Speeding Up ML-based IDSs through Data Preprocessing Techniques

Lawrence Owusu
Department of Computational Data Science and Engineering
North Carolina Agricultural and Technical State University
1601 E. Market Street, Greensboro, NC, 27411

Ahmad Patooghy
Department of Computer Systems and Technology
North Carolina Agricultural and Technical State University
1601 E. Market Street, Greensboro, NC, 27411

Masud R. Rashel
University of Evora/Institute of Earth Sciences, Evora, 7002-554, Portugal

Marwan Bikdash
Department of Computational Data Science and Engineering
North Carolina Agricultural and Technical State University
1601 E. Market Street, Greensboro, NC, 27411

Islam AKM Kamrul
Department of Computational Data Science and Engineering
North Carolina Agricultural and Technical State University
1601 E. Market Street, Greensboro, NC, 27411

ABSTRACT

Most of the current ML-based IDSs models priotize detection accuracy over detection latency, which is critical for real-time detection and mitigation of cyber-attacks. The study evaluated the impact of Principal Component Analysis (PCA) on optimizing machine learning-based IDS using the UNR-IDD dataset. We comprehensively analyzed the performance of Support Vector Machine (SVM), Naïve Bayes (NB), and Random Forest (RF) before and after PCA transformation. Experimental results show that PCA significantly reduced the detection latency for SVM and NB without compromising their performance. Specifically, NB + PCA and SVM + PCA achieved a whopping 99.52% and 49.9% reduction in detection latency respectively, making them viable low-latency solutions. However, the PCA transformation did not significantly impact the detection latency of the random forest model. The results demonstrate that NB + PCA is the most efficient and lightweight model for real-time network intrusion detection. These findings demonstrate that PCA is an effective preprocessing step to optimize ML-based IDS for real-time applications.

General Terms

Machine learning, supervised learning, network attacks

Keywords

Intrusion detection, principal component analysis, latency, data and network security

1. INTRODUCTION

The rapid advancement of digital technologies has led to exponential expansion of cyberspace [30]. While this digital transformation offers many benefits, it has also introduced significant cybersecurity challenges and vulnerabilities [26, 30, 45]. Cyber-criminals continuously exploit vulnerabilities through sophisticated attack vectors such as distributed denial of service [3], Man-in-the-middle attack [4, 49], phishing [10], malware [5], and password attacks[33, 45], social engineering [18] and zero-day exploit [19].

A network intrusion refers to an unauthorized access of a computer system, either by external attackers or malicious insiders. With the rapid growth of cybercrime and its potentially devastating impacts, robust security measures are crucial for preventing financial losses and organizational damage [30, 50]. Over time, researchers have developed antivirus software, firewalls, and intrusion detection systems (IDS) to safeguard networks. Among these tools, intrusion detection system has provided cutting-edge security by continuously scanning and monitoring incoming traffic to detect malicious activities, triggering alerts, and prompting system administrators to take immediate action[2, 30, 46].

Machine learning-based IDS have gained prominence due to their ability to process vast amounts of network traffic data and detect previously unseen attack patterns. Despite their success, detection latency remains a significant challenge. Many existing IDS models prioritize achieving high classification accuracy over the computational costs associated with real-time detection. However, focusing solely on accuracy is detrimental in security-critical environments, where real-time identification and response are essential to mitigating cyberattacks [41, 28].

The high dimensionality of network traffic data exacerbates this challenge, as machine learning-based IDS must process vast amount of features [35]. Handling such high-dimensional data increases processing time, leading to significant detection latency. Therefore, optimizing IDS for both accuracy and computational efficiency is crucial to enhancing network security resilience.

Principal Component Analysis (PCA) has been proposed for enhancing the performance of IDS. However, its effectiveness on optimizing IDS is still underexplored, especially when it comes to balancing detection accuracy and latency [27, 48, 51]. In this study, we explore the effects of the Principal Component Analysis on ML-based IDSs using the UNR-IDD dataset as a benchmark. We will evaluate the performance of Naïve Bayes (NB), Support Vector Machine (SVM), and Random Forest (RF) before and after PCA application to assess whether dimensionality reduction can reduce detection latency without degrading performance. The research provides insights into the trade-offs between detection latency and performance in IDS, offering practical guidance for optimizing machine learning-based network intrusion detection solutions in resource-constrained environments.

The remainder of the paper is organized as follows: the related work section discusses previous works in intrusion detection and the role of PCA in IDS. The methodology chapter explains the experimental setup such as dataset pre-processing, feature feature engineering, feature and model selection, and model training. Results and discussion section presents the findings, and assesses the impact of PCA in improving IDS performance.The conclusion highlights the main contributions and proposes future research.

2. RELATED WORK

Research on intrusion detection dates back as far as 1972 when James Anderson had written a report for the U.S. Air Force, urging the need of intrusion detection systems [6]. At first, manual inspection of logs by system administrators was the main method of identifying potential cyber threats. But as technology advanced, human expertise alone was unable to match up with the increasing complexity of security needs. This inspired the use of automated intrusion detection systems(IDS). Denning and Neumann later created the first real-time IDS based upon expert-written rules in 1985[7, 14].

Since then, many anomaly-based and signature-based IDSs have been proposed in the public domain to enhance computer security. Although signature-based technique is fantastic at detecting known types of attacks, it is unable to detect a zero-day exploit. On the other hand, aan anomaly-based IDS determines which behavior within a network is normal, based on a predefined baseline. The anomaly-based IDS continues to monitor network activity, compares it to the baseline, and flag any significant deviation from the baseline as an attack. However, this technique tends to have higher false positives[1, 15, 26].
.

