Improving Intrusion Detection in IoT Networks Using Median PCA for Robust Dimensionality Reduction

Elkhadir Zyad(✉) and Achkari Begdouri Mohammed

SIGL Research Laboratory, National School of Applied Sciences of Tetouan (ENSATE), Abdelmalek Essaâdi University, Tetouan, Morocco
{z.elkhadir,m.achkaribegdouri}@uae.ac.ma

Abstract. Intrusion Detection Systems (IDS) are crucial for maintaining security in Internet of Things (IoT) environments. However, the high-dimensional nature of IoT network data and the presence of numerous outliers pose significant challenges. Traditional Principal Component Analysis (PCA) has shown promise in reducing data dimensionality but suffers from sensitivity to outliers due to its reliance on the arithmetic mean. To address this issue, we propose Median PCA (MedPCA), a robust variant that replaces the mean with the median, enhancing resistance to outliers. Extensive experiments conducted on the IoT23 and CICIoT2023 datasets demonstrate that Median PCA achieves superior accuracy and reduced CPU time compared to traditional PCA, making it a robust and efficient solution for enhancing IDS performance in large-scale IoT networks.

Keywords: Internet of Things (IoT) · PCA · IDS · KNN · Median · Outliers

1 Introduction

The proliferation of the Internet of Things (IoT) has led to a substantial increase in the number of connected devices, offering significant benefits but also introducing new security challenges. Intrusion Detection Systems (IDS) are crucial for safeguarding IoT networks from malicious activities. However, the high-dimensional nature of IoT network data, complicates the task of effective intrusion detection.

To mitigate this challenge, various studies have explored the application of numerous feature extraction methods such as Principal Component Analysis (PCA) [1], Linear Discriminant Analysis (LDA) [2], and the combination of these methods. For instance, in the study [3], PCA, LDA, and Bayesian classification are combined to construct the PCA-LDA-BC classification algorithm, establishing an intrusion detection model. Simulation experiments using the CICIDS2017

A. B. Mohammed—Contributing authors.

C. Iwendi et al. (Eds.): ICACTCE 2024, LNNS 1312, pp. 57–66, 2025.
https://doi.org/10.1007/978-3-031-94620-2_6

dataset demonstrate that this algorithm improves detection rates while reducing false detection and missed alarm rates compared to traditional naive Bayesian classification.

In the article [4], an improved LDA-based Extreme Learning Machine (ELM) classification algorithm for intrusion detection (ILECA) is proposed. The improved LDA is used for feature dimension reduction, followed by classification using a single hidden layer neural network ELM algorithm, enhancing detection accuracy.

In another study proposed by [5], the authors evaluate the efficiency of PCA for intrusion detection, focusing on its reduction ratio, the ideal number of principal components for effective detection, and the impact of noisy data on PCA. Experiments on the KDD CUP and UNB ISCX datasets reveal that the first ten principal components are effective for classification.

Moreover, the study proposed by [6] compares PCA and Kernel PCA (KPCA) methods for intrusion detection. After dimensionality reduction, data samples are classified using k-nearest neighbor (KNN) or decision tree algorithms to identify normal or anomalous network connections. Results on the KDDcup99 and NSL-KDD datasets show that KPCA with the power kernel outperforms PCA, especially for detecting denial-of-service (DOS) and probing attacks. The KPCA method also demonstrated superior performance with the decision tree classifier using the spherical kernel.

In [7], various preprocessing techniques, such as robust scaling and encoding, were implemented. The experimental outcomes demonstrated a notable enhancement in detecting DDoS attacks on IoT devices when combining PCA with the Robust Scaler. Specifically, the Random Forest and KNN classifiers delivered outstanding results, with accuracies of 99.87% and 99.14%, respectively, while the Naïve Bayes classifier performed comparatively lower with an accuracy of 87.14%. These results offer valuable insights into strengthening IoT device security against DDoS threats. The approach emphasizes the crucial role of effective preprocessing methods in developing resilient intrusion detection systems for IoT ecosystems.

