

Enhancing internet of things attack detection using principal component analysis and kernel principal component analysis with cosine distance and sigmoid kernel

Zyad Elkhadir, Mohammed Achkari Begdouri

SIGL Research Laboratory, National School of Applied Sciences of Tetouan (ENSATE), Abdelmalek Essaadi University, Tetouan, Morocco

Article Info

Article history:

Received May 3, 2024

Revised Aug 16, 2024

Accepted Sep 3, 2024

Keywords:

Cosine distance

Internet of things

Intrusion detection systems

Kernel principal component analysis

K-nearest neighbors

ABSTRACT

The widespread adoption of internet of things (IoT) devices has brought about unprecedented levels of connectivity and convenience. However, it has also introduced significant challenges, particularly in the areas of security and privacy. This study addresses the critical issue of intrusion detection within IoT environments, with a specific focus on analyzing the Iot-23 dataset. Our methodology involves employing principal component analysis (PCA) and kernel PCA for dimensionality reduction. Subsequently, we utilize the k-nearest neighbors (KNN) algorithm for classification purposes. To optimize the performance of the KNN algorithm, we experiment with various feature scaling techniques, such as StandardScaler, MinMaxScaler, and RobustScaler, utilizing different distance metrics. In our analysis, we discovered that employing the cosine distance metric in combination with KNN resulted in superior intrusion detection performance when utilizing PCA. Additionally, when utilizing kernel PCA, we evaluated multiple kernel functions and determined that the radial basis function and sigmoid kernel yielded the most favorable results.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Zyad Elkhadir

SIGL Research Laboratory, National School of Applied Sciences of Tetouan (ENSATE), Abdelmalek Essaadi University

Tetouan, Morocco

Email: z.elkhadir@uae.ac.ma

1. INTRODUCTION

Internet of things (IoT) is considered as an important connected devices environment where many intelligent solutions could be developed from smart cities to military applications [1]–[8] and many other sectors [9]. With the escalating number of connected devices worldwide, multiple sensors are employed to enable the remote collection of real-time data from physical objects. The latter serves as a foundation for constructing intelligent decision-making algorithms and efficiently managing IoT environments. However, the widespread utilization of real-world devices introduces heightened vulnerabilities to cybersecurity threats.

Malicious devices have the potential to surreptitiously monitor individuals, remotely alter traffic signals, and destabilize networks [10]. Noteworthy real-time attacks include distributed denial of service (DDoS) [11], the Mirai botnet [12] and denial of service (DoS) [13] orchestrated by botnet creators who may also provide mitigation services for a fee charged to the victim. The protection of IoT devices against such intrusions holds paramount importance in the realm of security. There is a growing imperative to undertake necessary measures to ensure both physical and cybersecurity against these potent attacks. Consequently, a thorough examination of networks using intrusion detection systems (IDS) has become indispensable.

Maintaining a high anomaly-based IDS accuracy is challenging due to the high dimensional network structures with many useless information. To augment its efficacy, researchers employ many feature extraction methods to obtain just the valuable data. The latter will help the IDS in classifying the network connections.

Many articles [14]–[17] used principal component analysis (PCA) variant with IDS in various non-IoT networks and showed good results. Recent studies have utilized as a feature extraction PCA with k-nearest neighbors (KNN) classifiers to improve IDS accuracy in IoT environments. For instance, the work [18] built an IDS using PCA and a genetic algorithm (GA) before employing a KNN classifier to enhance IDS accuracy. Dash *et al.* [19] proposed two approaches, one utilizing PCA and another without PCA, to compare their performance, demonstrating a noteworthy improvement in DDoS attack detection accuracy in IoT devices by integrating PCA and RobustScaler. Similarly, Abdaljabar *et al.* [20] combined K-NN and decision tree classifiers to achieve outstanding results in IDS performance metrics. However, these approaches primarily employed PCA/KNN with Euclidean distance and did not explore the efficiency of other distances such as Manhattan, Minkowski, Chebyshev, cosine similarity, hamming, Jaccard similarity, and correlation-based distances. Each of these metrics has its strengths and may be more suitable for specific data types or patterns within IoT networks.

Few other studies have employed the non-linear version of PCA, namely kernel PCA (KPCA), to improve IDS performance in IoT environments. For example, the study [21] combined KPCA for feature extraction and CNN for attack recognition and classification, demonstrating the effectiveness of this approach using bench-mark datasets. Similarly, Yang *et al.* [22] proposed a feature engineering model incorporating KPCA, and the work [23] utilized KPCA for feature extraction from biometric data in IoT-based smart buildings. However, most KPCA implementations predominantly utilize the radial basis function (RBF) kernel, without thoroughly exploring the potential benefits of other kernel functions such as polynomial and sigmoid kernels.