With the advancement in technology, traditional signature and anomaly-based methods are not able to detect cyber-threats due to the evolving nature of the attack vectors. Bhattacharya et al. proposed a Novel PCA-Firefly Based XGBoost machine learning model to enhance the efficiency of Intrusion Detection Systems (IDS). The proposed hybrid approach combines PCA for dimensionality reduction and the Firefly optimization algorithm for feature selection. After the application of the PCA and the firefly optimization algorithms, the authors trained K-Nearest Neighbor (KNN), Naïve Bayes (NB), Random Forest, Support Vector Machine (SVM), and XGBoost on the optimized dataset. Thorough comparative analysis showed that the PCA-Firefly-XGBoost model outperformed the traditional machine learning models, achieving an accuracy (99.9%) [9, 43].

Sayeed et al. investigated the security risks of the rapid expansion of IoT devices. Recognizing the limitations of traditional security methods, they proposed machine learning-based IDS tailored for IoT environments. Using PCA for dimensionality reduction, they optimized the UNSW-NB15 dataset to streamline computational loads and trained XGBoost, CatBoost, KNN, SVM, QDA, and Naive Bayes on the optimized dataset. The experimental results indicated that the XGBoost and CatBoost models achieved near-perfect accuracy (99.99%) and impressive F1 score and Matthew's Correlation Coefficient [24].

Kumari et al 2024 suggested a highly accurate IDS based on Spider Monkey Optimization algorithm. The spider monkey optimization (SMO) algorithm simulatess the foraging behavior of spider monkeys to obtain better exploration and exploitation of the solution space for optimizing the neural network parameters. The authors reported that SMO + ANN model achieved 100% accuracy on the LuFlow dataset and 99% on the NSL-KDD dataset, showcasing the potential of SMO-ANN algorithms in enhancing detection systemsfor intricate cyber threats[41, 26]

Because of the fast growing number of IoT devices in industries such as healthcare, smart cities and industrial applications, Roy and Cheung (2018) proposed a deep learning IDS using Bi-Directional LSTM to tackle the severe security vulnerabilities and defend IoT networks from cyber-attacks. The results indicates that the proposed BiLSTM-based IDS achieved an accuracy over 95% in detecting attacks[36].

Das and his team introduced the UNR-IDD dataset in 2022 to improve the accuracy and reliability of ML-based Network Intrusion Detection Systems. Previous benchmark datasets such as the NSL-KDD, UNSW-NB15 and CIC-IDS-2018 tend to have poor representation for rare attack types, restricting their generalizability over various kinds of network topologies. To solve these problems, the authors incorporated network port statistics and port statistics into the UNR-IDD, granting more fine-grained analysis of network traffic. The attack vector of the UNR-IDD dataset includes TCP-SYN flood, Port Scan, Flow Table Overflow, Blackhole and Diversion. The authors applied different machine learning algorithms on the dataset. The results indicated that the Random Forest and Bagging Classifier performed the best, achieving an F-Measure of 94% [13]
.

Parveen et al. explored the application of deep learning to address the shortcomings of tradtional IDS techniques. Upon preprocessing the UNR- IDD dataset, they trained Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Artificial Neural Networks (ANNs), and a hybrid CNN-RNN model to capture both spatial and temporal features. The authors reported that the hybrid CNN-RNN model performed the best among all models with 96.2% accuracy and a false positive rate of 1.5% [34].

Surjeet et al. proposed the Optimized LightGBM Model for protecting cyber-physical systems (CPS) against zero-day vulnerabilities. The authors trained the LightGBM model on the UNR-IDD, a benchmark dataset with a variety of attack vectors.

After preprocessing and hyperparameter tuning through Bayesian optimization, the optimized LightGBM model outperformed other algorithms, achieving an accuracy of 99.17%. [12]

Samriya et al. explored the optimization of network intrusion detection in cloud computing environments to address low accuracy and higher false alarm rate associated with detecting complex cyber-attack like DDoS and ransomware. They proposed a hybrid model, comprising of Support Vector Machine (SVM) and Extreme Gradient Boosting (XGBoost), that was further improved with Crow Search Algorithm (CSA). On the NSL-KDD and UNR-IDD datasets, the CSA-SVM model outperformed others, achieving 99.58% accuracy on NSL-KDD and 99.79% on UNR-IDD [38].

Despite the tremendous success achieved in building ML-based IDSs using the UNR-IDD dataset, comprehensive review of the existing literature reveals a significant gap: existing ML-based IDS implementations have been prioritizing detection performance over detection latency. Given the real-time nature of cyber-attacks, it is important to optimize ML-based IDS for low-latency detection to ensure swift reactions to cyberattacks. This study addresses the limitation with the following contributions.

1. Analyzes the impact of PCA on detection latency, a critical yet underexplored aspect in machine learning-based IDS research.

2. Evaluates the trade-off between latency and accuracy to demonstrate that PCA improves computational efficiency without compromising performance.

3. Proposes a lightweight, efficient and low-latency machine learning-based IDS systems capable of real-time threat detection in modern network environments.

3. METHOD

3.1 Description and Collection of the UNR-IDD Dataset

The University of Nevada-Reno Intrusion Detection Dataset, UNR-IDD dataset (https://www.tapadhirdas.com/unr-idd-dataset) can be used for both binary and multi-class classification tasks. The attack vector includes TCP-SYN Flood, Port Scan, Flow Table Overflow, Blackhole, and Traffic Diversion. In binary classification, the objective is to differentiate between normal and attack traffic, without specifying the type of attack. In this study, we used the binary classification dataset. It contains 37,411 samples and 33 features. The dataset is highly imbalanced, with 33,638 samples of attack traffic and 3,773 samples of normal traffic[13]

3.2 Synthetic Minority Oversampling Technique

The dataset was highly imbalanced, so we leveraged the Synthetic Minority Oversampling Technique (SMOTE) to create distinct but representative samples from the minotity class to balance the dataset [21].