Despite the promising results of traditional PCA and LDA, they remain sensitive to outliers due to their reliance on the arithmetic mean. To address this limitation, the papers [8–10] proposed to use general mean, trimmed/truncated mean and optimal mean. In this paper we propose Median PCA (MedPCA), which replaces the mean with the median, thereby enhancing robustness to outliers. We evaluate the effectiveness of MedPCA against PCA using the IoT23 and CICIoT2023 datasets, demonstrating its superior accuracy and reduced CPU time.

This paper is organized as follows: Sect. 2 describes the proposed IDS mechanism. Section 3 discusses the datasets. Section 4 revisits Principal Component Analysis (PCA) and introduces Median PCA (MedPCA). Section 5 presents and analyzes the results. Finally, Sect. 6 concludes the paper and outlines future research directions.

2 The Proposed Approach

The proposed Intrusion Detection System (IDS) operates in two primary phases: training and testing.

2.1 Training Phase

- **Data Collection:** Data is gathered from various IoT devices.
- **Preprocessing:** This includes data cleaning and imputation to replace any missing values.
- **Scaling:** Standard scaling methods are employed to ensure all features are on a comparable scale, preventing any single feature from dominating due to its magnitude.
- **Feature Extraction:** Principal Component Analysis (PCA) or Median PCA (MedPCA) is applied to extract the principal components, resulting in training data with reduced dimensionality. Detailed explanations of PCA and MedPCA are provided in Sect. 4.

2.2 Testing Phase

- **Data Collection:** New data is collected from the same devices.
- **Preprocessing and Scaling:** The same preprocessing and scaling techniques used in the training phase are applied to the new data.
- **Projection:** The scaled data is projected onto the reduced space using the previously extracted principal components.
- **Classification:** The testing data is compared to the training data within this reduced space using the K-Nearest Neighbors (KNN) algorithm.

This systematic approach ensures consistency between the training and testing phases, enhancing the IDS's ability to accurately detect anomalies in IoT networks.

3 Datasets

3.1 Iot-23

This study employs IoT23 dataset as a first simulation dataset, as detailed in [11], this dataset was published in January 2020. It contains network traffic data from three different smart home IoT devices: Amazon Echo, Philips HUE, and Somfy Door Lock. The development of machine learning algorithms was the principal purpose behind the design of this dataset. The latter contains both real and labeled instances of IoT malware infections as well as benign traffic. In addition it includes 23 captures or scenarios, with 20 being malicious and 3 benign.

The dataset tags each capture from infected devices with the specific executed malware sample. The IoT-23 dataset includes various malware categories. Given

the large scale of the dataset, we selected a smaller sample of records from each capture and combined them into a new dataset. This approach enables our computational resources to handle the processing load more effectively, while still retaining most of the attack types present in the original IoT-23 dataset.

3.2 CIC IoT 2023

This study utilizes the CIC IoT 2023 data collection as the second simulation dataset. This dataset was recently made available by the Canadian Institute for Cybersecurity [12]. The primary motivation behind creating this dataset was to enable security analytics tools tailored for real-world IoT settings. It offers a unique collection of IoT attack data, consisting of 33 distinct attack types carried out across an IoT system of 105 connected devices. The attacks are classified into seven types: Web-based, Brute Force, Spoofing, Mirai, DDoS, DoS and Recon.

4 Approaches

This section outlines the methodologies used for feature extraction and dimensionality reduction in our study. We begin with an overview of the traditional Principal Component Analysis (PCA) method and then introduce our proposed enhancement, Median PCA (MedPCA). Detailed mathematical formulations for both techniques are provided, along with a discussion on why the median-based approach offers superior robustness against outliers compared to the mean-based PCA.

4.1 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a mathematical method employed to reduce the dimensionality of data. It achieves this by converting the original high-dimensional dataset into a lower-dimensional representation, using a set of orthogonal axes known as principal components. The procedure for performing PCA consists of the following steps:

1. **Standardization**: Adjust the data by subtracting the mean of each feature to center it.
2. **Covariance Matrix Calculation**: Compute the covariance matrix derived from the normalized dataset.
3. **Eigenvalue Decomposition**: Decompose the covariance matrix into characteristic values and their associated vectors.
4. **Principal Components Selection**: Select the leading k vectors corresponding to the highest eigenvalues to define the principal components (PCs).

