**Project Report**

**Smart power management for battery operated devices**

CONTENTS

|  |  |
| --- | --- |
| **TOPIC** | **PAGE NUMBER** |
| ABSTRACT | 3 |
| INTRODUCTION | 4 |
| METHODOLOGY | 5 |
| IMPLEMENTATION AND RESULT | 9 |
| CONCLUSION | 16 |
| REFERENCES | 17 |

**ABSTRACT**

In modern computing, ensuring optimal system performance is a critical challenge. Computers generate vast amounts of performance data from various hardware and software components, making it difficult to analyze and optimize resource utilization effectively. This project focuses on leveraging **machine learning techniques** to process and cluster **computer performance datasets**, thereby uncovering patterns that can aid in predictive maintenance, power optimization, and anomaly detection.

The proposed approach involves a multi-step methodology, beginning with **data preprocessing** to handle missing values, remove outliers, and engineer new meaningful features. Since raw performance data is often high-dimensional and redundant, **Principal Component Analysis (PCA)** is employed for dimensionality reduction while retaining critical information. Clustering is performed using **K-Means**, an unsupervised learning algorithm that groups data points based on their similarities. The effectiveness of the clustering model is evaluated using **Silhouette Score** and **Davies-Bouldin Index**, which assess the cohesion and separation of the formed clusters.

The results indicate that the optimized feature set and clustering approach effectively classify different performance states of a computer system, allowing for more efficient monitoring and management. By utilizing these techniques, system administrators and developers can enhance **resource allocation, power efficiency, and fault detection**. The study highlights the potential of machine learning in automating **performance optimization** and improving overall **computing efficiency**.

**INTRODUCTION**

With the increasing complexity of modern computing systems, optimizing performance while ensuring energy efficiency has become a crucial challenge. Computers, ranging from personal laptops to large-scale data centers, generate extensive system performance data that can be leveraged to analyze resource consumption patterns, detect inefficiencies, and optimize power usage. Traditional methods for monitoring system health and performance often rely on static threshold-based techniques, which fail to adapt to dynamic workloads and evolving system behaviors.

This study focuses on an unsupervised learning approach to analyzing computer performance data. By leveraging clustering techniques, we aim to identify patterns in system behavior, classify performance states, and detect anomalies that could indicate inefficiencies or potential failures. The dataset contains various system parameters such as CPU utilization, memory usage, power consumption, and thermal metrics. Preprocessing steps such as feature engineering, outlier removal, and dimensionality reduction are applied to enhance data quality and improve model performance.

The primary goal of this project is to use **K-Means clustering** along with **Principal Component Analysis (PCA)** to group similar performance states and evaluate how effectively the model distinguishes between normal and abnormal system conditions. The performance of the clustering model is assessed using metrics such as the **Silhouette Score** and **Davies-Bouldin Index**, ensuring that the identified clusters are well-separated and meaningful. This study aims to contribute to **intelligent system monitoring and optimization**, enabling more adaptive and data-driven approaches for managing computer performance.

**METHODOLOGY**

**1. Data Preprocessing and Feature Engineering**

**1.1 Data Cleaning and Preprocessing**

The dataset consists of system performance parameters such as CPU utilization, memory usage, power consumption, and thermal characteristics. The preprocessing steps included:

* **Irrelevant Feature Removal:** Columns such as timestamps, device identifiers, and non-informative categorical attributes were discarded.
* **Handling Missing Values:** Missing data points were imputed using the **median** to preserve statistical consistency.
* **Feature Encoding:** Categorical attributes were **one-hot encoded** for numerical representation.

**1.2 Feature Engineering**

To enhance data quality, additional features were engineered:

* **Discharge Rate:**
* **Power Efficiency:**
* **Memory Utilization Ratios:** Computed as ​ for both RAM and ROM.
* **CPU Frequency Aggregation:** Averaged core frequency readings into a single **average frequency** feature.
* **Screen Power Impact:**

Screen\_Status × Brightness ×

These engineered features provided a more meaningful representation of system performance states.

