used the classical PCA in explaining the reduction and concentration structure with a few linear uncorrelated combinations of the original variables. He implemented PCA in data reduction and interpretation [6].

In 2018, Feng et al. proposed a tensor SVD algorithm for cloud cyber data reduction [7].

3. Principal components analysis, PCA

As in [8], a dataset, X with m rows representing the variables and n columns representing the observations is represented in a matrix with $m-$ row vectors, each of length n, as in equation (1).

$$X = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{pmatrix} = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_m \end{pmatrix} \tag{1}$$

It is required to linearly transform matrix X into $Y_{m\times n}$ matrix by using a $P_{m\times m}$ matrix with row vectors $p_1, p_2, \ldots, p_m$ and column vectors of X matrix: $x_1, x_2, \ldots, x_n$; equation (2).

$$Y_{m\times n} = P_{m\times m}\ X = \begin{pmatrix} p_1 \cdot x_1 & p_1 \cdot x_2 & \cdots & p_1 \cdot x_n \\ p_2 \cdot x_1 & p_2 \cdot x_2 & \cdots & p_2 \cdot x_n \\ \cdots & \cdots & \ddots & \cdots \\ p_m \cdot x_1 & p_m \cdot x_2 & \cdots & p_m \cdot x_n \end{pmatrix} \tag{2}$$

The dot product $p_i \cdot x_j$ indicates the projection of the original dataset X on the columns of P. The rows of P are a new basis that represents the principal component directions.

PCA considers the variance of the data in the original basis, seeks to de-correlate the original data by finding the directions in which variance is maximum, and then use these directions to define the new basis.

3.1. Standardize the variables

The first step of PCA analysis is subtracting the mean values from the dataset Z to provide a proper normalization and to give equal weights in the analysis phase of the original dataset, equation (3).

$$X_{i,j} = Z_{i,j} - \overline{X}_j \tag{3}$$

Where $X_{i,j}$ is the standardized dataset, $Z_{i,j}$ is the data variable j in the sample unit i and $\overline{X}_j$ is the sample mean for the variable j.

3.2. Covariance matrix

The covariance of two variables is a measure of their correlation. It has the all-possible covariance pairs between m variables and its diagonal includes all variances. Equation (4) represents the covariance of the standardized dataset, X.

$$C_X = \frac{1}{n-1} X\ X^T = \frac{1}{n-1} \begin{pmatrix} x_1 x_1^T & x_1 x_2^T & \cdots & x_1 x_m^T \\ x_2 x_1^T & x_2 x_2^T & \cdots & x_2 x_m^T \\ \vdots & \vdots & \ddots & \vdots \\ x_m x_1^T & x_m x_1^T & \cdots & x_m x_m^T \end{pmatrix} \tag{4}$$

The large variance values are important as they correspond to the interesting dynamics in systems while the small variance values may represent the noise in these systems. Therefore, the covariance, C_Y, of the transformed matrix $Y_{m\times n}$ should meet the following requirements:

- Maximize the signal by maximizing the diagonal entries.
- Minimize the covariance between variables by minimizing the off-diagonal entries.

We can achieve the requirements by choosing the transformation matrix $P_{m\times m}$ with orthogonal vectors

$p_1, p_2, \ldots, p_m$ such that C_Y is similar to a diagonal matrix as in equation (5).

$$C_Y = \frac{1}{n-1} Y Y^T = \frac{1}{n-1}(PX)(PX)^T = \frac{1}{n-1}(PX)(X^T P^T) = \frac{1}{n-1} P(XX^T) P^T = \frac{1}{n-1} PSP^T \quad (5)$$

Since $(XX^T)^T = XX^T$ and every square symmetric matrix is orthogonally diagonalizable, then the symmetric matrix $S_{m \times m}$ can be written as in equation (6).

$$S = E D E^T \quad (6)$$

where E is an $m \times m$ orthogonal matrix whose columns are the orthogonal eigenvectors of S, and D is a diagonal matrix that has the eigenvalues of S as its diagonal entries. The rows of the matrix P are the eigenvectors of S such that $P = E^T$. Thus, the covariance matrix of Y is in equation (7).

$$C_Y = \frac{1}{n-1} PSP^T = \frac{1}{n-1} E^T (EDE^T) E = \frac{1}{n-1} D \quad \{as\ EE^T = I; where\ I\ is\ an\ m \times m\ Identity\ matrix\} \quad (7)$$

In accordance with the spectral decomposition theorem, we can write the covariance matrix as a summation over the ℓ eigenvalues, multiplied by the product of the corresponding eigenvectors times its transpose.