This paper focuses on several critical areas of study. First, we examine how various distance metrics impact the performance of the KNN classifier and PCA in IDS for IoT security. Next, we provide a comparative analysis of kernel PCA, utilizing different kernel functions, against traditional PCA. The rest of the paper is arranged in the following manner: Section 2 details the proposed IDS framework. Section 3 presents and discusses the experimental results. Finally, Section 4 summarizes the key findings.

2. METHOD

The proposed IDS, as depicted in Figure 1, consists of two main phases: the training phase and the testing phase. In the initial phase, data is collected from three devices, followed by data preprocessing procedure, which involves cleaning the data and replacing missing values. Subsequently, the processed data undergoes scaling methods, including standard, min-max, and RobustScalers, discussed extensively in Section 2.2 to ensure that all features are comparably scaled, preventing any individual feature from dominating solely based on its magnitude.

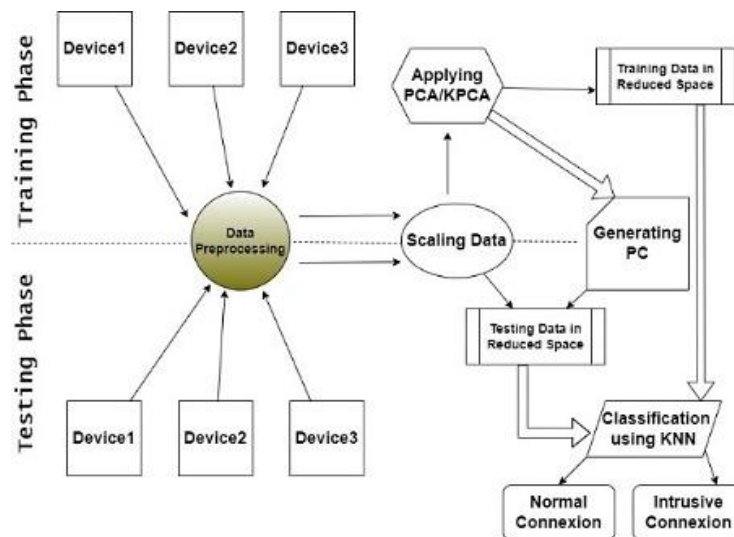


Figure 1. The proposed IDS model

Following scaling, principal component PCA or kernel KPCA is applied to the resulting data to extract the axes (PCs) of the reduced space and generate training data with reduced features, more details about PCA and KPCA are given in subsection 2.3. In the classification or testing phase, data is collected from the same devices, and identical preprocessing and scaling techniques are applied as in the training phase. The resulting data is then projected onto the reduced space using the PCs. Finally, the obtained testing data is compared to the training data in the same reduced space using the KNN algorithm to classify normal and malicious data. Subsection 2.4. gives an overview about KNN and the employed distances.

2.1. Dataset

The dataset utilized in this study was sourced from [24], known as the IoT-23 dataset, it contains real and labeled instances of IoT malware infections alongside normal traffic. The 23 captures are constructing the dataset and organized as 3 normal captures and 20 malicious captures. Every capture from corrupted devices is labeled with a corresponding malware sample executed. Due to the vast size of the dataset, a decision was made to retain a small part of connections from each individual dataset and subsequently merge them into a novel dataset. This manipulation was undertaken to ensure that our computational resources could efficiently manage the workload of the novel dataset and simultaneously extracting the most of attack types present within the initial IoT-23 dataset.

Initially, the Python library Pandas was employed to individually load all 23 datasets from the IoT-23 dataset into data frames. A condition was applied to skip the first 10 rows and read the subsequent one hundred thousand rows. Subsequently, these 23 data frames were consolidated into a single data frame. Missing values were replaced with 0. Finally, the combined dataset was generated and saved as the "iot23combined.csv" file. The resulting "iot23combined.csv" file comprises a total of 1,444,674 records. It encompasses 10 types of attacks.

2.2. Scaling techniques

In the domain of data preprocessing for machine learning, scaling techniques are instrumental in enhancing the robustness and efficacy of predictive models. Three commonly utilized scalers are StandardScaler, MinMaxScaler, and RobustScaler. StandardScaler: this scaler standardizes the features by subtracting the mean (μ) and dividing by the standard deviation (σ).

$$\text{Standardized Feature} = \frac{\text{Feature} - \mu}{\sigma}$$

It is particularly effective when features exhibit different scales, ensuring that each feature contributes equally to the model. The transformed data has a mean of zero and a standard deviation of one. MinMaxScaler: MinMaxScaler scales the features to a specified range, typically between 0 and 1, using the following formula:

$$\text{Scaled Feature} = \frac{\text{Feature} - \min(\text{Feature})}{\max(\text{Feature}) - \min(\text{Feature})}$$

It is useful when the data distribution is not Gaussian, helping mitigate the impact of outliers while maintaining the shape of the original distribution. RobustScaler: it handles outliers by scaling features based on robust statistics. It utilizes the median ($\tilde{\mu}$) and interquartile range (IQR).

$$\text{Robustly Scaled Feature} = \frac{\text{Feature} - \tilde{\mu}}{\text{IQR}}$$

This scaler is less sensitive to extreme values, making it suitable for datasets with outliers or skewed distributions. In summary, the selection of a scaler depends on the data characteristics and the specific requirements of the machine learning task. Each scaler offers unique advantages in terms of handling data distributions, outliers, and maintaining the integrity of the underlying information.