**1.3 Outlier Detection and Removal**

Outliers were identified and removed using the **Interquartile Range (IQR) method** to prevent extreme values from skewing clustering performance.

**1.4 Feature Scaling**

All numerical features were **normalized using MinMax Scaling** to bring values into the range [0,1].

**2. Dimensionality Reduction Using Principal Component Analysis (PCA)**

Given the dataset’s high dimensionality, **Principal Component Analysis (PCA)** was applied to extract the most informative features.

* PCA transformed the dataset into **two principal components (PC1, PC2)** while preserving maximum variance.
* This step improved clustering efficiency by reducing redundant correlations.

**3. Clustering Using K-Means**

**3.1 Choice of Algorithm**

K-Means clustering was selected due to its computational efficiency and ability to partition data into distinct performance states. The **Euclidean distance metric** was used for cluster assignments.

**3.2 K-Means Clustering Process**

1. **Initialization:** Cluster centroids were randomly selected.
2. **Assignment Step:** Each data point was assigned to the nearest centroid using **Euclidean distance**.
3. **Update Step:** Centroids were recalculated based on the mean of assigned points.
4. **Convergence:** Steps 2-3 were repeated until centroids stabilized.

**3.3 Determining the Optimal Number of Clusters**

The number of clusters (K) was determined using the **Elbow Method**, which analyses the rate of decrease in within-cluster variance.

**4. Model Evaluation Metrics**

**4.1 Silhouette Score**

Silhouette Score assesses how well-separated clusters are, calculated as:

Where:

* **a** is the average intra-cluster distance.
* **b** is the average nearest-cluster distance.
* Higher values indicate **better clustering performance**.

**4.2 Davies-Bouldin Index (DBI)**

Davies-Bouldin Index measures the compactness and separation of clusters:

Where:

* σi ​ is the dispersion of cluster iii.
* d(ci,cj) is the distance between centroids ci and cj
* Lower values indicate **better-defined clusters**.