Mathematically, let $\mathbf{X}$ be the standardized data matrix with s samples and d features. The covariance matrix $\mathbf{C}$ is given by:

$$\mathbf{C} = \frac{1}{s-1}\mathbf{X}^T\mathbf{X}$$

The eigenvalue decomposition of $\mathbf{C}$ is:

$$\mathbf{C} = \mathbf{V}\mathbf{D}\mathbf{V}^T$$

where $\mathbf{V}$ is the matrix of eigenvectors and $\mathbf{D}$ is the diagonal matrix of eigenvalues. The top k eigenvectors form the principal components.

While PCA is effective in reducing dimensionality, its reliance on the mean makes it sensitive to outliers, which can skew the results.

4.2 Median PCA (MedPCA)

Median PCA (MedPCA) addresses the sensitivity of PCA to outliers by replacing the mean with the median in the standardization step. The median, defined as the middle value of a sorted dataset, is a robust measure of central tendency that is less affected by extreme values. This property makes MedPCA more resilient to outliers compared to traditional PCA. The steps for MedPCA are similar to PCA, with the key difference being the use of the median instead of the mean:

1. **Standardization**: Center the data by subtracting the median of each feature. To compute the median for each feature vector $\mathbf{x}_j$, sort the observations $x_{1j}, x_{2j}, \ldots, x_{sj}$ in ascending order. If the number of samples s is odd, the median $\tilde{x}_j$ is the middle value:

$$\tilde{x}_j = x_{\left(\frac{s+1}{2}\right)j}$$

   If s is even, the median is the average of the two middle values:

$$\tilde{x}_j = \frac{1}{2}\left(x_{\left(\frac{s}{2}\right)j} + x_{\left(\frac{s}{2}+1\right)j}\right)$$

   where $x_{\left(\frac{s}{2}\right)j}$ and $x_{\left(\frac{s}{2}+1\right)j}$ are the middle elements of the sorted vector.
2. **Covariance Matrix Computation**: Compute the covariance matrix using the median-centered data.
3. **Eigenvalue Decomposition**: Decompose the covariance matrix to derive the eigenvalues and eigenvectors.
4. **Principal Components Selection**: Select the k most important eigenvectors corresponding to the largest eigenvalues to form the principal components.

Let $\mathbf{X}$ be the matrix of median-centered data with s samples and d features. The covariance matrix $\mathbf{C}$ is computed as follows:

$$\mathbf{C} = \frac{1}{s-1}\mathbf{X}^T\mathbf{X}$$

The eigenvalue decomposition of $\mathbf{C}$ is:

$$\mathbf{C} = \mathbf{V}\mathbf{D}\mathbf{V}^T$$

where $\mathbf{V}$ is the matrix of eigenvectors and $\mathbf{D}$ is the diagonal matrix of eigenvalues. The top k eigenvectors define the principal components.

Using the median for data centering in MedPCA provides robustness against outliers, leading to more reliable dimensionality reduction and improved performance in applications like Intrusion Detection Systems (IDS).

5 Experiments and Results

In our experiments, we employed a random selection of 10,000 normal data samples and 1,000 malicious data samples for the training phase. For testing, we utilized 5,000 normal data samples and 1,000 malicious data samples. This random selection process was iterated 20 times to ensure comprehensive and robust evaluation.

Our main evaluation criteria included accuracy, which gauges the percentage of correctly classified instances, and CPU time, which reflects the computational efficiency across different PCA variants.

The formula for accuracy is given by:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Instances}}$$

where:

True Positives (TP) : Instances correctly identified as intrusions.

True Negatives (TN) : Instances correctly identified as normals.

Total Instances : TP + TN + False Positives + False Negatives.

The experimental results are structured into two parts. The first part investigates how varying the number of principal components (PC) while maintaining k at 1 affects IDS accuracy and CPU time. The second part assesses the influence of changing the number K (of KNN) on IDS accuracy and CPU time, with the number of principal components (PC) held constant at 3.

Figures 1 and 2 concern the first part of experiments, on the other hand Figs. 3 and 4 illustrate the second part.