3.3. The I[th] principal component PCAi of the transformed dataset Y_i

Each PC is a linear combination of $x-$variables with coefficients that meet the following requirements.

- Maximizing the variance of Y_i.
- The sum of squared coefficients of each eigenvector equals one. That is the norm of each eigenvector equals one.
- The new principal component is uncorrelated with all previously defined principal components.

3.4. The essential Principal Components

Considering $\lambda_1 \ldots \lambda_\ell$ are the eigenvalues of the covariance matrix C_Y with $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_\ell$, thus Y_i represents a proportion of the total variation that equals (equation 8):

$$\frac{\lambda_i}{\lambda_1 + \lambda_1 + \cdots + \lambda_\ell} \quad (8)$$

To avoid loss of information, the proportion of variation by the first k principal components should be large. Thus, equation 9:

$$\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_\ell} \simeq 1 \quad (9)$$

4. Singular value decomposition, SVD

As in [9], SVD is a general method for a change of basis and is used in the principal component analysis. According to SVD, we can factorize any matrix $A \in R^{m \times n}$ into, equation 10:

$$
\begin{aligned}
&A = U\,\Sigma\,V^T \\
&U \in R^{m\times m} \qquad \textit{is orthogonal}\ \left(i.e.\ U\,U^T = I\right) \\
&\Sigma \in R^{m\times n} \qquad \textit{is diagonal with leading positive diagonal entries} \\
&V \in R^{n\times n} \qquad \textit{is orthogonal}\ \left(i.e.\ V\,V^T = I\right)
\end{aligned} \tag{10}
$$

The diagonal entries, σ_i of Σ are non-negative and represent the singular values of A. They are ordered in descending order: $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_\ell \geq 0$; where $\ell = \min(n,m)$.

Since $U \in R^{m\times m}$ and $V \in R^{n\times n}$ are orthogonal matrices, then their columns form bases for the vector spaces R^m and R^n, respectively. Thus, we can expand any vector $b \in R^m$ in the basis formed by the columns of U (the left singular vectors of A) and we can expand any vector $v \in R^n$ in the basis formed by the columns of V (the right singular vectors of A). The vectors for these expansions $\hat{b}$ and $\hat{x}$ are in equation 11:

$$
\begin{aligned}
&\hat{b} = U^T b \quad \textit{and} \quad \hat{x} = V^T x \\
&\quad \textit{if}\ b = Ax\,,\ \textit{then} \\
&\quad U^T b = U^T A\,x \\
&\hat{b} = U^T\left(U\,\Sigma V^T\right)x = \Sigma\hat{x}
\end{aligned} \tag{11}
$$

Therefore, V moves original basis to a new one, Σ stretches the new basis, and U describes results with respect to actual data. Linear algebra theorem states that the non-zero singular values of A are the square roots of the nonzero eigenvalues of AA^T or A^TA, equation (12).

$$
A^TA = \left(U\,\Sigma V^T\right)^T\left(U\,\Sigma V^T\right) = \left(V\,\Sigma^T U^T\right)\left(U\,\Sigma V^T\right) = V\left(\Sigma^T\,\Sigma\right)V^T \tag{12}
$$

As A^TA is similar to $\Sigma^T\Sigma$, then it has the same eigenvalues. Since $\Sigma^T\Sigma$ is a square $m\times m$ diagonal matrix, the eigenvalues are the diagonal entries, which are the squares of the singular values. Thus, the nonzero eigenvalues of each of the covariance matrices, AA^T and A^TA are actually identical. Since A^TA is symmetric, then it is an orthogonal diagonalization and thus the eigenvectors of A^TA are the columns of V.