3.3 Principal Component Analysis

Principal Component Analysis (PCA) is a statistical and feature extraction technique that uses an orthogonal transformation to convert a set of correlated variables into a new set of uncorrelated features. It maps instances from an N-dimensional space into a k-dimensional subspace where K is less than N. PCA extracts the most important components from the data by creating linear combinations of the original features that capture the maximum variance. [42]. The PCA process involves the following steps:

3.3.1 Standardization of the raw data. Standardization is the process of scaling the input data so that each feature has mean of zero and a unit variance. The formula is:

$$z_i = \frac{x_i - \mu_i}{\sigma_i} \tag{1}$$

σ_i is the standard deviation of the feature x_i,
μ_i is the mean of feature x_i

3.3.2 Calculation of the Cov matrix to identify correlations between variables. The covariance matrix, C is symmetrical; its size is d x d where d is the number of features. It is calculated as;

$$C = \frac{1}{n-1} Z^T Z \tag{2}$$

where Z is the standardized data matrix (with n samples and d features), Z^T is the transpose of Z. The elements of the covariance matrix C are the covariances between pairs of features.

$$Cov(x_i, x_j) = \frac{1}{n-1} \sum_{k=1}^{n} (z_{ki} - \mu_{zi})(z_{kj} - \mu_{zj}) \tag{3}$$

3.3.3 Determine the eigenvectors and eigenvalues of the covariance matrix to identify the principal components. We calculate the eigenvalues and eigenvectors of the covariance matrix to define the directions of the principal components and the amount of variance captured by each component respectively.

$$Cv = \lambda v \tag{4}$$

C is the covariance matrix,
λ is the eigen value,
v is the eigenvector corresponding to eigenvalue λ

3.3.4 Sort the eigenvalues and select the first k principal components. Once we have the eigenvalues and eigenvectors, we sort the eigenvalues in descending order and select the top k eigenvectors to form a new subspace.

3.3.5 Project the original dataset onto the new feature space, based on the top k eigenvectors. The final step is to multiply the original standardized data matrix, Z by the top k eigenvectors to project the original dataset onto the new k-dimensional subspace V_k [37].

$$Z_{new} = ZV_k \tag{5}$$

3.4 Machine Learning Models

3.4.1 Naive Bayes. Naive Bayes is a probabilistic machine learning algorithm for classification tasks. Naive Bayes is computationally efficient, making it ideal for large and high dimentional datasets. The algorithm is based on the Bayes' theorem which calculates the posterior probability of an event, based on prior knowledge of conditions related to the event.

$$P(C|X) = \frac{P(X|C)\ .\ P(C)}{P(X)} \tag{6}$$

where $P(C|X)$ is the posterior probability, $P(X|C)$ is the likelihood, P(C) is the prior probability of event C and P(X) is the evidence. Given a set of features X = $(x_1, x_2, x_3, ...x_n)$, $P(X|C)$

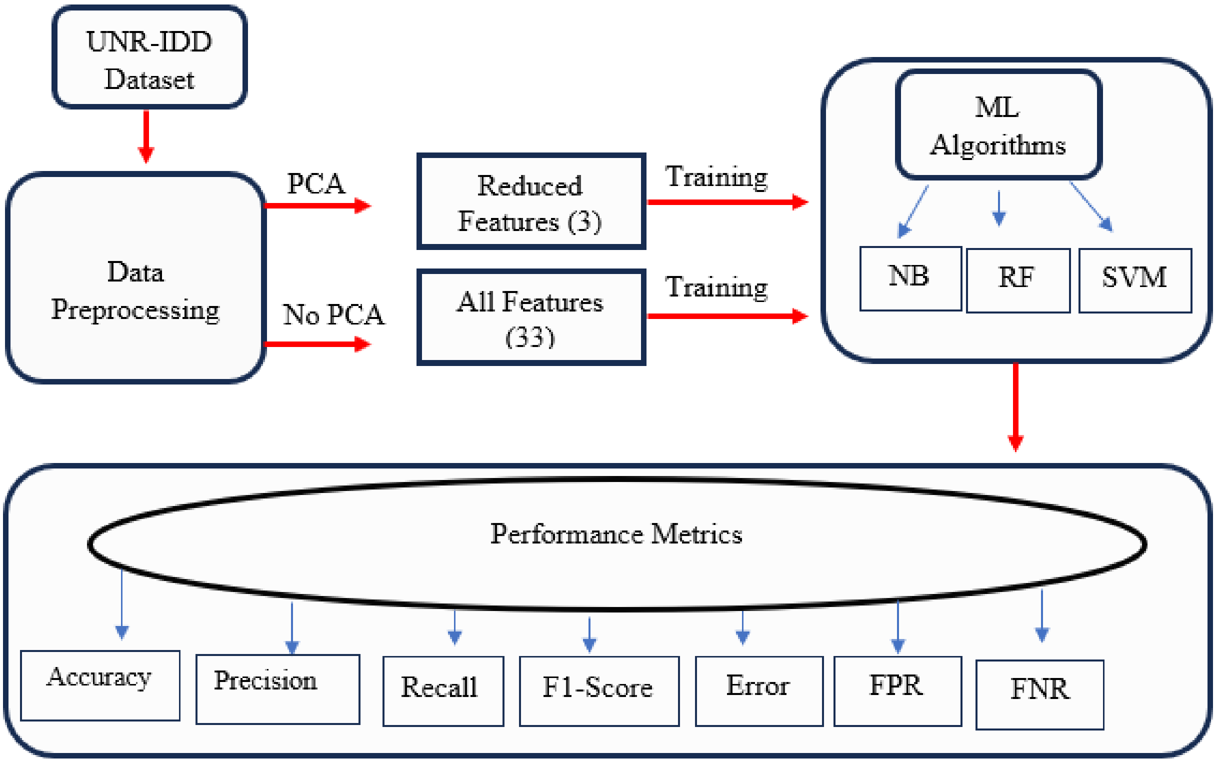


Fig. 1: Flow diagram of the study

is the product of the individual feature probabilities based on the assumption of independence of features.

$$P(C|X) = \frac{P(x_1|C) \; . \; P(x_2|C) \; . \; ....P(x_n|C) \; . \; P(C)}{P(X)} \tag{7}$$

The algorithm calculates the posterior probability for each class and selects the class with the highest probability.

$$C = argmax \left[P(C) \; . \; \prod_{i=1}^{n} \; . \; P(X_I|C) \right] \tag{8}$$

The Naive Bayes algorithm is computationally efficient and performs well on small and noisy datasets [11, 31, 32].