2.3. Feature extraction methods

2.3.1. Principal component analysis

The goal of PCA is to decrease dimensionality while preserving as much variance as possible. This is achieved by focusing on the leading principal components (PCs), which are arranged in descending order of their variance [25]. Mathematically, given a training set of M vectors w_i each with n features. We obtain n' ($n' \ll n$) principal components from the training set through the following steps:

- Calculate the arithmetic mean σ of this set:

$$\sigma = \frac{1}{M} \sum_{i=1}^M (w_i) \quad (1)$$

- Remove the mean σ from w_i to obtain ρ_i :

$$\rho_i = w_i - \sigma \quad (2)$$

- Construct the covariance matrix C , where:

$$C_{n \times n} = \frac{1}{M} \sum_{i=1}^M (\rho_i \rho_i^T) = AA^T \quad (3)$$

$$\text{and } A_{n \times M} = \frac{1}{\sqrt{M}} \rho_i \quad (4)$$

Let U_k be the k -th eigenvector of C corresponding to the λ_k associated eigenvalue. Form a matrix $Un \times n' = [U_1 \dots Un']$ consisting of these eigenvectors, such that:

$$CU_k = \lambda_k U_k \quad (5)$$

The first U_k corresponding to the largest eigenvalues λ_k are termed principal components (PCs).

2.3.2. Kernel principal component analysis

Kernel principal component analysis (kernel PCA) [26] is an extension of PCA that uses kernel methods to perform nonlinear dimensionality reduction. The basic idea behind kernel PCA is to implicitly map the input data into a higher-dimensional feature space, where it becomes linearly separable, and then perform PCA in that space. This allows kernel PCA to capture nonlinear relationships between the original features. The kernel trick is used to compute the dot product in the higher-dimensional feature space efficiently without explicitly computing the transformation. Given a kernel function $K(x_i, x_j)$, where x_i and x_j are input data points, the kernel PCA algorithm can be summarized as (6)-(11):

- Compute the kernel matrix K :

$$K_{ij} = K(x_i, x_j) \quad (6)$$

- Center the kernel matrix:

$$K' = K - 1K - K1 + 1K1 \quad (7)$$

where 1 denotes a matrix of ones.

- Compute the eigenvectors α_i and eigenvalues λ_i of the centered kernel matrix K' :

$$K' \alpha_i = \lambda_i \alpha_i \quad (8)$$

Select the first n' eigenvectors corresponding to the largest eigenvalues to form the principal components the RBF kernel is defined as (9):

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) \quad (9)$$

where σ is the kernel bandwidth parameter. The Polynomial kernel is defined as (10):

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + c)^d \quad (10)$$

where d is the degree of the polynomial and c is a constant term. The sigmoid kernel is defined as (11):

$$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\alpha \mathbf{x}_i^T \mathbf{x}_j + \beta) \quad (11)$$

where α and β are scaling parameters.

2.4. k-nearest neighbors

In order to classify a data point, KNN looks at the majority class of its k-nearest neighbors in the feature space. Given a dataset $(x_1, y_1), (x_2, y_2) \dots (x_n, y_n)$ where x_i represents the feature vector and y_i is the corresponding class label, the KNN algorithm operates as: i) Choose the number of neighbors k ; ii) Compute the distance between the target point and each data point in the training set; iii) Determine the k closest neighbors based on the computed distances; and iv) For classification, assign the class label by majority vote among the k neighbors. For regression, predict the arithmetic mean of the k neighbors target values. Various distance metrics and similarity measures are used to quantify the dissimilarity or similarity between data points.

- Manhattan distance: The Manhattan distance, also known as Taxicab or L_1 norm, between two points p and q in an n-dimensional space is given by (12).

$$d_{\text{Manhattan}}(p, q) = \sum_{i=1}^n |p_i - q_i| \quad (12)$$

- Minkowski distance: The Minkowski distance of order p between two points p and q is a generalization of both the Euclidean and Manhattan distances.

$$d_{\text{Minkowski}}(p, q) = (\sum_{i=1}^n |p_i - q_i|^p)^{\frac{1}{p}} \quad (13)$$

- Chebyshev distance: The Chebyshev distance, also known as L-infinity norm, between two points p and q is the maximum absolute difference along any dimension.

$$d_{\text{Chebyshev}}(p, q) = \max_i |p_i - q_i| \quad (14)$$

- Cosine similarity: Cosine similarity measures the cosine of the angle between two non-zero vectors p and q .

