Vol.4, No.2, Page 1656- 1664, 2026 https://doi.org/10.59628/jast.v4i2.2270 ISSN 2958-9568 JAST

SUN PUBLISHER

مجلة جامعة صنعاء للعلوم التطبيقية والتكنولوجيا

Sana'a University Journal of Applied Sciences and Technology

https://journals.su.edu.ye/index.php/jast/

# Attacks Detection Model Based on a Machine Learning Algorithm

**Essra Abd wazkool [1]*, Abdulrahman A. Alsabri[2] and Suad Mohammed Othman[3]**

[1]**Department of Computer Science, Faculty of Computer & IT, Sana'a University, Sana'a, Yemen,**
[2]**Department of Information System, Faculty of Computer & IT, Sana'a University, Sana'a, Yemen,**
[3]**Department of Information Technology, Faculty of Computer & IT, Sana'a University, Sana'a, Yemen**

*Corresponding author:e.wazkool@gmail.com

## Abstract

With the rapid developments in data volume and the complexity of cyber-attacks, attack detection systems face increasing challenges. This paper presents a model based on a Support Vector Machine (SVM) algorithm to detect attacks. To improve detection accuracy and reduce computational complexity, the Principal Component Analysis (PCA) algorithm was used as a first stage to select the most important features in the NSL-KDD dataset. The proposed model was applied to the NSL-KDD dataset. The results showed that using the SVM and PCA algorithms helps reduce the data dimensions, leading to improved classification accuracy and reduced computational complexity of the model. This model provides a machine learning-based detection system that can effectively identify attacks, which leads to enhancing network security against complex threats.

## Article Info

**Keywords:**
Attacks Detection, Cloud Computing, Machine Learning, Classification, Support Vector Machine, Principal Component Analysis.

**Article History:**
**Received:** 20-September-2025,
**Revised:** 20-December-2025,
**Accepted:** 27-December-2025,
**Published:** 28 February 2026.

## 1. INTRODUCTION

Cyber threats are constantly increasing in the current era owing to increased reliance on electronic networks and digital services, which poses major challenges to cybersecurity. Therefore, it is necessary to develop effective systems that are capable of detecting attacks. It helps to protect networks from attacks that can lead to the loss of sensitive data and information. Intrusion detection (ID) is a critical component of network security that aims to detect and alert malicious activities in a network [1]. An intrusion detection system (IDS) is a hardware or software that monitors and analyzes data to detect threats in a system or network [2]. An IDS alerts administrators; this alert helps the administrator find and determine the vulnerability in a system or network [3]. In addition, an IDS detects cyberattacks in a network using rule-based or anomaly based methods, and alerts the cybersecurity team in a company [4]. IDSs use three techniques for detecting attacks: anomaly based detection, signature-based detection, and hybrid detection [5]. Machine learning is used by an IDS to achieve an accuracy that exceeds the constraints of existing rule-based techniques [6]. The authors [7, 8] explained many techniques to detect attacks, including signature-based detection, Setting Thresholds, Connection Limits, User Behavior Analysis, and Machine Learning.

### Signature-based detection

Signature-based detection systems store known attack patterns, and then match the attacks to the stored patterns. In this type of system, the known attacks can be detected more efficiently. [9, 10]. However, new attacks are not detected, which is a major problem, particularly in cloud environments where attacks are increasing.

### 1. Anomaly-based detection

The process of analyzing data to identify patterns deviates from normal behavior [11]. Anomaly based learning of normal behavior using machine learning algorithms [12]. Machine learning is a powerful tool for detecting attacks on networks and devices. Supervised learning was used to classify attack types based on past data.

Unsupervised learning is also used to identify anomalies and abnormal behaviors [13].

Research is ongoing to find an effective way to detect attacks using machine learning. In this paper, we present a model based on machine learning for an ID.

The main purpose of the proposed model is to achieve high performance while reducing the time spent on the training and testing processes, making the model applicable in environments that require real-time operation.

Contributions of this paper:

1. In the first stage of the model, preliminary analysis and cleaning techniques were applied to guarantee the validity of the data, remove inconsistencies, and prepare for effective training and accurate results.
2. The feature selection stage is presented using the PCA algorithm.
3. The attack classification stage is presented using the SVM algorithm.

The remainder of the paper is structured as follows: Section II presents related work. Section III provides the proposed method. Section IV follows this and introduces the results and discussion. Finally, Section V presents our conclusions and future work.

**2. Related Work**

Many studies have provided attack- or intrusion-detection models. In this section, we summarize some of these studies.

Sun et al. proposed an IDS for innovative health platforms that combined AdaBoost and particle swarm optimization algorithms to detect and classify malware-related records. This study employed the NSL KDD dataset [14].