3.4.2 Support Vector Machine (SVM). Support Vector Machine (SVM) is a supervised learning algorithm for classification and regression tasks. SVMs are not limited to linear classification; they also excel at handling non-linear data through the kernel trick. The goal is to find a hyperplane that maximizes the margin, ensuring the best possible separation between the classes [17, 29, 39]. The equation of the optimal hyperplane is defined as;

$$wx^T + b = 0 \tag{9}$$

where b is the bias, x is the input feature vector, and w is the weight vector.

$$wx_i^T + b \geq \; 1 \;\; if \; y_i = 1 \tag{10}$$

$$wx_i^T + b \leq \; 1 \;\; if \; y_i = -1 \tag{11}$$

The SVM algorithm is trained to find the optimal values of w and b that maximize the margin.

$$\frac{1}{||w||^2} \tag{12}$$

[22].

3.4.3 Random Forest (RF). Random forest is an ensemble approach which leverages two levels of randomization to build multiple decision trees for classification and regression tasks. The first layer of randomness is bootstrap aggregation, where the dataset is sampled with replacement. The second layer of randomization occurs at the decision nodes to reduce inter-tree correlation. For classification tasks, the final decision is reached through a majority voting.

$$C = mode(c_1, \; c_2....., c_n) \tag{13}$$

The predictions of the trees, y_i are averaged for regression tasks.

$$Y = \frac{1}{T} \sum_{i=1}^{T} y_i \tag{14}$$

[8]

3.5 Model Training and Evaluation

The dataset was divided into 80% training, 10%validation, and 10% testing subsets. Predictive latency refers to the amount of time between an inference request and prediction for a machine learning model. The training time and predictive latency of the three trained machine learning models before and after applying Principal Component Analysis (PCA), were measured using Python's time module. This process involved setting start and end times, then calculating the elapsed time (CPU time) by subtracting the start time from the end time. This CPU time accurately reflects predictive latency because it excludes any waiting period for I/O operations [16, 40].

4. RESULTS AND DISCUSSION

4.1 Experimental Environment

The models were trained on Google Colab, a powerful cloud-based platform for heavy computational tasks [47]. We leveraged the GPU within this environment (NVIDIA-SMI 535.104.05 TP4) to maximize processing power. The virtual machine featured up 7 processors with 4 core each (Intel(R) Xeon(R) CPU @ 2.00GHz), and up to 53.47 GB of RAM. We used Python 3.10.12 as the runtime, NumPy 1.26.2 and Pandas 2.2.2 for numerical computations. We used Seaborn 0.13.1 and Matplotlib 3.8.4 for the data visualization and Sklearn 1.4.2. for training the machine learning algorithms.

4.2 Effect of PCA on the Performance, Training Time and Detection Latency of the Models

We assessed the performance of the models using precision, accuracy, recall, F1-score, ROC-AUC score, false negative rate, and error rate. The confusion matrices of the three models, both before and after applying PCA, are presented in Fig. 2. A confusion matrix visually represents the performance of a supervised learning algorithm by comparing the predicted and actual labels of the test set [15, 44, 52]. We derived the true positives, TP (attack traffic predicted as attack traffic), true negatives, TN (normal traffic predicted as normal traffic), false positives, FP (normal traffic predicted as attack traffic), and false negatives, FN (attack traffic predicted as normal traffic) from the confusion matrices [23]. We computed the performance metrics for each model using the TP, TN, FP, and FN values and the results presented in table 1.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{15}$$

$$Recall = \frac{TP}{TP + FN} \tag{16}$$

$$Precision = \frac{TP}{TP + FP} \tag{17}$$

$$F1\_score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{18}$$

$$Error = \frac{FP + FN}{TP + TN + FP + FN} \tag{19}$$

$$False\ negative\ rate = \frac{FN}{FN + TP} \tag{20}$$

$$False\ positive\ rate = \frac{FP}{FP + TN} \tag{21}$$

[20].

Our experimental results show that the PCA significantly reduced the training time of all the three models. The training time per sample (TTPS) significantly decreased for all the models. For instance, the TTPS of the Support Vector Machine model decreased significantly from 0.0343 to 0.0177 milliseconds, while the Naïve Bayes model showed a steeper decline (from 0.0041 to 0.00029) milliseconds. However, the PCA increased the TTPS of the Random Forest (RF) model sligtly from 0.0488 to 0.0549 milliseconds. Overall, the application of PCA significantly reduced the TTPS of the SVM and NB models by 48.4 and 92.93% respectively.

