Principal Component Analysis for Medical Diagnostics: A Baymax-Inspired Approach to Automated Health Scanning

Muhammad Edo Raduputu Aprima - 13523096[1,2]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13523096@std.stei.itb.ac.id, [2]muhammadedo017@gmail.com

***Abstract*—The ability to efficiently scan and diagnose medical conditions is a cornerstone of modern healthcare technology. This paper explores the application of Principal Component Analysis (PCA) as a dimensionality reduction technique in medical diagnostics, inspired by the scanning capabilities of Baymax, a fictional healthcare robot from *Big Hero 6*. By reducing high-dimensional medical data into smaller set of principal components, PCA enables the identification of critical patterns and anomalies that may indicate underlying conditions. This study discusses the broader implications of intregating PCA into autonomous diagnostic system, drawing parallels with Baymax's futuristic scanning methods.**

***Keywords*—Principal Component Analysis, medical diagnostics, Baymax, eigenvectors.**

I. Introduction

Medical diagnostics have always been a cornerstone of healthcare, evolving from manual assessments to advanced computational technologies. In modern medicine, diagnostic tools such as imaging devices and biochemical analyzers generate vast amounts of high-dimensional data. While these data sets hold critical insights, their size and complexity often make them challenging to process effectively. This is particularly evident in fields like radiology, where analyzing medical images such as CT or MRI scans requires not only precision but also efficiency.

The concept of a healthcare robot capable of instantaneous and accurate health scanning, as represented by Baymax in *Big Hero 6*, offers a vision of what the future of diagnostics might look like. Baymax demonstrates the ability to assess a patient's condition using advanced scanning techniques, analyze the collected data in real time, and provide actionable insights. While the fictional character's technology is beyond current capabilities, its foundation in computational data analysis and pattern recognition is already being explored in real-world applications.

Principal Component Analysis (PCA) is one such method that has become essential in handling high-dimensional data. PCA simplifies complex data by reducing its dimensions while preserving the most significant features. This ability to isolate key patterns makes it a powerful tool for tasks like medical diagnostics, where identifying anomalies or disease markers is crucial. By leveraging eigenvalues, eigenvectors, and variance maximization, PCA allows the extraction of meaningful insights from seemingly overwhelming datasets.

II. Theoritical Foundation

A. Eigenvalues and Eigenvectors

Eigenvalues and eigenvectors are fundamental concepts in linear algebra, used to understand matrix tranformations. For a square matrix A of size $n \times n$, an eigenvector $x \neq 0$ is defined as a vector that satisfies the following equation:

$$Ax = \lambda x$$

where λ denotes a scalar referred to as the eigenvalue associated with the eigenvector. This equation indicates that when the matrix A operates on the vector x, the outcome is a scaled variant of x, with λ defining the scaling factor. Importantly, eigenvectors only change in magnitude (scaling), not direction, unless the eigenvalue is negative, which causes a reverse in direction.

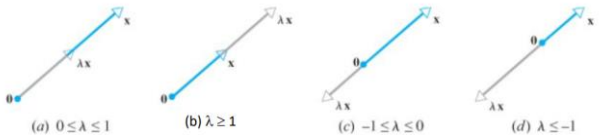

Fig. 2.1 Illustration of eigenvectors and eigenvalues in a 2D matrix tranformation. (Source:Aljabar Linear dan Geometri (Dr. Rinaldi Munir))

In other words, Eigenvalues denote the scaling factor imposed on the eigenvector during matrix transformation, while eigenvectors indicate the direction in which the matrix transformation results just in scaling, without altering direction. The eigenvalue λ quantifies this scalling:

- $\lambda > 1$: The eigenvector is stretched.
- $0 < \lambda < 1$: The eigenvector is compressed.
- $\lambda < 0$: The eigenvector's orientation is inverted.

This attribute is essential for comprehending the structure of data or systems undergoing transformation, since it emphasizes significant orientations within the transformation process.