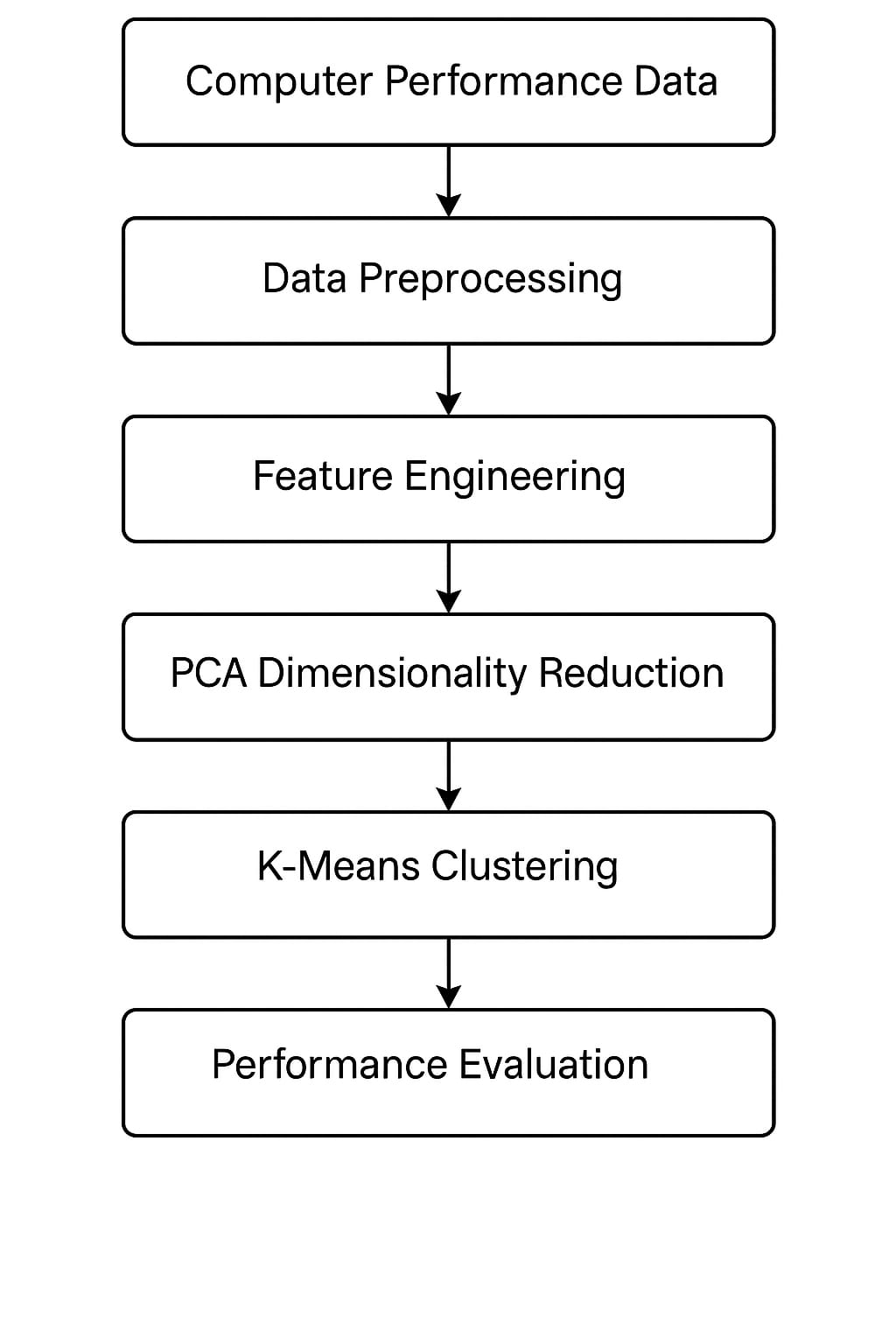
 The block diagram in **Fig. 1** represents the workflow of the clustering-based performance analysis system. It begins with **data preprocessing**, followed by **feature selection using PCA** to reduce dimensionality. The **elbow method** determines the optimal number of clusters, and **K-Means clustering** is applied. The model's performance is evaluated using **Silhouette Score** and **Davies-Bouldin Index**, ensuring meaningful cluster formation. Finally, the results are analyzed through visualizations to gain insights into system performance.

Fig. 1: Block Diagram

**IMPLEMENTATION AND RESULTS**

**1. Feature Selection and Preprocessing**

**1.1 Feature Distribution Analysis**

* Histograms were generated for all features to examine their distribution.
* This visualization helped identify skewed data, potential outliers, and the need for feature scaling.