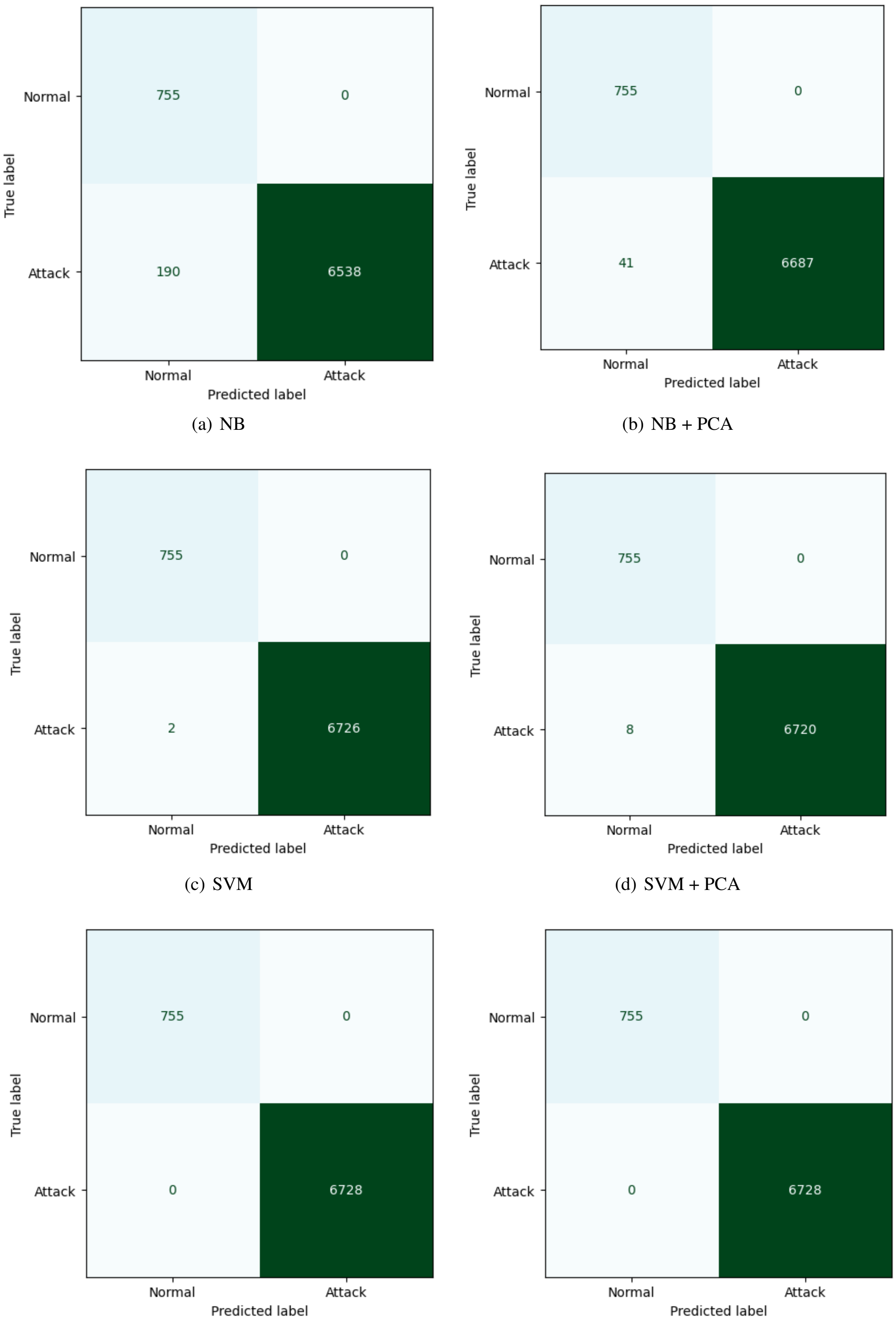
Moreover, the experimental results show that PCA significantly reduced the detection latency (time between inference request and prediction) for both the Naïve Bayes and Support Vector Machine models. For instance, the detection latency per test example for the Naïve Bayes model decreased significantly from 0.0021 milliseconds to 0.00001 milliseconds, and for the SVM model, it decreased from 0.0501 to 0.0251 milliseconds. This represents a whopping 99.52% and 49.90% reduction for Naïve Bayes and SVM respectively. However, the detection latency per example increased marginally for the Random Forest model .

The substantial reduction in both the training time and detection latency for the Naïve Bayes and SVM models suggests that the PCA denoised the dataset to retain only the most relevant information within the first three principal components. By reducing the number of dimensions, the computational cost of the matrix operations performed by these models reduced, further contributing to the reduction in both training and predictive times. For instance, Stojcic and team reported that PCA denoised datasets. Similarly, Bhattacharya and his team reported a significant improvement in performance when using a hybrid PCA-firefly algorithm [42, 43].

In contrast, the PCA transformation had minimal impact on the training time and detection latency of the Random Forest model. This could be attributed to inherent complexity of the model, as both training time and predictive latency are largely influenced by the algorithmic complexity[42].

Table 1 outlines the classification metrics for the three models both before and after PCA. Our analysis shows that, before the PCA transformation, the Naïve Bayes (NB) and Support Vector Machine (SVM) models achieved accuracies of 97.46% and 99.97% respectively. The Random Forest (RF) model outperformed the Support Vector Machine (SVM) and Naive Bayes (NB) models, achieving an accuracy of 100%.

Next, we applied PCA to reduce the dimensionality of the dataset from 33 features to only 3 principal components. Following the PCA transformation, the NB and SVM models achieved accuracy of 99.43% and 99.89% respectively. The PCA transformation significantly improved the performance of the Naive Bayes model, increasing the accuracy by 2.0% compared to the pre-PCA results. The Random Forest model consistently outperformed the Naïve Bayes (NB) and Support Vector Machine (SVM) models across all the performance metrics. This suggests that the PCA transformation denoised the UNR-IDD by eliminating multicollinearity and irrelevant features. The smaller feature space exposed the underlying patterns for better learning by the Naïve Bayes algorithm. Our findings concur with existing literature; for example, Vasan and Surendiran also reported significant improvement in the performance of intrusion detection systems after applying PCA to