The process of calculating eigenvalues and eigenvectors involves solving the characteristic equation

$$\det(A - \lambda I) = 0$$

where I is the identity matrix. The roots of this polynomial equation are the eigenvalues of A. For each eigenvalue, the corresponding eigenvectors are determined by solving the system

$$(A - \lambda I)x = 0$$

Diagonalization is a significant application of eigenvalues and eigenvectors. A matrix A is diagonalizable if there exists a matrix P, composed of the eigenvectors of A, such that

$$A = PDP^{-1}$$

where D is a diagonal matrix containing the eigenvalues of A.

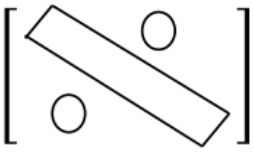

Fig. 2.2 Diagonal matrix. (Source:Aljabar Linear dan Geometri (Dr. Rinaldi Munir))

Diagonalization is especially advantageous for streamlining matrix operations, as exponentiating a diagonal matrix D is computationally efficient, allowing

$$A^k = PD^kP^{-1}$$

However, not all matrices are diagonalizable; a requisite number of linearly independent eigenvectors must be present to construct the matrix P.

B. Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) is a fundamental technique in linear algebra with extensive applications in various fields, including machine learning, data analysis, and signal processing. The SVD of a matrix provides a way to decompose it into three constituent matrices, revealing intrinsic properties such as rank, range, and null space [2]. It is especially vital in dimensionality reduction tasks, such as Principal Component Analysis (PCA), as it enables the representation of high-dimensional data in a reduced manner while preserving its fundamental attributes. Every matrix A of dimensions m×n can be decomposed as:

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

where U is an orthogonal m×m matrix comprising the left singular vectors of A, Σ is a diagonal m×n matrix including singular values, which are the square roots of the eigenvalues of A^TA or AA^T, arranged in decreasing order, and V^T: The transpose of an orthogonal n×n matrix whose columns represent the right singular vectors of A.

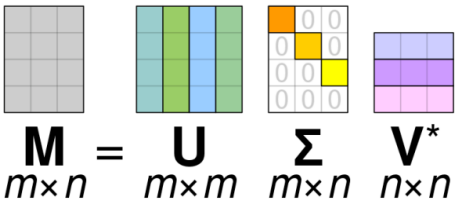

Fig. 2.3 Matrix Illustration with SVD methods. (Source:Aljabar Linear dan Geometri (Dr. Rinaldi Munir))

The SVD method begins with the normalization of the data in matrix A, ensuring that all features possess a mean of zero and a variance of one. This guarantees that discrepancies in scales among features do not skew the results. The covariance matrix $C = A^TA$ is calculated to elucidate the correlations among characteristics. The matrix SVD decomposes A into three components: U, Σ, and V^T. Here, Σ is a diagonal matrix including singular values that signify the significance of each direction in the data. U and V are orthogonal matrices that comprise the left and right singular vectors, respectively. Ultimately, data is mapped into a lower-dimensional space by choosing the greatest k singular values and their associated singular vectors, so preserving maximum variance while minimizing dimensionality and noise.

In addition to its applications in machine learning and data analysis, SVD is also instrumental in various scientific fields. For example, in spectroscopy, SVD is employed to enhance the accuracy of spectral data analysis, which is vital for understanding chemical and biochemical processes [7].

C. Principal Component Analysis (PCA)

A statistical method for feature extraction and dimensionality reduction is Principal Component Analysis (PCA). It makes it possible to represent high-dimensional data in a lower-dimensional space by identifying the directions (principal components) along which the variation in the data is largest. The first principal component accounts for the largest variance, followed by the second, and so on. This method not only aids in reducing the dimensionality of the dataset but also enhances interpretability by filtering out noise and redundant features [3].

Principal Component Analysis (PCA) begins with data normalization, which guarantees that every feature in the dataset is on the same scale. Because characteristics with greater scales have the potential to dominate the covariance matrix and produce biased results, this stage is essential. The data matrix A's features are all adjusted to have a standard deviation of one and a mean of zero. This is accomplished by deducting each feature's mean from

the values that correspond to it, and then dividing the result by the feature's standard deviation. The input for the next PCA phases is the normalized dataset, or A_{norm}.

$$A_{\text{norm}} = \frac{A - \mu}{\sigma}$$

where μ is the mean of each feature, and σ is the standard deviation.