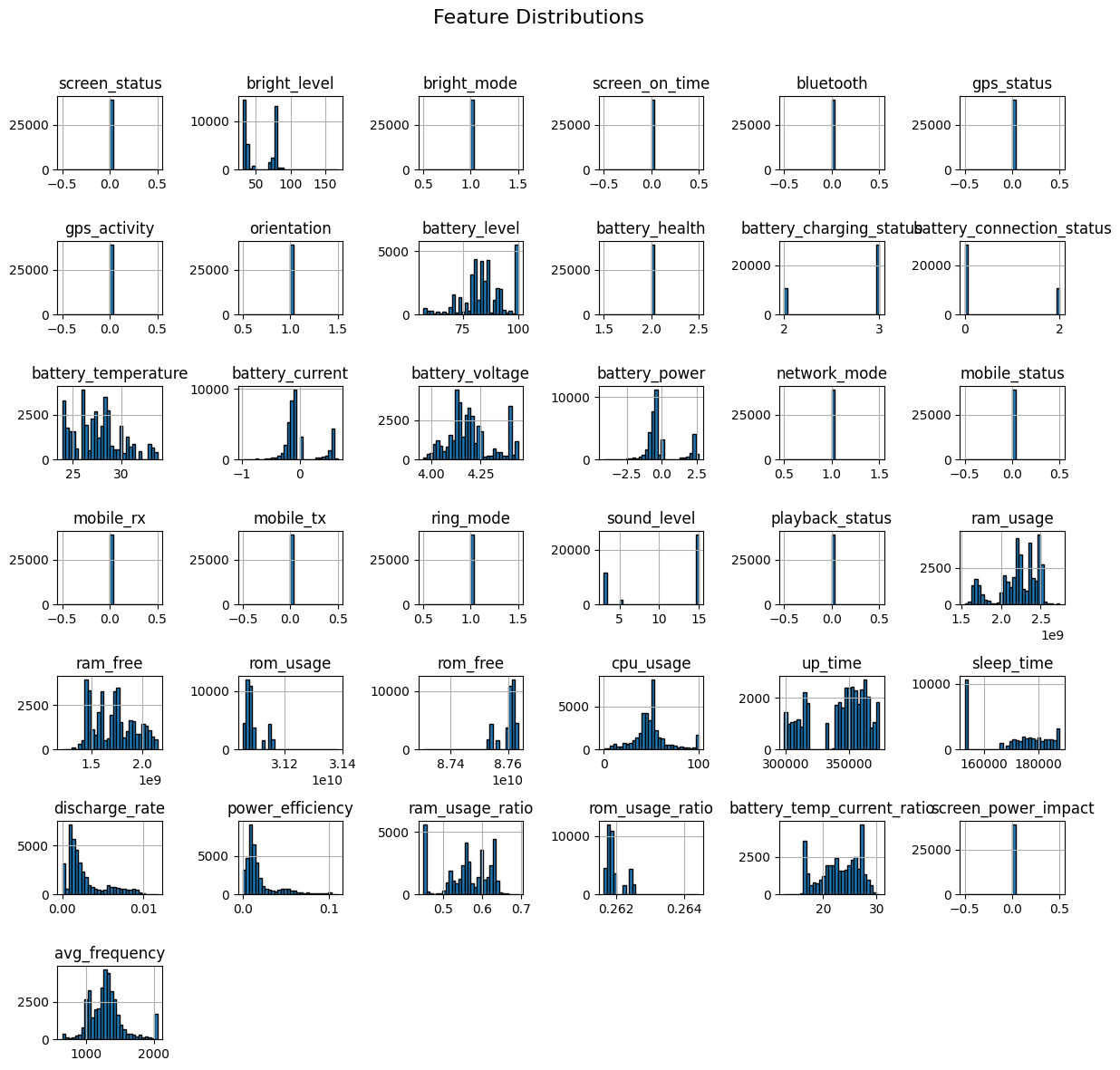


Fig. 2: Distribution of features in the dataset.

**1.2 Correlation Analysis**

* A heatmap was generated to analyze the correlation between features.
* Highly correlated features were identified to prevent redundancy and improve clustering accuracy.