(a) NB

(b) NB + PCA

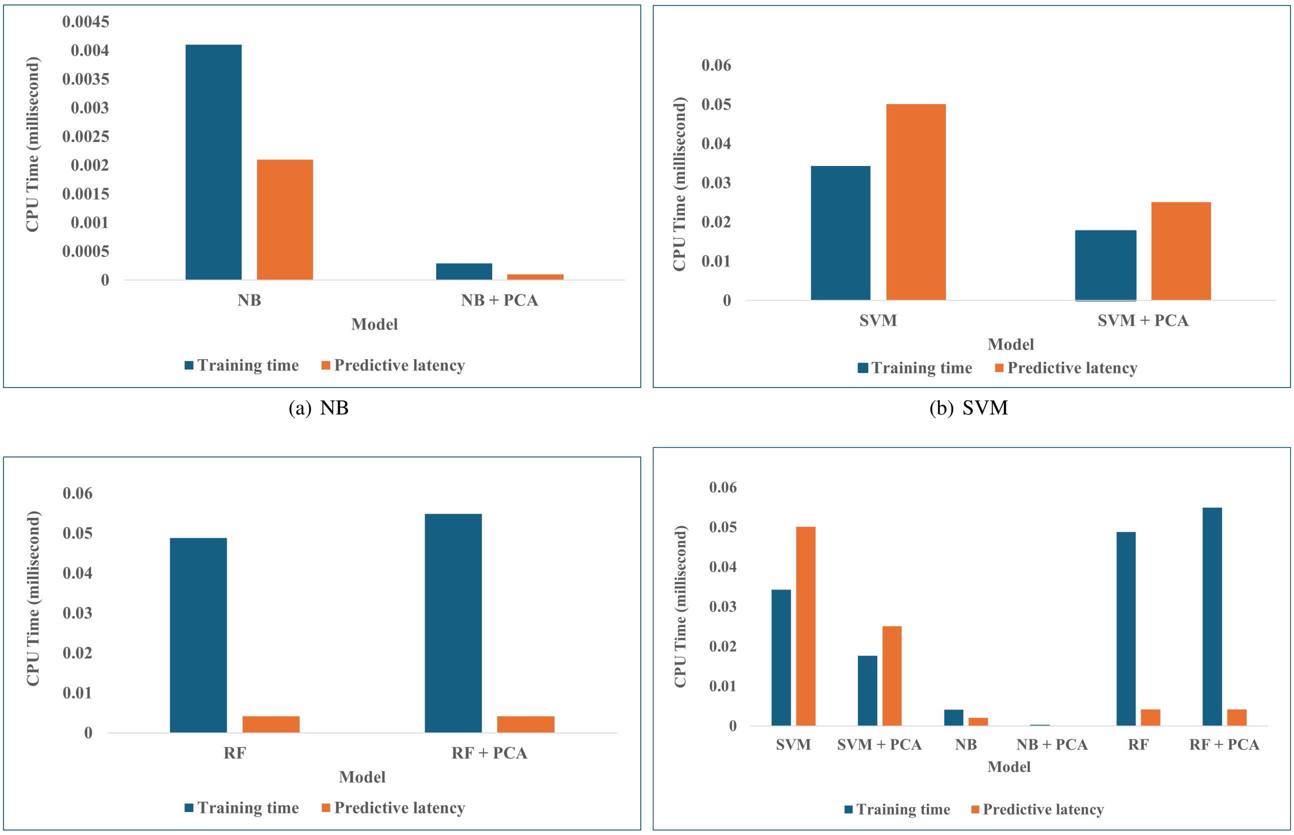
(c) SVM

(d) SVM + PCA

(e) RF

(f) RF + PCA

Fig. 2: Confusion matrices of SVM, NB and RF before and after PCA transformation



(a) NB

(b) SVM

(c) RF

(d) All models

Fig. 3: Training time and detection latency of SVM, NB and RF before and after PCA transformation

Table 1. : Predictive Performances of the Models before and after PCA

M tric	SVM	RF	NB	SVM + PCA	RF + PCA	NB + PCA
Accuracy	0.9997	1.0000	0.9746	0.9989	1.0000	0.9945
Precision	1.000	1.0000	1.0000	1.0000	1.0000	1.0000
Recall	0.9997	1.0000	0.9716	0.9988	1.0000	0.9939
AUC-Score	0.9999	1.0000	0.9859	0.9994	1.0000	0.9970
F1-Score	0.9999	1.0000	0.9857	0.9994	1.0000	0.9969
Error	0.0003	0.0000	0.0254	0.0011	0.0000	0.0055
False Negative Rate	0.0003	0.0000	0.0284	0.0012	0.0000	0.0061
False Positive Rate	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

denoise the KDD Cup 2016 dataset [25].

5. CONCLUSION

In this study, we investigated the impact of PCA on the performance, training time, and detection latency of Support Vector Machine (SVM), Naïve Bayes (NB), and Random Forest (RF) using the UNR-IDD dataset. Our experimental results showed PCA transformation resulted in a significant reduction in both the training times and detection latencies for the SVM and NB models, without compromiusing their performance. The findings indicate that PCA is an effective preprocessing technique for optimizing ML-based IDSs.

While all the three models demonstrated strong performance across the selected metrics, PCA had a particularly significant impact on the NB model, reducing its detection latency by 99.52%, compared to 49.9% and 0% for SVM model and RF models respectively. The combination of NB and PCA stands out as the optimal choice for our proposed robust, low-latency machine learning-based intrusion detection system in network security. We recommend exploring the and investigating other dimensionality reduction to develop more efficient machine learning-based intrusion detection solutions.

Future research should explore hybrid dimensionality reduction techniques, the use of multi-class network intrusion datasets to detect the specific attack vector and real-time deployment architectures. Furthermore, future studies should focus on evaluating optimized ML-based IDSs across diverse and evolving network environments.

6. REFERENCES

[1] Emad E. Abdallah, Wafa' Eleisah, and Ahmed Fawzi Otoom. Intrusion Detection Systems using Supervised Machine Learning Techniques: A survey. *Procedia Computer Science*, 201(C):205–212, 2022.

[2] Zeeshan Ahmad, Adnan Shahid Khan, Cheah Wai Shiang, Johari Abdullah, and Farhan Ahmad. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, 32(1):1–29, 2021.

[3] Nisha Ahuja, Gaurav Singal, Debajyoti Mukhopadhyay, and Neeraj Kumar. Automated DDOS attack detection in software defined networking. *Journal of Network and Computer Applications*, 187(November 2020):103108, 2021.

[4] Mahmood A. Al-Shareeda and Selvakumar Manickam. Man-in-the-Middle Attacks in Mobile Ad Hoc Networks (MANETs): Analysis and Evaluation. *Symmetry*, 14(8), 2022.