5. Proposed PCA approach for data dimensional reduction

To reduce the dimensions of a d-dimensional dataset and to project it onto a k-dimensional subspace where $k < d$ with high computational efficiency and retaining most of the information, the procedures are:

- Standardize the dataset
- Obtain eigenvalues from the covariance matrix or perform SVD.
- Sort eigenvalues in descending order and choose k eigenvectors that correspond to the largest k eigenvalues.
- Create the projection matrix P from the selected k eigenvectors.
- Transform the original dataset X via P to obtain a k-dimensional subspace Y.
- The eigenvalues and eigenvectors of a square matrix A are calculated from $|A - \lambda I| = 0$ & $\left(A - \lambda_j I\right)e_j = 0$. Considering $\lambda_1 \cdots \lambda_\ell$ are the eigenvalues of the covariance matrix with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_\ell$ and the vectors $e_1 \cdots e_\ell$ are the corresponding eigenvectors. Then, the variance of Y_i = the variance of $\left(e_{i1}x_1 + e_{i2}x_2 + \cdots + e_{i\ell}x_\ell\right) = \lambda_i$

6. Iris dataset dimensional reduction

6.1. Iris information

The famous dataset in pattern recognition is the Iris dataset, shown in Fig. 1. It contains measurements of "150" Iris flowers from "3" different species, "50" measurements for each class. The three classes, types of Iris plant, are Iris-Setosa, Iris-Versicolor, and Iris-Virginica. The four features are Sepal-length in cm, Sepal-width in cm, petal-length in cm, and petal-width in cm. One class is linearly separable from the other "2"; the latter are NOT linearly separable from each other [10].

Fig. 1. Iris flowers: Versicolor, Virginica, Setosa, and Versicolor with labels.

6.2. Procedures, results, and discussion

The approach starts with loading the dataset into the MATLAB package and splitting it into data and class files. The data-file is 150×4 in which the columns represent the different features and each row represents a specific flower sample. Each sample-row X is a *4*-dimensional vector as in Table 1. The algorithm visualizes the plots of each two variables of the *THREE* different flower-classes along the *FOUR* different features shown in Fig. 2a.

For optimal response (although all features are measured in centimeters), the second step of the algorithm is standardizing the dataset onto the unit scale (zero-mean and unity-variance), shown in Fig. 2b. The algorithm determines the eigenvalues and the corresponding eigenvectors from the covariance matrix and the SVD.

In the third step, the approach selected the principal components by ranking the eigenvalues in descending order and selecting the top K eigenvectors, given in Fig. 3a. The calculated variance from the eigenvalues informs how much information is attributed to each principal components.

The last step in the approach is projecting the original data onto the new feature space. The $d\times k$ projection matrix is a catenation of the top k eigenvectors and the *FOUR*-feature space of Iris data is reduced to a 2-dimensional feature subspace be selecting the top *TWO* eigenvectors with the highest eigenvalues. Thus, we use the 4×2 projection matrix to transform the Iris data into a new 150×2 matrix. Figs. 3b, 3c, and 3d show the representation of the Iris data in the new feature space using the first principal component and the first two principal components (obtained from covariance and SVD), respectively. Table 2 includes the intermediate data and shows that the first two PCs contain 99.96% of the information. In addition, the MATLAB algorithm takes only about 5 seconds.

Table 1. Measurements of 150 Iris flowers.

Sepal length	Sepal width	Petal length	Petal width	Species (0: Setosa, 1: Versicolor, 2: Virginica
5.1	3.5	1.4	0.2	0
4.9	3	1.4	0.2	0
⋮	⋮	⋮	⋮	⋮
5	3.3	1.4	0.2	0
7	3.2	4.7	1.4	1
6.4	3.2	4.5	1.5	1

⋮	⋮	⋮	⋮	⋮
5.7	2.8	4.1	1.3	1
6.3	3.3	6	2.5	2
5.8	2.7	5.1	1.9	2
⋮	⋮	⋮	⋮	⋮
6.2	3.4	5.4	2.3	2