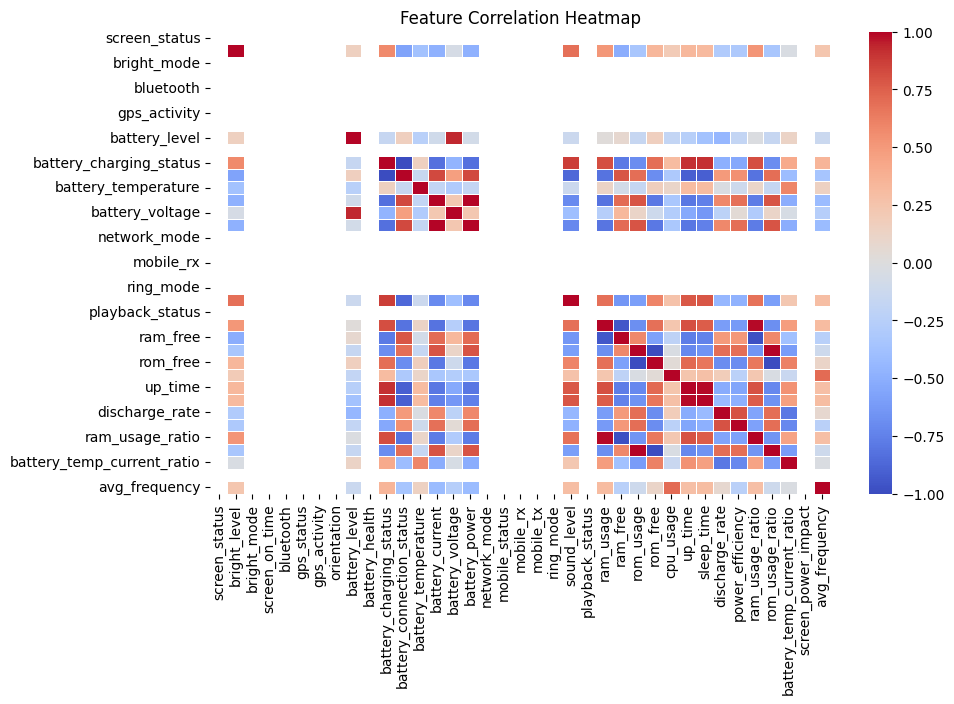


Fig. 3: Heatmap showing correlation between features.

**1.3 Principal Component Analysis (PCA) for Dimensionality Reduction**

* PCA was applied to reduce dataset complexity while retaining the most significant features.
* The top 10 features contributing the most to the first principal component were identified and visualized.

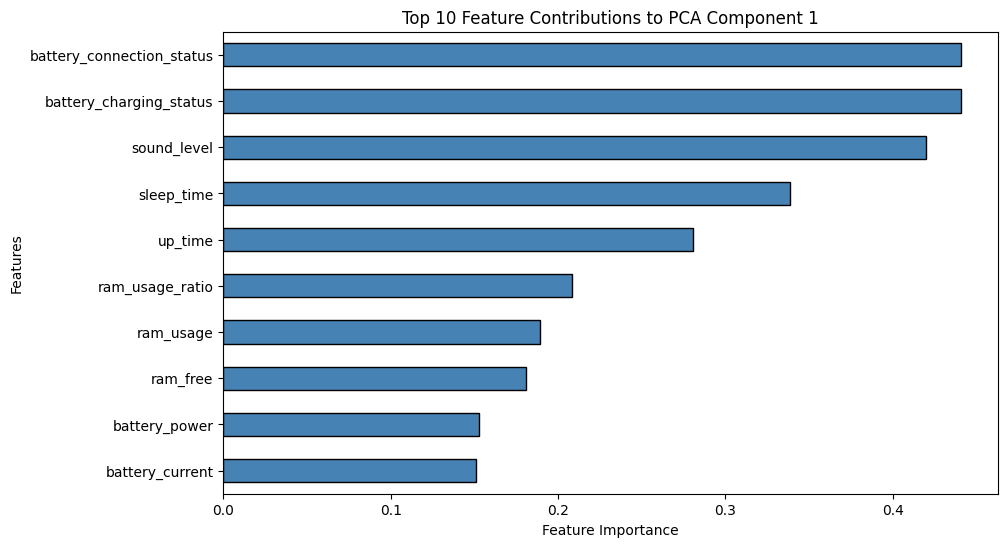


Fig. 4: Most significant features contributing to the first principal component in PCA.

**2. Selection of Optimal Clusters (K) and PCA Components**

**2.1 Elbow Method for Optimal K**

* The optimal number of clusters was determined using the Elbow Method, which analyzes the Within-Cluster Sum of Squares (WCSS).
* A plot was generated to identify the point where WCSS stabilizes, indicating the best K value.