$$\text{Cosine}_{\text{similarity}}(p, q) = \frac{\sum_{i=1}^n (p_i \cdot q_i)}{\sqrt{\sum_{i=1}^n (p_i^2)} \cdot \sqrt{\sum_{i=1}^n (q_i^2)}} \quad (15)$$

- Hamming distance: The Hamming distance between two strings of equal length is defined as the count of positions where the corresponding symbols are different.

$$d_{\text{Hamming}}(\text{string}_1, \text{string}_2) = \sum_{i=1}^n (\text{string}_1[i] \neq \text{string}_2[i]) \quad (16)$$

- Jaccard similarity: Jaccard similarity measures the ratio of the size of the intersection to the size of the union of two sets.

$$\text{Jaccard}_{\text{similarity}}(\text{set}_1, \text{set}_2) = \frac{|\text{set}_1 \cap \text{set}_2|}{|\text{set}_1 \cup \text{set}_2|} \quad (17)$$

- Correlation-based distances: Correlation-based distances consider the correlation coefficient between two vectors. The Pearson correlation coefficient is commonly used (18).

$$\text{Correlation}(p, q) = \frac{\sum_{i=1}^n ((p_i - \bar{p})(q_i - \bar{q}))}{\sqrt{\sum_{i=1}^n ((p_i - \bar{p})^2)} \cdot \sqrt{\sum_{i=1}^n ((q_i - \bar{q})^2)}} \quad (18)$$

3. RESULTS AND DISCUSSION

The accuracy of an IDS is a critical metric that measures its effectiveness in correctly identifying and classifying instances of intrusion or abnormal behavior within a network. The accuracy is generally calculated as the ratio of correctly identified instances (true positives and true negatives) to the total number of instances. The formula for accuracy is as follows:

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

where:

- True positives (TP): instances correctly identified as intrusions.
- True negatives (TN): instances correctly identified as normal.
- Total instances: $TP + TN + \text{false positives} + \text{false negatives}$.

In the conducted experiments, we randomly selected 10,000 normal data and 1,000 malicious data for the training phase. For the testing phase, 5,000 normal data and 1,000 malicious data were selected. This selection process was repeated 20 times.

This setting ensures the reliability and generalizability of findings by mitigating biases introduced by a single random split and assessing variability across iterations. By mimicking real-world scenarios and assessing system performance across diverse data samples, it enables insights into generalization capabilities and stability. Furthermore, statistical significance can be established, facilitating efficient resource utilization despite multiple repetitions.

The experiments are divided in two parts, the first one concerns PCA/KNN and investigate the efficiency of other distances such as Manhattan distance, Minkowski distance, Chebyshev distance, cosine similarity, Hamming distance, Jaccard similarity, and correlation-based distances. We explored three different scaling techniques: StandardScaler, MinMaxScaler, and RobustScaler. When assessing the KNN distances using StandardScaler, we maintained a consistent number of principal components at “3” as shown in Figure 2, “5” as shown in Figure 3, and “7” as shown in Figure 4, while systematically adjusting the value of K . In another experiment as shown in Figure 5, we kept K fixed at 1 and varied the number of principal components. Across both experiments involving MinMaxScaler and RobustScaler, we set K to 1 while modifying the number of principal components. The results illustrated by Figure 6 and Figure 7 consistently indicated that cosine distance exhibited slightly better performance compared to other distance metrics. However, it is worth noting that Jaccard distance did not yield favorable outcomes in any of the investigated scenarios.

The robustness of cosine distance to noise and outliers contributes significantly to the obtained results. Unlike distance metrics that consider both the direction and magnitude of vectors, cosine distance focuses solely on directional similarity. By measuring the cosine of the angle between two vectors, it emphasizes the similarity in their directions rather than their magnitudes. This property enables cosine distance to effectively capture essential directional patterns in the data, even in the presence of noise or outliers. When vectors exhibit similar directions, cosine similarity remains high, indicating their similarity regardless of potential differences.

In the second part of our experiments, we aimed to further enhance the accuracy of our IDS by employing kernel PCA while exploring various kernel functions. Specifically, we investigated the RBF, polynomial, and sigmoid functions, comparing their performance against PCA with both Euclidean and cosine distance metrics. The considered values of kernel parameters are given by: $\sigma = 1/(\text{number of principal components})$ for radial basis function (RBF). $d = 3$ is the degree and $c = 1$ for polynomial kernel. α and β are scaling parameters and given by $\alpha = 1/(\text{number of principal components})$ and $\beta = 1$ for sigmoid kernel.