For the IoT23 dataset (Fig. 1), we can observe several key insights regarding the performance of MedPCA and traditional PCA when used in conjunction with a K-Nearest Neighbors (KNN) classifier. The top panel of Fig. 1 shows that the MedPCA + KNN configuration consistently maintains a high accuracy around 0.90, regardless of the number of components. This stability indicates that MedPCA is highly robust and less sensitive to changes in the number of components, effectively managing the presence of outliers in the data. In contrast, the accuracy of the PCA + KNN configuration fluctuates more significantly, peaking at three components with an accuracy of approximately 0.88 before declining as the number of components increases. This variability suggests that traditional PCA is more susceptible to the influence of outliers and the selection of the number of components.

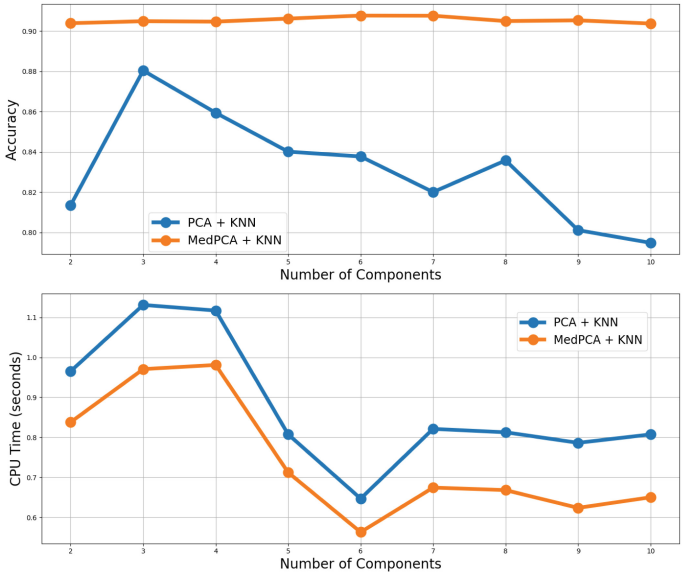

Fig. 1. Number of PC vs Accuracy and CPU time for iot23 dataset

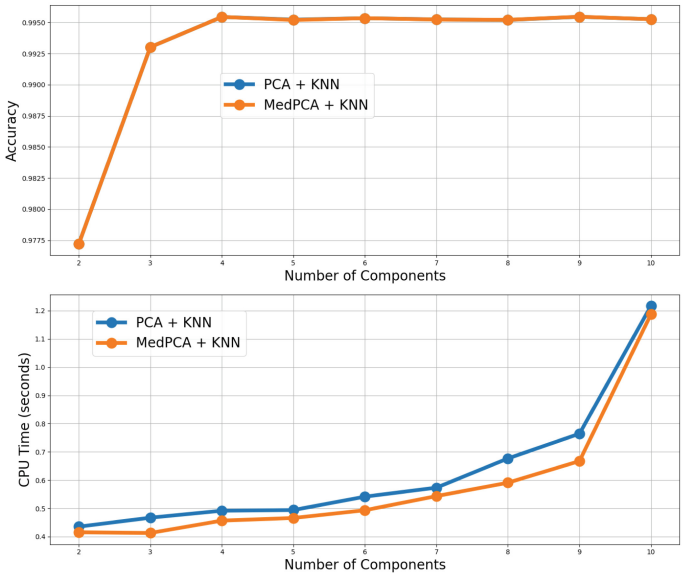

Fig. 2. Number of PC vs Accuracy and CPU time CICIOT23 dataset

The bottom panel of Fig. 1 illustrates the CPU time required for both PCA variants. MedPCA + KNN demonstrates a lower and more stable CPU time across different numbers of components, highlighting its computational efficiency.

Traditional PCA, on the other hand, exhibits greater variability in CPU time, peaking at around four components before stabilizing. This suggests that traditional PCA is less predictable and often more computationally intensive than MedPCA.