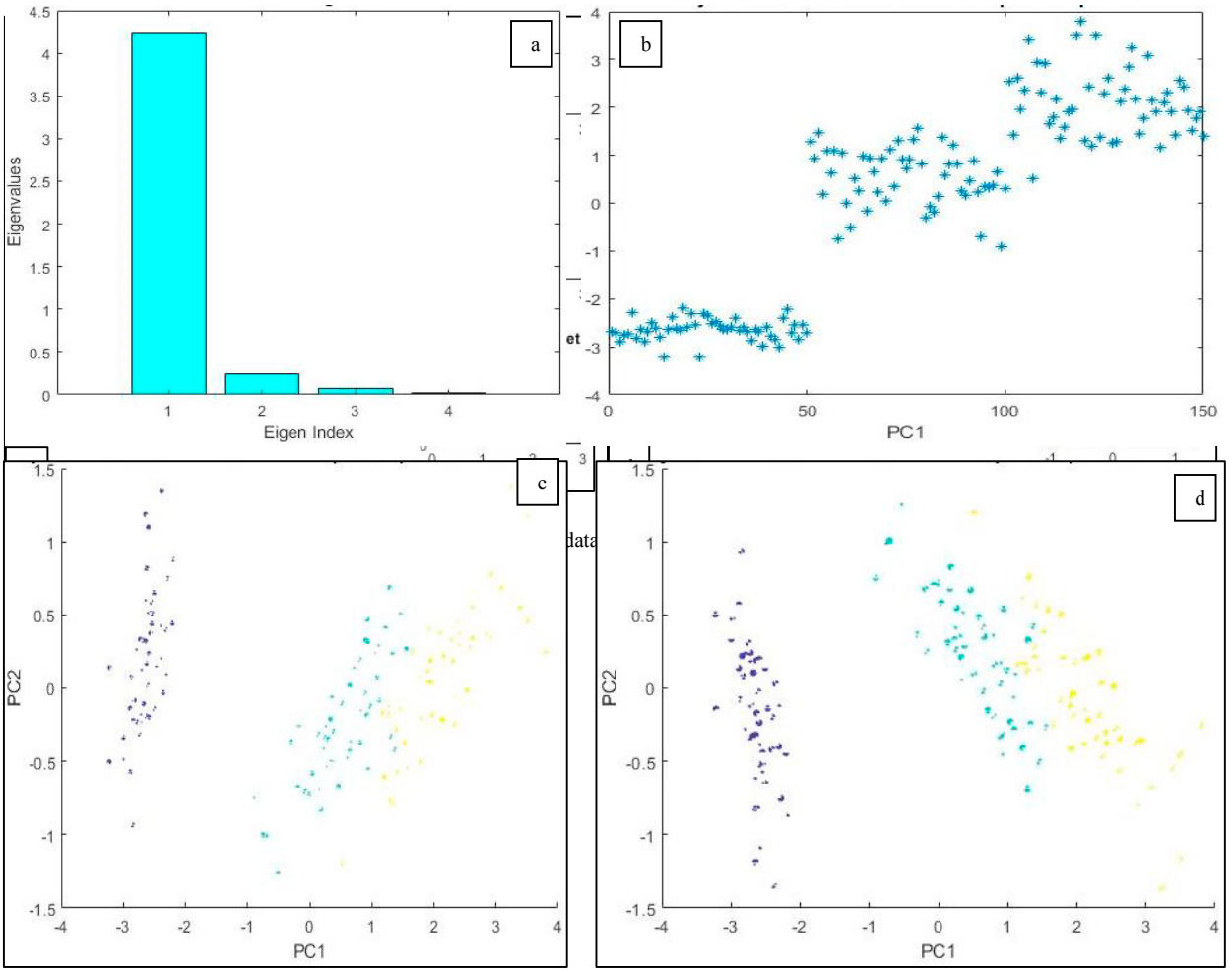

Fig. 2. (a) Variation of the eigenvalues; (b) projection on the 1st PC; (c) projection on the 1st two PCs (variance); (d) projection on the 1st two PCs (SVD).