Following normalization, the covariance matrix is used to record the relationships between the dataset's characteristics. The degree to which feature pairings fluctuate together is measured by this matrix. Whereas a negative covariance suggests an adverse link, a positive covariance shows that two traits rise together. The following formula is used to determine the covariance matrix C:

$$C = \frac{1}{m-1} A_{\text{norm}}^T A_{\text{norm}}$$

where m is the number of samples in the datasets and A_{norm} is the normalized data matrix. Since its eigenvalues and eigenvectors serve as the basis for determining main components, this covariance matrix is an essential component of PCA. PCA finds the directions in the data that capture the most variance by breaking down C.

PCA operates by identifying the principal components of a dataset, which are the directions of maximum variance. This is achieved through the Singular Value Decomposition (SVD) of the data matrix, where PCA seeks to find a low-dimensional subspace that best approximates the high-dimensional data in a least-squares sense [6].

Moreover, PCA has been adapted to handle various challenges in data processing, such as missing values and outliers. Techniques have been developed to make PCA robust against such issues, ensuring that the analysis remains valid even when the data is imperfect [5].

D. Baymax as a Robotic Healthcare

Baymax, the robotic healthcare assistant from Disney's "Big Hero 6," epitomizes the potential of robotic technology in medical settings. As a soft, friendly robot designed to provide care and comfort, Baymax represents a significant shift in the perception and functionality of healthcare robots. The integration of robotic technologies in healthcare has significantly transformed medical practices, particularly in the context of surgery, patient care, and rehabilitation. The application of robotics in these areas has been driven by the need for enhanced precision, reduced invasiveness, and improved patient outcomes.

The evolution of soft robotics represents another significant advancement in medical technology. Soft robotic devices, characterized by their compliance and adaptability, have been developed for various applications, including rehabilitation and personalized medicine [4]

III. IMPLEMENTATION

The primary objective of this project is to create an automated health diagnosis system using Principal Component Analysis (PCA), drawing inspiration from the methodology of Big Hero 6's Baymax healthcare robot. In order to provide precise and effective diagnosis, this system attempts to integrate patient medical data, including characteristics like blood pressure, glucose levels, BMI, and medical history, which are acquired from wearable sensors or medical devices. The main technique for dimensionality reduction is PCA, which helps find pertinent patterns in patient data without losing any important information in the process. In order to improve diagnosis accuracy and computational efficiency, the system also employs a classification model constructed with techniques like Random Forest to forecast a patient's health state based on the data processed by PCA.

A. Data Collection

For this experiment, we used the Pima Indians Diabetes Dataset, which consists of 768 samples with 8 features, including Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, and Age.. This dataset was chosen because of its relevance to diabetes prediction, as it contains both numerical data and a binary target variable indicating the presence (1) or absence (0) of diabetes.

```
1  https://raw.githubusercontent.com/jbrownlee/Datas
   ets/master/pima-indians-diabetes.data.csv"
2  columns = ['Pregnancies', 'Glucose', 'BloodPressu
   re', 'SkinThickness', 'Insulin', 'BMI', 'Diabetes
   PedigreeFunction', 'Age', 'Outcome']
3  df = pd.read_csv(url, names=columns)
```

Fig. 3.1 Load Dataset from Online Source.

B. Data Preprocessing

The data is preprocessed to make sure it is clean and standardized before PCA is applied. Normalization Data is normalized using *StandardScaler* to standardize all features to have a mean of 0 and a standard deviation of 1, ensuring PCA works optimally.

```python
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Fig. 3.2 Code to Normalized Data.

C. PCA Implementation

Following preprocessing, PCA is used to reduce the number of features while maintaining the majority of the variance in the data. By lowering the dimensionality of the data, PCA can help prevent overfitting and increase model efficiency. Based on the explained variance ratio, from eight components we can chose six principal components for our experiment because they were able to explain more than 90% of the dataset's variance:

```python
from sklearn.decomposition import PCA
pca = PCA(n_components=6)
X_train_pca = pca.fit_transform(X_train)
X_test_pca = pca.transform(X_test)
```

Fig. 3.3 Code to Selected 6 Principal Components.

We utilized a Scree Plot to display the cumulative variance in order to guarantee that the six components were able to capture an adequate amount of useful information:

```python
plt.figure(figsize=(10, 6))
plt.plot(range(1, len(pca.explained_variance_ratio_) + 1), pca.explained_variance_ratio_.cumsum(), marker='o', label='Cumulative Variance')
plt.axhline(y=0.9, color='r', linestyle='--', label='90% Variance Threshold')
plt.xlabel('Number of Principal Components')
plt.ylabel('Cumulative Variance Explained')
plt.title('Scree Plot - PCA')
plt.legend()
plt.grid()
plt.show()
```

Fig. 3.4 Code for Scree Plot.

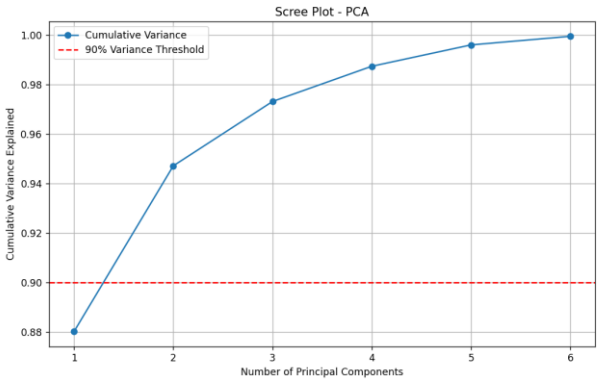

Fig. 3.5 Visualization of Scree Plot – PCA.

D. Model Training and Evaluation

Several machine learning models were built to predict diabetes after principal component analysis (PCA) was applied. Using accuracy, precision, recall, and F1-score, these models were examined to determine their performance. For the purpose of training the models, the PCA-transformed training data was utilized:

```python
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(random_state=42)
rf.fit(X_train_pca, y_train)
```

Fig. 3.6 Model Training using the PCA-transformed training data.

For evaluation, we used accuracy, precision, recall, F1-score, and confusion matrix. The evaluation results for Random Forest after applying PCA are as follows:

```python
def evaluate_model_with_pca(model, X_train_pca, y_train, X_test_pca, y_test):
    start_time = time.time()
    model.fit(X_train_pca, y_train)
    y_pred = model.predict(X_test_pca)
    end_time = time.time()
    elapsed_time = end_time - start_time
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred, average='macro')
    recall = recall_score(y_test, y_pred, average='macro')
    f1 = f1_score(y_test, y_pred, average='macro')
    return accuracy, precision, recall, f1, elapsed_time