For CIC IoT 2023 dataset (Fig. 2), we can draw several important conclusions regarding the performance of MedPCA and traditional PCA when used in conjunction with a K-Nearest Neighbors (KNN) classifier. The top panel of Fig. 2 demonstrates that MedPCA + KNN consistently achieves a near-perfect accuracy of around 0.995, regardless of the number of components. This stability highlights MedPCA's robustness and ability to handle outliers effectively, ensuring high accuracy across different component numbers. Conversely, the accuracy of PCA + KNN shows significant improvement as the number of components increases, peaking at 0.995 with four components. However, beyond this point, the accuracy plateaus, indicating that PCA also performs well but may require careful tuning of the number of components to achieve optimal results.

The bottom panel of Fig. 2 reveals that MedPCA + KNN consistently requires less CPU time compared to PCA + KNN across all numbers of components. This suggests that MedPCA is more computationally efficient, making it a better choice for real-time applications where processing time is critical. While both PCA and MedPCA exhibit an increase in CPU time as the number of components rises, MedPCA maintains a consistently lower CPU time, underscoring its efficiency advantage.

For the IoT23 dataset (Fig. 3), several key observations can be made about the performance of MedPCA and standard PCA when paired with a K-Nearest Neighbors (KNN) classifier. In the top panel, MedPCA + KNN shows a more consistent and higher accuracy compared to PCA + KNN. While the PCA + KNN configuration fluctuates significantly, peaking at k=2 with an accuracy near 0.994 and then dropping noticeably after k=6, MedPCA + KNN maintains a more stable and slightly lower peak accuracy around 0.994 at k=2 but with fewer variations thereafter.

In terms of computational efficiency, the bottom panel of Fig. 3 highlights that MedPCA + KNN is consistently faster than PCA + KNN. The CPU time for MedPCA + KNN starts around 0.68 s and remains relatively stable below 0.72 s. On the other hand, PCA + KNN has a CPU time ranging from 0.74 to 0.76 s, which is consistently higher. This indicates that MedPCA is not only more accurate but also more efficient in processing the IoT23 dataset.

For the second dataset (Fig. 4), the trends observed are similar, further reinforcing the advantages of MedPCA. In the top panel, MedPCA + KNN shows superior accuracy stability across different k values. Unlike PCA + KNN, which exhibits significant drops at k=4 and k=8, MedPCA + KNN maintains an accuracy around 0.90 with minimal fluctuations. This consistent performance underscores MedPCA's robustness in handling datasets with potential outliers.