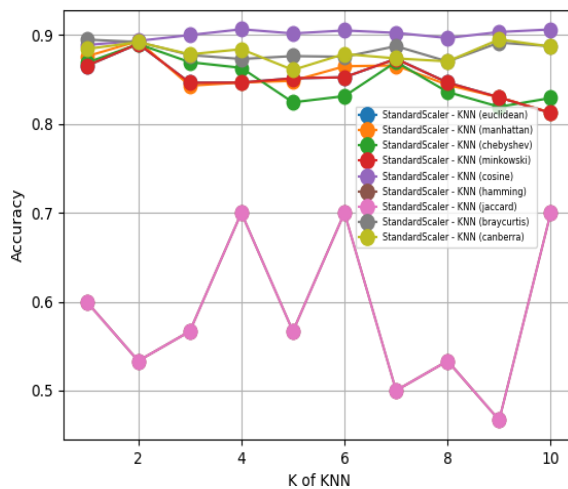


Figure 2. K vs Accuracy (%) for 3 PC using StandardScaler

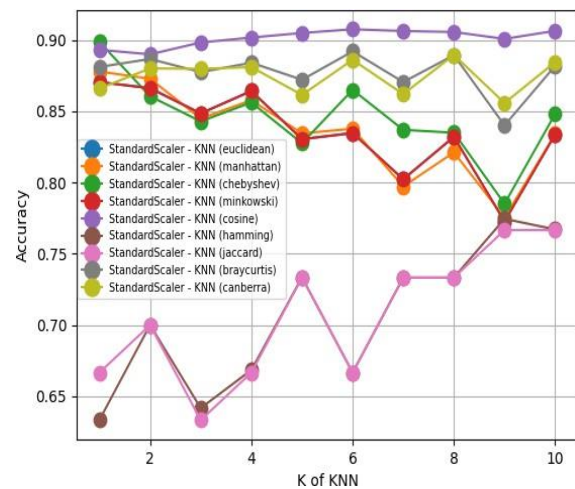


Figure 3. K vs Accuracy (%) for 5 PC using StandardScaler

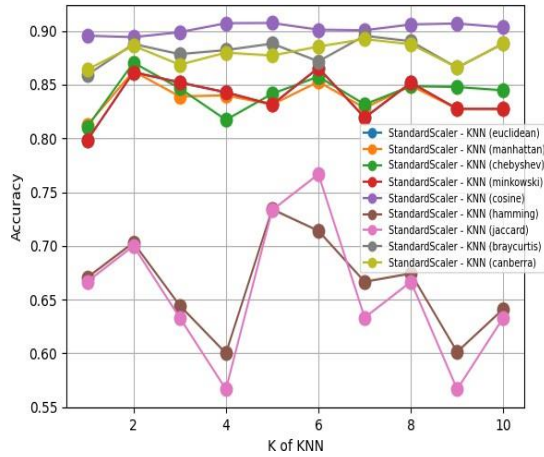


Figure 4. K vs Accuracy (%) for 7 PC using StandardScaler

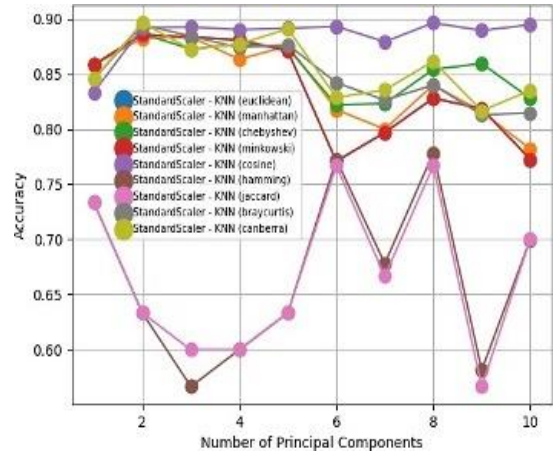


Figure 5. PC vs Accuracy (%) for K=1 using StandardScaler

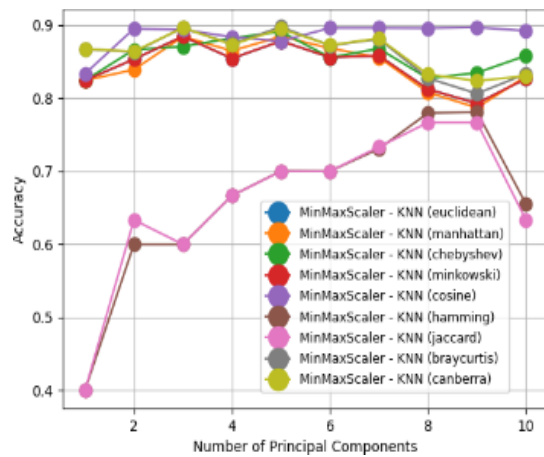


Figure 6. PC vs Accuracy (%) for K=1 using MinMaxScaler

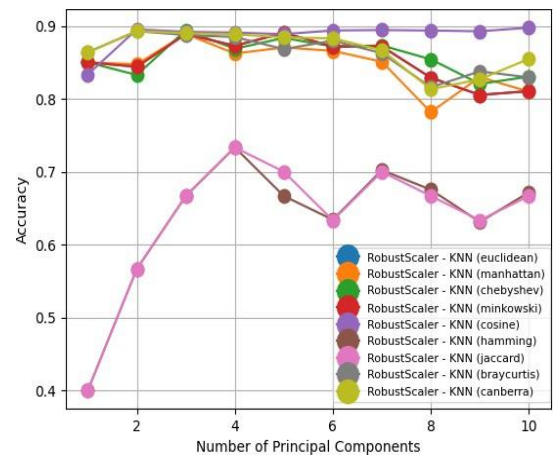


Figure 7. PC vs Accuracy (%) for K=1 using RobustScaler

Figures 8 and 9 illustrate experiments where the number of principal components (PCs) was set to 3 and 5 respectively, while the value of K in the K-nearest neighbors (KNN) algorithm varied from 1 to 10. The accuracy of the intrusion detection system (IDS) using Euclidean distance was visualized. Interestingly, the results indicate that KPCA outperformed traditional PCA under these conditions.

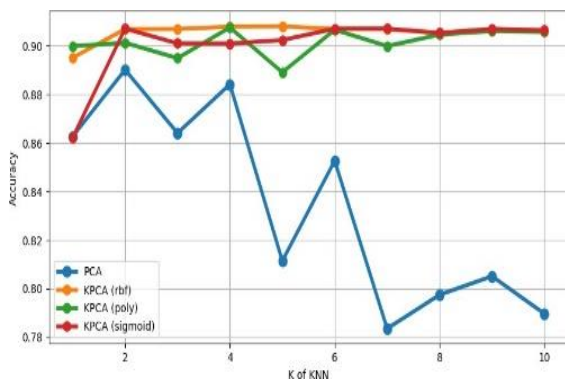


Figure 8. K vs Accuracy (%) for 3 PC using Euclidean distance

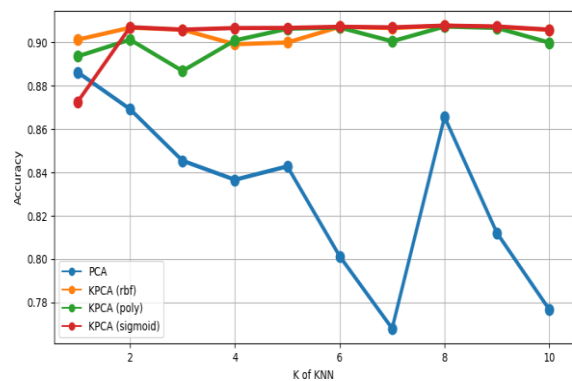


Figure 9. K vs Accuracy (%) for 5 PC using Euclidean distance

In Figures 10 and 11, similar experiments were conducted, but with cosine distance utilized instead. Here, we observed that the sigmoid function surpassed PCA and other kernel functions in terms of accuracy. One possible reason why the sigmoid function surpassed PCA and other kernel functions in terms of accuracy could be its ability to effectively capture complex, non-linear relationships present in the data. Unlike traditional PCA, which assumes linear relationships between variables, the sigmoid kernel introduces non-linearity into the feature space, allowing it to capture more intricate patterns and structures. While RBF and polynomial kernels are also effective in capturing non-linear relationships, the superiority of the sigmoid function in these experiments may be attributed to its ability to better handle non-linear data structures, adaptability in shaping decision boundaries, robustness to noise and outliers. In the last experiment illustrated by Figure 12, we modify the number of principal components and fix K at 1, we observe that RBF and sigmoid kernels give the best results.