def evaluate_model_without_pca(model, X_train, y_train, X_test, y_test):
    start_time = time.time()
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    end_time = time.time()
    elapsed_time = end_time - start_time
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred, average='macro')
    recall = recall_score(y_test, y_pred, average='macro')
    f1 = f1_score(y_test, y_pred, average='macro')
    return accuracy, precision, recall, f1, elapsed_time
```

Fig. 3.7 Evaluation Model With and Without PCA.

E. Result and Discussion

The results for each model are evaluated based on accuracy, precision, recall, and F1-score. Among all the models tested, Random Forest and XGBoost achieving the highest accuracy, both have accuracy scores of 0.77, with similarly precision, recall, and F1-score.

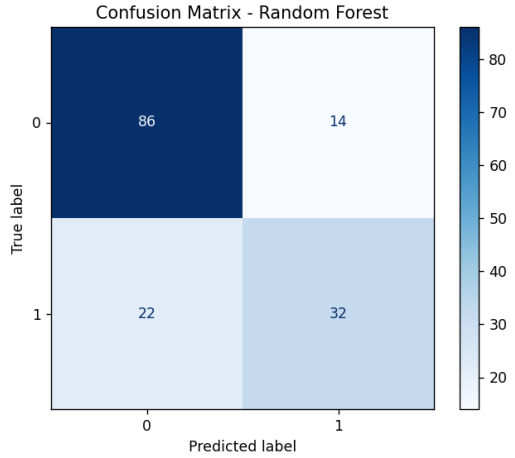

Fig. 3.8 . Random Forest Model Result

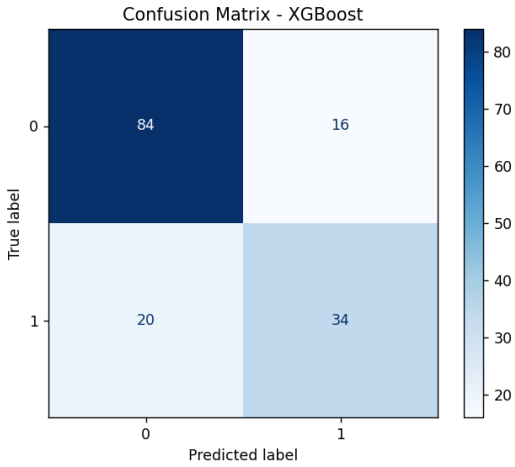

Fig. 3.9 . XGBoost Model Result

It is clear from the experimental data that PCA greatly enhanced the classification models' performance, especially with regard to accuracy, precision, recall, and F1-score. All models, including Random Forest and XGBoost, showed this. The PCA-trained model consistently performed better than the non-PCA-trained model.

- Accuracy: Compared to the model without PCA, which had an accuracy of 0.74, the XGBoost model with PCA had the highest accuracy (0.77). This suggests that the model was able to better generalize and detect patterns in the dataset by lowering the dimensionality of the data through PCA, especially when working with intricate, high-dimensional characteristics.
- Precision: All models showed comparable gains in precision, with Random Forest exhibiting the highest precision (0.74) when PCA was included. This indicates that PCA increased the model's overall accuracy while also lowering false positives, increasing its dependability in predicting positive cases (diabetes).
- Recall: Recall, which gauges the model's accuracy in identifying all pertinent examples, also improved. Better recall scores were demonstrated by the PCA-based model, which reduced false negatives and increased the model's sensitivity to diabetic cases.
- F1-Score: This metric, which weighs recall and precision, also showed that models with PCA performed better than those without. For instance, the XGBoost model with PCA produced a higher F1-score of 0.74 than the model without PCA (0.71).

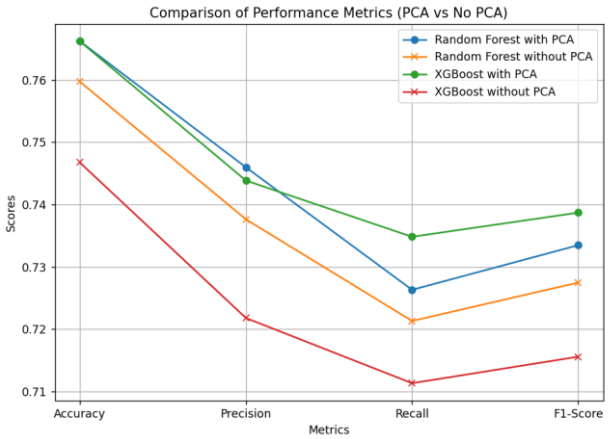

Fig. 3.10 . Comparison of Performance Model with PCA and without PCA

IV. CONCLUSION

This study demonstrates the advantages of Principal Component Analysis (PCA) for dimensionality reduction in classification tasks, specifically in predicting diabetes with the Pima Indians Diabetes Dataset. PCA has demonstrated its utility in enhancing the performance metrics—accuracy, precision, recall, and F1-score of classification models such as Random Forest and XGBoost.

PCA facilitated dimensionality reduction, enabling models to concentrate on the most pertinent features, thereby enhancing accuracy and F1-score. The XGBoost model incorporating PCA attained the highest accuracy of 0.77, in contrast to the 0.74 accuracy of the model without PCA, demonstrating PCA's efficacy in enhancing predictive performance. The models trained with PCA demonstrated enhanced precision and recall, indicating superior performance in accurately identifying diabetes cases while minimizing false positives and false negatives.

The enhancements noted in accuracy, precision, recall, F1-score, and computation time establish PCA as a crucial method for managing high-dimensional datasets, particularly in medical diagnostics. The capacity of PCA

to eliminate redundant features while preserving essential information markedly diminishes the likelihood of overfitting, thereby enhancing the model's generalization to novel data.

This study highlights the significance of dimensionality reduction in the development of effective and efficient automated medical diagnostic systems. The application of PCA enhanced the predictive accuracy of the models and improved computational efficiency, establishing it as a crucial tool for healthcare applications requiring timely and accurate diagnoses. The findings underscore the significance of PCA in the advancement of Baymax-inspired health diagnostic systems, facilitating quicker and more precise predictions for patients with diabetes and various medical conditions.