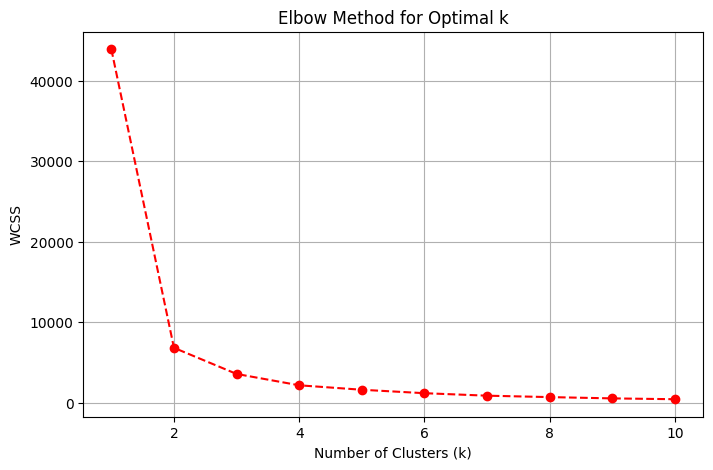


Fig. 5: Elbow method used to determine the optimal number of clusters (k).

**2.2 Scree Plot for PCA Component Selection**

* A cumulative explained variance plot was generated to determine the number of principal components required to retain maximum variance.
* This helped in selecting the minimum number of features necessary for effective clustering.

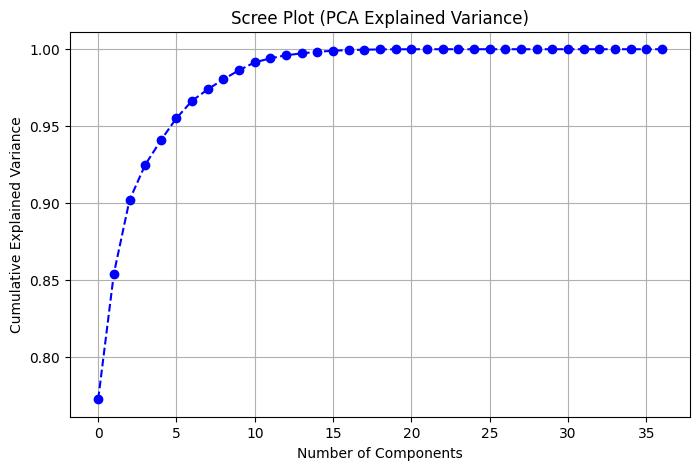


Fig. 6: Scree plot showing the cumulative explained variance by PCA components.

**3. Clustering and Visualization**

**3.1 K-Means Clustering**

* K-Means clustering was performed on the PCA-transformed data using the optimal K value obtained from the Elbow Method.
* The clusters were visualized in a scatter plot, showing data distribution in a reduced dimensional space.

**3.2 Cluster Centers and Assignments**

* The centroids of the clusters were plotted to highlight the separation between groups.
* The final clustered data was analyzed to interpret variations in CPU performance across different clusters.

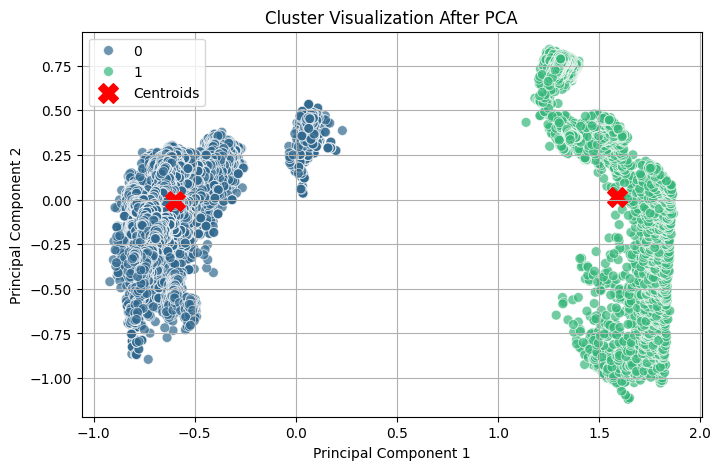


Fig. 7: 2D representation of clusters obtained after applying PCA.