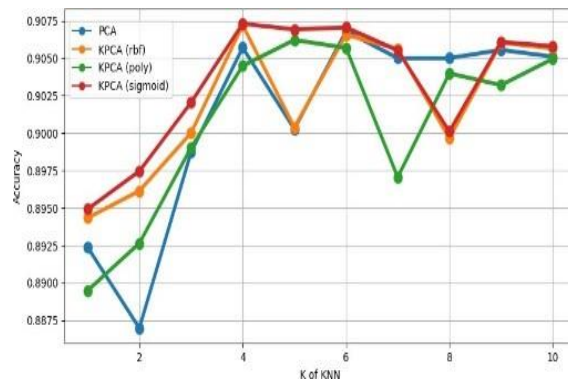


Figure 10. K vs Accuracy (%) for 3 PC using cosine distance

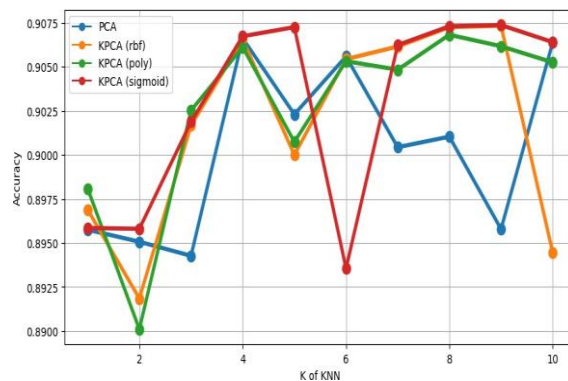


Figure 11. K vs Accuracy (%) for 5 PC using cosine distance

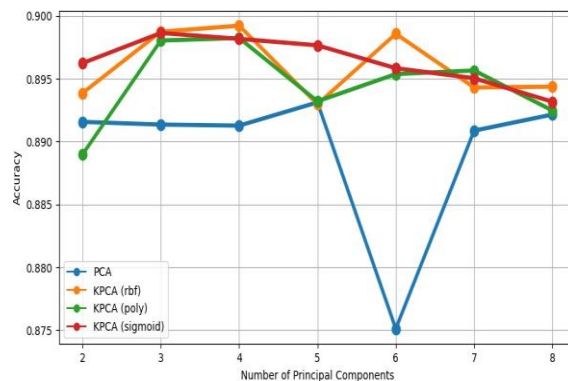


Figure 12. PC vs. Accuracy (%) for K=1 using cosine distance




4. CONCLUSION

This study evaluated the effectiveness of a novel deep learning-based IDS designed to identify IoT attacks by leveraging PCA and KPCA in conjunction with the KNN algorithm. We investigated various KNN distance metrics and found that the cosine distance consistently provided superior intrusion detection performance, especially in high-dimensional network data. Additionally, KPCA with sigmoid kernels outperformed traditional PCA, capturing complex non-linear relationships within the data. These findings underscore the potential for using cosine distance and advanced kernel functions to significantly enhance the accuracy and robustness of IDS. The implications of this study are far-reaching, suggesting pathways for developing more effective security solutions for IoT networks and other critical systems. Future research should focus on the scalability of these methods in large-scale networks and their application in real-time detection systems.