V. Suggestions

This study illustrates the benefits of Principal Component Analysis (PCA) in enhancing model performance for automated health diagnostics. Future research should concentrate on the integration of PCA into real-time health diagnostic systems, exemplified by Baymax-inspired health scanners. PCA effectively diminishes the complexity of data obtained from wearable sensors, facilitating quicker and more precise real-time diagnosis. Future research may investigate the use of PCA in analyzing multimodal sensor data, integrating information from various devices (e.g., glucose meters, heart rate monitors, blood pressure cuffs) to develop a comprehensive diagnostic platform for continuous health monitoring.

Future research should focus on improving model interpretability alongside enhancing real-time performance. PCA facilitates dimensionality reduction; however, it is essential for healthcare professionals to comprehend the mechanisms behind AI model predictions. Integrating PCA with explainable AI techniques such as LIME or SHAP provides clear and interpretable insights into model predictions, enhancing their trustworthiness and applicability in medical contexts.

Expanding the dataset and incorporating more complex models, such as deep learning, should be prioritized. The growing accessibility of medical imaging, patient history, and genetic data suggests that future systems may leverage advanced machine learning models to manage extensive and varied datasets, thereby enhancing the accuracy and robustness of the models. This may enhance diagnostic efficacy across a wider array of conditions, thereby increasing the system's utility in preventive and real-time healthcare applications.

VI. Appendix

Program applied in this paper is visible here.

VII. Acknowledgment

All praise to Allah Subhanahu wa Ta'ala, for His blessing, which have guided me to complete this paper succesfully. I would also like to thank to lecturer for IF2123 Linear and Geometric Algebra, Dr. Ir. Rinaldi Munir, M.T. for the motivation, knowledge, and guidance throughout the course. I hope that this paper will provide valuable insight and benefits to its readers.

References

[1] R. Munir, "Nilai Eigen dan Vektor Eigen (Bagian 1),". [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/AljabarGeometri/2023-2024/Algeo-19-Nilai-Eigen-dan-Vektor-Eigen-Bagian1-2023.pdf. [Accessed: Jan. 1, 2025]

[2] Zhang, Z. (2015). The singular value decomposition, applications and beyond. [Online]. Available: https://doi.org/10.48550/arxiv.1510.08532. [Accessed: Jan. 1, 2025].

[3] Hasan, B. and Abdulazeez, A. (2021). A review of principal component analysis algorithm for dimensionality reduction. Journal of Soft Computing and Data Mining, 02(01). [Online]. Available: https://doi.org/10.30880/jscdm.2021.02.01.003. [Accessed: Jan. 1, 2025]

[4] Garcia, L., Kerns, G., O'Reilley, K., Okesanjo, O., Lozano, J., Narendran, J., … & Golecki, H. (2021). The role of soft robotic micromachines in the future of medical devices and personalized medicine. Micromachines, 13(1), 28. [Online]. Available: https://doi.org/10.3390/mi13010028. [Accessed: Jan. 2, 2025]

[5] Severson, K., Molaro, M., & Braatz, R. (2017). Principal component analysis of process datasets with missing values. Processes, 5(3), 38.). [Online]. Available: https://doi.org/10.3390/pr5030038. [Accessed: Jan. 2, 2025]

[6] Tsagkarakis, N., Markopoulos, P., Sklivanitis, G., & Pados, D. (2018). L1-norm principal-component analysis of complex data. Ieee Transactions on Signal Processing, 66(12), 3256-3267. [Online]. Available: https://doi.org/10.1109/tsp.2018.2821641C. [Accessed: Jan. 2, 2025] J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.

[7] Galo, A. and Colombo, M. (2013). Singular value decomposition and ligand binding analysis. Journal of Spectroscopy, 2013, 1-7. [Online]. Available: https://doi.org/10.1155/2013/372596. [Accessed: Jan. 2, 2025]

Statement

I hereby declare that this paper I have written is my own work, not a paraphrase or translation of someone else's paper, and not plagiarism.

Bandung, 2 January 2025

Muhammad Edo Raduputu Aprima - 13523096