[5] Abdullah Alqahtani and Frederick T. Sheldon. A Survey of Crypto Ransomware Attack Detection Methodologies: An Evolving Outlook. *Sensors*, 22(5):1–19, 2022.

[6] James P Anderson. Computer Security Technology Planning Study. *Physical Review E*, Volume I(ESD-TR-73-51):1–43, 1972.

[7] James P Anderson. Computer And Security Journal Catalog, 1980.

[8] Emil D. Attanasi and Timothy C. Coburn. Random Forest. pages 1182–1185, 2023.

[9] Jasmin P Bharadiya. A Tutorial on Principal Component Analysis for Dimensionality Reduction in Machine Learning. *International Journal of Innovative Research in Science Engineering and Technology*, 8(5):2028–2032, 2023.

[10] Fiona Carroll, John Ayooluwa Adejobi, and Reza Montasari. How Good Are We at Detecting a Phishing Attack? Investigating the Evolving Phishing Attack Email and Why It Continues to Successfully Deceive Society. *SN Computer Science*, 3(2):1–10, 2022.

[11] Hong Chen, Songhua Hu, Rui Hua, and Xiuju Zhao. Improved naive Bayes classification algorithm for traffic risk management. *Eurasip Journal on Advances in Signal Processing*, 2021(1), 2021.

[12] Surjeet Dalal, M. Poongodi, Umesh Kumar Lilhore, Fadl Dahan, Thavavel Vaiyapuri, Ismail Keshta, Sultan Mesfer Aldossary, Amena Mahmoud, and Sarita Simaiya. Optimized LightGBM model for security and privacy issues in cyber-physical systems. *Transactions on Emerging Telecommunications Technologies*, 34(6):1–18, 2023.

[13] Tapadhir Das, Osama Abu Hamdan, Raj Mani Shukla, Shamik Sengupta, and Engin Arslan. UNR-IDD: Intrusion Detection Dataset using Network Port Statistics. *Proceedings - IEEE Consumer Communications and Networking Conference, CCNC*, 2023-Janua:497–500, 2023.

[14] Dorothy E Denning and Peter G Neumann. Requirements and model for IDESa real-time intrusion detection expert system, 1985.

[15] Ayesha S. Dina and D. Manivannan. Intrusion detection based on Machine Learning techniques in computer networks. *Internet of Things (Netherlands)*, 16(August):100462, 2021.

[16] Miguel González-Rodríguez, Lorena Otero-Cerdeira, Encarnación González-Rufino, and Francisco Javier Rodríguez-Martínez. Study and evaluation of CPU scheduling algorithms. *Heliyon*, 10(9):e29959, 2024.

[17] Rosita Guido, Stefania Ferrisi, Danilo Lofaro, and Domenico Conforti. An Overview on the Advancements of Support Vector Machine Models in Healthcare Applications: A Review. *Information (Switzerland)*, 15(4), 2024.

[18] Oliver Gulyas and Gabor Kiss. Impact of cyber-Attacks on the financial institutions. *Procedia Computer Science*, 219:84–90, 2023.

[19] Yang Guo. A review of Machine Learning-based zero-day attack detection: Challenges and future directions. *Computer Communications*, 198(November 2022):175–185, 2023.

[20] Md Alamgir Hossain and Md Saiful Islam. A novel hybrid feature selection and ensemble-based machine learning approach for botnet detection. *Scientific Reports*, 13(1):1–28, 2023.

[21] Md Alamgir Hossain and Md Saiful Islam. Ensuring network security with a robust intrusion detection system using ensemble-based machine learning. *Array*, 19(May):100306, 2023.

[22] Shujun Huang, C. A.I. Nianguang, Pedro Penzuti Pacheco, Shavira Narandes, Yang Wang, and X. U. Wayne. Applications of support vector machine (SVM) learning in cancer genomics. *Cancer Genomics and Proteomics*, 15(1):41–51, 2018.

[23] Fayaz Itoo, Meenakshi, and Satwinder Singh. Comparison and analysis of logistic regression, Naïve Bayes and KNN machine learning algorithms for credit card fraud detection. *International Journal of Information Technology (Singapore)*, 13(4):1503–1511, 2021.

[24] Yakub Kayode Saheed, Aremu Idris Abiodun, Sanjay Misra, Monica Kristiansen Holone, and Ricardo Colomo-Palacios. A machine learning-based intrusion detection for detecting internet of things network attacks. *Alexandria Engineering Journal*, 61(12):9395–9409, 2022.

[25] K. Keerthi Vasan and B. Surendiran. Dimensionality reduction using Principal Component Analysis for network intrusion detection. *Perspectives in Science*, 8:510–512, 2016.

[26] Deepshikha Kumari, Abhinav Sinha, Sandip Dutta, and Prashant Pranav. Optimizing neural networks using spider monkey optimization algorithm for intrusion detection system. *Scientific Reports*, 14(1):1–16, 2024.

[27] Fatima Ezzahra Laghrissi, Samira Douzi, Khadija Douzi, and Badr Hssina. Intrusion detection systems using long short-term memory (LSTM). *Journal of Big Data*, 8(1), 2021.

[28] Xiao Xue Li, Dan Li, Wei Xin Ren, and Jun Shu Zhang. Loosening Identification of Multi-Bolt Connections Based on Wavelet Transform and ResNet-50 Convolutional Neural Network. *Sensors*, 22(18), 2022.

[29] Batta Mahesh. Machine Learning Algorithms - A Review — Enhanced Reader. (October), 2019.

[30] M. Manjula, Venkatesh, and K. R. Venugopal. Cyber Security Threats and Countermeasures using Machine and Deep Learning Approaches: A Survey. *Journal of Computer Science*, 19(1):20–56, 2023.

[31] Vivek Narayanan, Ishan Arora, and Arjun Bhatia. Fast and accurate sentiment classification using an enhanced Naive Bayes model. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8206 LNCS:194–201, 2013.