REFERENCES

- [1] Z. Chen, C. B. Sivaparthipan, and B. A. Muthu, "IoT based smart and intelligent smart city energy optimization," *Sustainable Energy Technologies and Assessments*, vol. 49, 2022, doi: 10.1016/j.seta.2021.101724.
- [2] A. N. Soni, "Smart devices using internet of things for health monitoring," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 7, no. 5, pp. 6355–6361, 2018, doi: 10.15680/IJIRSET.2018.0705233.
- [3] C. Stoloiescu-Crisan, C. Crisan, and B. P. Butunoi, "An IoT-based smart home automation system," *Sensors*, vol. 21, no. 11, 2021, doi: 10.3390/s21113784.
- [4] E. Fantin Irudaya Raj and M. Appadurai, "Internet of things-based smart transportation system for smart cities," *Springer*, 2022, pp. 39–50, doi: 10.1007/978-981-19-0770-8_4.
- [5] T. M. Ghazal *et al.*, "IoT for smart cities: machine learning approaches in smart healthcare—A review," *Future Internet*, vol. 13, no. 8, p. 218, 2021, doi: 10.3390/fi13080218.
- [6] R. Akhter and S. A. Sofi, "Precision agriculture using IoT data analytics and machine learning," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, pp. 5602–5618, 2022, doi: 10.1016/j.jksuci.2021.05.013.
- [7] M. A. Alomar, "An IoT based smart grid system for advanced cooperative transmission and communication," *Physical Communication*, vol. 58, no. 8, p. 102069, 2023, doi: 10.1016/j.phycom.2023.102069.
- [8] V. Gotarane and S. Raskar, "IoT practices in military applications," *Proceedings of the International Conference on Trends in Electronics and Informatics, ICOEI 2019*, vol. 2019-April, pp. 891–894, 2019, doi: 10.1109/icoei.2019.8862559.
- [9] A. Khanna and S. Kaur, "Internet of things (IoT), applications and challenges: a comprehensive review," *Wireless Personal Communications*, vol. 114, no. 2, pp. 1687–1762, 2020, doi: 10.1007/s11277-020-07446-4.
- [10] S. Al-Sarawi, M. Anbar, R. Abdullah, and A. B. Al Hawari, "Internet of things market analysis forecasts, 2020-2030," *Proceedings of the World Conference on Smart Trends in Systems, Security and Sustainability, WS4 2020*, pp. 449–453, 2020, doi: 10.1109/WorldS450073.2020.9210375.
- [11] A. Bahaa, A. Abdelaziz, A. Sayed, L. Elfangary, and H. Fahmy, "Monitoring real time security attacks for IoT systems using devsecops: A systematic literature review," *Information (Switzerland)*, vol. 12, no. 4, 2021, doi: 10.3390/info12040154.
- [12] A. Marzano *et al.*, "The evolution of Bashlite and Mirai IoT botnets," *Proceedings - IEEE Symposium on Computers and Communications*, vol. 2018-June, pp. 813–818, 2018, doi: 10.1109/ISCC.2018.8538636.
- [13] V. Tomer and S. Sharma, "Detecting IoT attacks using an ensemble machine learning model," *Future Internet*, vol. 14, no. 4, 2022, doi: 10.3390/fi14040102.
- [14] Z. Elkhadir, K. Chougali, and M. Benattou, "Intrusion detection system using PCA and kernel PCA methods," *Lecture Notes in Electrical Engineering*, vol. 381, pp. 489–497, 2016, doi: 10.1007/978-3-319-30298-0_50.
- [15] E. Ziad, A. Taha, and B. Mohammed, "Improve R2L attack detection using trimmed PCA," *Proceedings - 2019 International Conference on Advanced Communication Technologies and Networking, CommNet 2019*, 2019, doi: 10.1109/COMMNET.2019.8742361.
- [16] C. Khalid, E. Ziad, and B. Mohammed, "Network intrusion detection system using L1-norm PCA," *Proceedings of the 2015 11th International Conference on Information Assurance and Security, IAS 2015*, pp. 118–122, 2016, doi: 10.1109/ISIAS.2015.7492755.
- [17] E. Ziad, C. Khalid, and B. Mohammed, "An effective cyber attack detection system based on an improved OMPCA," *Proceedings - 2017 International Conference on Wireless Networks and Mobile Communications, WINCOM 2017*, 2017, doi: 10.1109/WINCOM.2017.8238162.
- [18] M. Mohy-eddine, A. Guezzaz, S. Benkirane, and M. Azrou, "An efficient network intrusion detection model for IoT security using K-NN classifier and feature selection," *Multimedia Tools and Applications*, vol. 82, no. 15, pp. 23615–23633, 2023, doi: 10.1007/s11042-023-14795-2.
- [19] S. K. Dash *et al.*, "Enhancing DDoS attack detection in IoT using PCA," *Egyptian Informatics Journal*, vol. 25, 2024, doi: 10.1016/j.eij.2024.100450.
- [20] Z. H. Abdaljabar, O. N. Ucan, and K. M. Ali Altheeti, "An intrusion detection system for IoT using KNN and decision-tree based classification," *International Conference of Modern Trends in ICT Industry: Towards the Excellence in the ICT Industries, MTICTI 2021*, 2021, doi: 10.1109/MTICTI53925.2021.9664772.
- [21] T. Gaber, J. B. Awotunde, M. Torkey, S. A. Ajagbe, M. Hammoudeh, and W. Li, "Metaverse-IDS: Deep learning-based intrusion detection system for Metaverse-IoT networks," *Internet of Things (Netherlands)*, vol. 24, 2023, doi: 10.1016/j.iot.2023.100977.
- [22] L. Yang, A. Moubayed, and A. Shami, "MTH-IDS: A multitiered hybrid intrusion detection system for internet of vehicles," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 616–632, 2022, doi: 10.1109/JIOT.2021.3084796.
- [23] C. Annadurai *et al.*, "Biometric authentication-based intrusion detection using artificial intelligence internet of things in smart city," *Energies*, vol. 15, no. 19, p. 7430, Oct. 2022, doi: 10.3390/en15197430.
- [24] S. Garcia, A. Parmisano, and M. J. Erquiaga, "IoT-23: a labeled dataset with malicious and benign IoT network traffic," *Zenodo*, vol. Version 1, 2020, doi: 10.5281/zenodo.4743746.
- [25] I. Jolliffe, "Principal component analysis," in *Encyclopedia of Statistics in Behavioral Science*, Wiley, 2005, doi: 10.1002/0470013192.bsa501.
- [26] B. Schölkopf, A. Smola, and K. R. Müller, "Kernel principal component analysis," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 1327, pp. 583–588, 1997, doi: 10.1007/bfb0020217.

BIOGRAPHIES OF AUTHORS

Zyad Elkhadir    is a computer science researcher and professor, graduated from Faculty of science of Ibn Tofail University, Kenitra, Morocco. He obtained his master degree in software quality in 2013 and Ph.D. degree in 2018 from the same faculty. He is also an IEEE member. His main research interest is to develop new feature extraction algorithms for pattern recognition problem such as network intrusion detection. He was a contractual professor of computer science at the national school of applied sciences of Kenitra (ENSA-K). He is currently a professor of computer science at Polydisciplinary Faculty of Larache (Morocco). He can be contacted at email: z.elkhadir@uae.ac.ma.



Mohammed Achkari Begdouri    obtained the state engineer diploma in computer science from the National School of Applied Sciences of Tangier, Morocco, in 2011. The same year, he joined the Interior Ministry as head of the information systems department in Tangier. He received the Ph.D. degree in 2017 from Abdelmalek Essaadi University, Morocco. He is currently professor at the Polydisciplinary Faculty of Larache (Morocco). He teaches and supervises projects on software engineering and systems administration. His research area focuses on theoretical computer science and software applications. He published various works in international journals. He participated to national and international conferences. He was member of various program committees, organizing committees and scientific projects. He can be contacted at email: m.achkaribegdouri@uae.ac.ma.