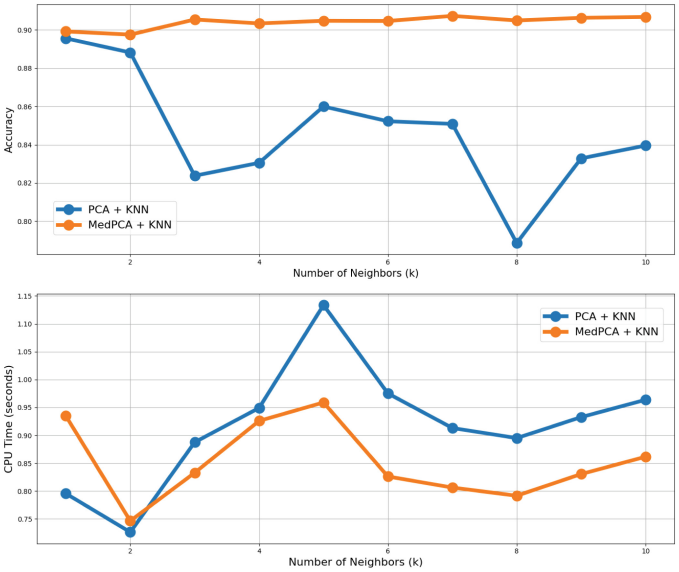

Fig. 3. Number of K vs Accuracy and CPU time for iot23

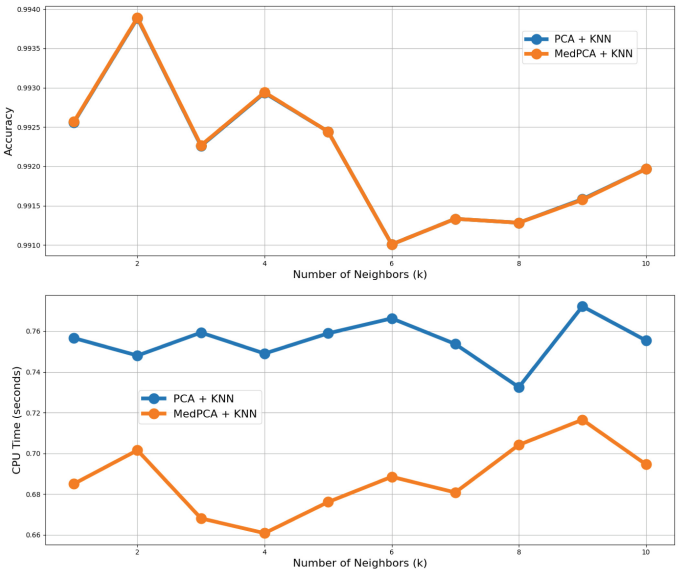

Fig. 4. Number of K vs Accuracy and CPU time CICIOT23 dataset

The bottom panel of Fig. 4 demonstrates that MedPCA also outperforms PCA in terms of computational efficiency for this dataset. The CPU time for MedPCA + KNN is consistently below 1 s, showing minor variations. In contrast, PCA + KNN displays more considerable fluctuations, peaking at k=6 with the highest CPU time observed. Overall, MedPCA maintains a lower and more stable CPU time compared to PCA.

6 Conclusion

In this article, we show that Median PCA (MedPCA) offers significant advantages over traditional PCA in the context of IDS for IoT networks. MedPCA not only provides higher and more consistent accuracy but also demonstrates superior computational efficiency. These attributes make MedPCA a viable and effective solution for real-time intrusion detection in large-scale, high-dimensional, and outlier-prone IoT networks.

Future work could extend this research by exploring the integration of MedPCA with other machine learning algorithms and testing on a broader range of IoT datasets. Additionally, real-world deployment and testing could provide further validation of MedPCA's practical applicability and performance in live environments.

References

1. Greenacre, M., Groenen, P.J., Hastie, T., d'Enza, A.I., Markos, A., Tuzhilina, E.: Principal component analysis. Nature Rev. Methods Primers **2**(1), 100 (2022)
2. Xanthopoulos, P., et al.: Linear discriminant analysis. Robust data mining, pp. 27–33 (2013)
3. Shen, Z., Zhang, Y., Chen, W.: A Bayesian classification intrusion detection method based on the fusion of PCA and LDA. Secur. Commun. Netw. **2019**(1), 6346708 (2019)
4. Zheng, D., Hong, Z., Wang, N., Chen, P.: An improved IDA based elm classification for intrusion detection algorithm in Iot application. Sensors **20**(6), 1706 (2020)
5. Vasan, K.K., Surendiran, B.: Dimensionality reduction using principal component analysis for network intrusion detection. Perspect. Sci. **8**, 510–512 (2016)
6. Elkhadir, Z., Chougdali, K., Benattou, M.: Intrusion detection system using PCA and kernel PCA methods. In: Proceedings of the editerranean Conference on Information & Communication Technologies 2015, pp. 489–497 (2016). Springer
7. Dash, S.K., et al.: Enhancing DDOS attack detection in IoT using PCA. Egyptian Inform. J. **25**, 100450 (2024)
8. Elkhadir, Z., Mohammed, B.: A cyber network attack detection based on gm median nearest neighbors LDA. Comput. Secur. **86**, 63–74 (2019)
9. Zyad, E., Khalid, C., Mohammed, B.: An effective network intrusion detection based on truncated mean LDA. In: 2017 International Conference on Electrical and Information Technologies (ICEIT), pp. 1–5 (2017).IEEE
10. Elkhadir, Z., Chougdali, K., Benattou, M.: An effective cyber attack detection system based on an improved OMPCA. In: 2017 International Conference on Wireless Networks and Mobile Communications (WINCOM), pp. 1–6 (2017). IEEE
11. https://www.stratosphereips.org/datasets-iot23
12. https://www.unb.ca/cic/datasets/iotdataset-2023.html