Table 2. Intermediate calculations: 4x4 covariance matrix; 4x4 eigenvalues matrix; sorted eigenvalues and the corresponding PCA matrix; strength of PCs; run time.

Covariance matrix				***Eigenvalues matrix***				***Sorted eigenvalues and their indices***	
0.6857	-0.0393	1.2737	0.5169	0.0237	0	0	0	4.2248	4
-0.0393	0.1880	-0.3217	-0.1180	0	0.0785	0	0	0.2422	3

1.2737	-0.3217	3.1132	1.2964	0	0	0.2422	0	0.0785	2
0.5169	-0.1180	1.2964	0.5824	0	0	0	4.2248	0.0237	1
Principal components matrix					***Sorted principal components***			***The strength of PCs***	
-0.3173	0.5810	0.6565	0.3616	0.3616	0.6565	0.5810	-0.3173	*Total Variations* = 17.9147	
0.3241	-0.5964	0.7297	-0.0823	-0.0823	0.7297	-0.5964	0.3241		
0.4797	-0.0725	-0.1758	0.8566	0.8566	-0.1758	-0.0725	0.4797	*Weight of the first 2 PCs* = 17.908 = 99.96%	
-0.7511	-0.5491	-0.0747	0.3588	0.3588	-0.0747	-0.5491	-0.7511		
								Run Time ≅ 5s	

7. Conclusion

PCA is an optimal orthogonal transformation for a group of vectors providing a minimum number of non-correlated PCs with a concentrated maximum part of the original set's energy. Each PC has zero mean and standard deviation equals the square root of the eigenvalue. Interpretation of the PCs depends on finding the most strongly correlated variables of the original data with each component (large in magnitude and the farthest from zero in both directions).

The calculation of PCA and eigenvectors (*that are the PCs*) depends on a very large, in size, covariance matrix. Thus, the complexity of the computational approach increases, especially with large datasets. To overcome this problem, this research proves that PCA based on SVD is a simple method for extracting significant information from bulky datasets. With minimal effort, PCA offers a roadmap for reducing a complex dataset to a lower dimension revealing the hidden structures. In addition, PCA based on SVD is a powerful tool in machine learning and data analysis in various fields as science and engineering. Further work is the implementation of PCA based on SVD in critical applications as classification in image processing, and extracting specific brain rhythms from noisy recorded EEG signals.

References

[1] Qiao, Hanli. (2015) "New SVD based initialization strategy for non-negative matrix factorization" *Pattern Recognition Letters* 63: 71-77.

[2] Kumar, Ranjeet; A. Kumar; and G. K. Singh. (2015) "Electrocardiogram signal compression based on singular value decomposition (SVD) and adaptive scanning wavelet difference reduction (ASWDR) technique" *AEU-International Journal of Electronics and Communications* 69.12: 1810-1822.

[3] Houari, Rima; et al. (2016) "Dimensionality reduction in data mining: A Copula approach." *Expert Systems with Applications* 64: 247-260.

[4] Menon, Vineetha; Qian Du; and James E. Fowler. (2016) "Fast SVD with random Hadamard projection for hyperspectral dimensionality reduction." *IEEE Geoscience and Remote Sensing Letters* 13.9: 1275-1279.

[5] Kumar, Manoj; and Ankita Vaish. (2017) "An efficient encryption-then-compression technique for encrypted images using SVD." *Digital Signal Processing* 60: 81-89.

[6] Olive, David J. (2017) "Principal component analysis." *Robust Multivariate Analysis*. Springer, Cham, 189-217.

[7] Feng, Jun; et al. (2018) "A Secure Higher-Order Lanczos-Based Orthogonal Tensor SVD for Big Data Reduction." *IEEE Transactions on Big Data*.

[8] I. T. Jolliffe. (2002) "Principal Component Analysis", 2nd ed. *Springer series in statistics*.

[9] P. David. (2015) "Linear Algebra: A Modern Introduction", 4th ed. *Cengage Learning*.

[10] D. a. K. T. Dua, E. (2017) "UCI Machine Learning Repository", *Available: http://archive.ics.uci.edu/ml*