**4. Evaluation Metrics and Performance Analysis**

**4.1 Silhouette Score**

* The Silhouette Score was calculated to measure the separation between clusters.
* A higher score indicates well-separated and meaningful clusters.

**4.2 Davies-Bouldin Score**

* The Davies-Bouldin Index was computed to evaluate cluster compactness and separation.
* Lower values indicate better-defined clusters.

**5. Results**

The clustering model was evaluated using key performance metrics to assess the quality of the formed clusters. The obtained scores are summarized in below(Table 1):

**Evaluation Metrics for Clustering Performance**

|  |  |  |
| --- | --- | --- |
| **Metric** | **Score** | **Interpretation** |
| Silhouette Score | 0.7959 | Higher values indicate well-defined clusters. |
| Davies-Bouldin Index | 0.3800 | Lower values indicate better separation between clusters. |

**Table 1: Clustering Evaluation Metrics**  
*Silhouette Score and Davies-Bouldin Index for model performance assessment.*

The Silhouette Score suggests that the clusters are well-separated and cohesive, while the Davies-Bouldin Index confirms minimal overlap between clusters, indicating good clustering performance.

**CONCLUSION**

This project presented an unsupervised learning approach to analyze computer performance data using clustering techniques. We employed feature selection and principal component analysis (PCA) to reduce dimensionality and extract the most significant features. The optimal number of clusters was determined using the elbow method, ensuring efficient data segmentation. K-Means clustering was then applied to group similar performance patterns, providing insights into system behaviour.

The evaluation of clustering effectiveness was conducted using the **Silhouette Score** and **Davies-Bouldin Index**, where the results indicated well-defined clusters with minimal overlap. The **Silhouette Score of 0.7959** suggests that the clusters are well-separated, while the **Davies-Bouldin Index of 0.3800** confirms compact and distinct cluster formations.

The study demonstrates that clustering can effectively categorize system performance patterns, aiding in anomaly detection, workload optimization, and resource allocation. Future work could explore hybrid clustering approaches, incorporating domain-specific heuristics or supervised learning for refined classification. Additionally, integrating real-time monitoring can enhance adaptability and responsiveness to performance fluctuations.

This research provides a foundation for further advancements in performance analytics, contributing to more efficient computing environments through intelligent workload management and data-driven decision-making.

**REFERENCES**

1. Homayounfar, M.; Malekijoo, A.; Visuri, A.; Dobbins, C.; Peltonen, E.; Pinsky, E.; Teymourian, K.; Rawassizadeh, R. Understanding Smartwatch Battery Utilization in the Wild. *Sensors* **2020**, *20*, 3784. <https://doi.org/10.3390/s20133784>
2. LEE, Jong-Yong; LEE, Daesung. K-means clustering-based WSN protocol for energy efficiency improvement. **International Journal of Electrical and Computer Engineering (IJECE)**, [S.l.], v. 11, n. 3, p. 2371-2377, june 2021. ISSN 2722-2578. Available at: <<https://ijece.iaescore.com/index.php/IJECE/article/view/23903>>. Date accessed: 31 mar. 2025. doi:<http://doi.org/10.11591/ijece.v11i3.pp2371-2377>.
3. B. Zhu, E. Bedeer, H. H. Nguyen, R. Barton and J. Henry, "Improved Soft-k-Means Clustering Algorithm for Balancing Energy Consumption in Wireless Sensor Networks," in IEEE Internet of Things Journal, vol. 8, no. 6, pp. 4868-4881, 15 March15, 2021, doi: 10.1109/JIOT.2020.3031272.
4. Homayounfar, M.; Malekijoo, A.; Visuri, A.; Dobbins, C.; Peltonen, E.; Pinsky, E.; Teymourian, K.; Rawassizadeh, R. Understanding Smartwatch Battery Utilization in the Wild. *Sensors* **2020**, *20*, 3784. https://doi.org/10.3390/s20133784