[32] P. J.Beslin Pajila, B. Gracelin Sheena, A. Gayathri, J. Aswini, M. Nalini, and R. Siva Subramanian. A Comprehensive Survey on Naive Bayes Algorithm: Advantages, Limitations and Applications. *Proceedings of the 4th International Conference on Smart Electronics and Communication, ICOSEC 2023*, pages 1228–1234, 2023.

[33] Jeonghoon Park, Jinsu Kim, B. B. Gupta, and Namje Park. Network Log-Based SSH Brute-Force Attack Detection-Model. *Computers, Materials and Continua*, 68(1):887–901, 2021.

[34] Fakhra Parveen, Sajid Iqbal, Gohar Mumtaz, and Muqaddas Salahuddin. Real-Time Intrusion Detection with Deep Learning : Analyzing the UNR Intrusion Detection Dataset. 07(02), 2024.

[35] Jawad Rasheed, Alaa Ali Hameed, Chawki Djeddi, Akhtar Jamil, and Fadi Al-Turjman. A machine learning-based framework for diagnosis of COVID-19 from chest X-ray images. *Interdisciplinary Sciences – Computational Life Sciences*, 13(1):103–117, 2021.

[36] Bipraneel Roy and Hon Cheung. A Deep Learning Approach for Intrusion Detection in Internet of Things using Bi-Directional Long Short-Term Memory Recurrent Neural Network. *2018 28th International Telecommunication Networks and Applications Conference, ITNAC 2018*, pages 1–6, 2018.

[37] Nema Salem and Sahar Hussein. Data dimensional reduction and principal components analysis. *Procedia Computer Science*, 163:292–299, 2019.

[38] Jitendra Kumar Samriya, Surendra Kumar, Mohit Kumar, Huaming Wu, and Sukhpal Singh Gill. Machine Learning Based Network Intrusion Detection Optimization for Cloud Computing Environments. *IEEE Transactions on Consumer Electronics*, PP(Xx):1, 2024.

[39] K. Saravanan, R. Banu Prakash, C. Balakrishnan, Gade Venkata Prasanna Kumar, R. Siva Subramanian, and M. Anita. Support Vector Machines: Unveiling the Power and Versatility of SVMs in Modern Machine Learning. *3rd International Conference on Innovative Mechanisms for Industry Applications, ICIMIA 2023 - Proceedings*, (Icimia):680–687, 2023.

[40] Serhack. How to Measure Execution Time of a Program - SerHack.

[41] Sugandh Seth, Gurvinder Singh, and Kuljit Kaur Chahal. A novel time efficient learning-based approach for smart intrusion detection system. *Journal of Big Data*, 8(1), 2021.

[42] Mirko Stojčić, Milorad K. Banjanin, Milan Vasiljević, Aleksandar Stjepanović, and Zoran Ćurguz. PCA modeling of extraction and selection of variables influencing LTE network delay in urban mobility conditions. pages 117–125, 2023.

[43] Bhattacharya Sweta, Rama Krishnan S. Siva, Kumar Maddikunta Praveen, Kaluri Rajesh, Singh Saurabh, Reddy Gadekallu Thippa, Alazab Mamoun, and Usman Tariq. A Novel PCA-Firefly Based XGBoost Classification Model for Intrusion Detection in Networks. *Electronics (Switzerland)*, 9(2):219, 2020.

[44] Abdulrahman Takiddin, Muhammad Ismail, Mahmoud Nabil, Mohamed M. E. A. Mahmoud, and Erchin Serpedin. Detecting Electricity Theft Cyber-Attacks in AMI Networks Using Deep Vector Embeddings. *IEEE Systems Journal*, 15(3):4189–4198, 2020.

[45] Hatice Beyza Taşçı, Serkan Gönen, Mehmet Ali Barışkan, Gökçe Karacayılmaz, Birkan Alhan, and Ercan Nurcan Yılmaz. Password Attack Analysis Over Honeypot Using Machine Learning Password Attack Analysis. *Turkish Journal of Mathematics and Computer Science*, 13(2):388–402, 2021.

[46] R. Vinayakumar, Mamoun Alazab, K. P. Soman, Prabaharan Poornachandran, Ameer Al-Nemrat, and Sitalakshmi Venkatraman. Deep Learning Approach for Intelligent Intrusion Detection System. *IEEE Access*, 7:41525–41550, 2019.

[47] Xiaojuan Wang, Yun Zhong, Lei Jin, and Yabo Xiao. Scale Adaptive Graph Convolutional Network for Skeleton-Based Action Recognition. *Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/Journal of Tianjin University Science and Technology*, 55(3):306–312, 2022.

[48] Zhen Yang, Xiaodong Liu, Tong Li, Di Wu, Jinjiang Wang, Yunwei Zhao, and Han Han. A systematic literature review of methods and datasets for anomaly-based network intrusion detection. *Computers and Security*, 116, 2022.

[49] Huanhuan Yuan, Yuanqing Xia, Yuan Yuan, and Hongjiu Yang. Resilient strategy design for cyber-physical system under active eavesdropping attack. *Journal of the Franklin Institute*, 358(10):5281–5304, 2021.

[50] Huanhuan Yuan, Yuanqing Xia, Yuan Yuan, and Hongjiu Yang. Resilient strategy design for cyber-physical system under active eavesdropping attack. *Journal of the Franklin Institute*, 358(10):5281–5304, 2021.

[51] Diyar Qader Zeebaree, Habibollah Haron, Adnan Mohsin Abdulazeez, and Dilovan Asaad Zebari. Trainable Model Based on New Uniform LBP Feature to Identify the Risk of the Breast Cancer. *2019 International Conference on Advanced Science and Engineering, ICOASE 2019*, pages 106–111, 2019.

[52] Changming Zhu and Daqi Gao. Influence of data preprocessing. *Journal of Computing Science and Engineering*, 10(2):51–57, 2016.