In 2024, Talukder et al. [1] proposed an innovative approach that combines machine learning (ML) with the Synthetic Minority Oversampling Technique Tomek Link (SMOTE-TomekLink) algorithm to detect intrusions. The results of a balanced dataset significantly improved the detection accuracy. In addition, the study used feature scaling by standardization to facilitate precise detection and training. To counteract imbalanced datasets, the authors employed the SMOTE-Tomek resampling method, which reduces overfitting and underfitting problems.

Saleh et al. [15] employed Stochastic Gradient Descent (SGD) and Gaussian Nave Bayes (GNB) machine learning algorithms. The authors used PCA and singular value decomposition on network traffic data to reduce the burden on the machines. The authors used the WSN-DS and IoMT datasets to evaluate the proposed SG–IDS model. The proposed SG-IDS model achieved 96% accuracy, 96% recall, 97% F1 measurement, and 0.87 accuracy and precision of 1.00, respectively.

Ali et al. [16] proposed individual models and an ACLR model for the botnet identification system using convolutional neural network (CNN), recurrent neural network (RNN), long short-term memory (LSTM), and artificial neural network (ANN) algorithms. The authors conducted a performance comparison between both models. The UNSW-NB15 dataset was used to evaluate this model. The experimental results show that the proposed model achieved an accuracy of 0.9698.

The proposed model [17] presented a cloud IDS model using random forest (RF), decision trees (DTs), and SVM algorithms for detecting attacks. In addition, the authors used graphic visualization for feature engineering. In addition, the model was evaluated using two datasets, Bot-IoT and NSL-KDD. The results of the model were 98.3% ACC and 99.99% ACC for both the datasets.

Logeswari et al. [18] proposed a novel Hybrid Feature Selection-LightGBM (HFS-LGBM IDS) for Software-Defined Networking (SDN). For feature selection, a two-stage hybrid method was used to decrease dimensionality and obtain optimal data. The correlation-based feature selection algorithm was used in the first stage to extract the initial set of data. In the second stage, the researchers used a random-forest-based iterative elimination method. The researcher then applied the LightGBM algorithm to detect the different types of attacks. The researcher used the NSL-KDD dataset and adopted accuracy, precision coefficient, recall, and F-measure metrics to measure the effectiveness of the proposed model.

To enhance the performance of the IDS, Bakro et al. [19] proposed an IDS model using a hybrid algorithm for feature selection by combining a genetic algorithm (GA) and grasshopper optimization algorithm (GOA). The authors used random forest (RF) to classify optimal features. The authors addressed imbalanced data using a hybrid approach: the adaptive synthetic (ADASYN) algorithm and random undersampling (RUS). The authors evaluated the proposed approach using three datasets, CICDDoS2019, UNSW-NB15, and CIC Bell DNS EXF 2021, with accuracies of 99%, 98%, and 92%, respectively.

Ahmed [20] recommended applying machine learning methods to detect intrusions in wireless sensor networks( WSNs). Implementing an SVM with stochastic gradient descent (SGD) improves the detection precision. To improve the performance of recommendation systems, the research also suggests integrating context knowledge, sometimes referred to as context awareness, PCA, and singular value decomposition (SVD), to reduce the initial traffic data to decentralize the system's computational load. A VG-IDS strategy is used to further classify the identified network threats. The accuracy, recall, and F1-measure rates improved to improved results of 98%, 97%, and 96%, respectively.

The authors of [21] proposed a hybrid and layered (IDS). The author used a combination of naive Bayes, random forest, fuzzy logic, decision trees, ANN, decision supporting machines, K Nearest neighbor, and K-means

machine learning algorithms. The authors used the CfsSubsetEval and WrapperSubsetEval feature selection algorithms to enhance the ID for different attacks and reduce the dataset size. The NSL-KDD dataset was used to train and test the proposed model.

Jayaraj et al. [5] presented a novel method for selecting features called the Cumulative Distribution Function gradient (CDF-g) algorithm, which researchers used as a first step in generating a subset and then used the data perturbation ensemble method to generate other subsets. In this study, researchers applied SVM, RF, DT, and ANN algorithms to detect phishing attacks.

Alqahtani et al.[22] employed several popular machine learning classification techniques, including a Bayesian Network, Naïve Bayes classifier, RF, Decision Tree, Decision Table, and ANN, to evaluate ID in cyber security.

A researcher [23] proposed using oversampling and undersampling to solve the problem of data imbalance that affects the accuracy of a model. The researchers also applied five ML algorithms to detect attacks in the model, where DT and RF, XGBoost, CatBoost, and multilayer perception were used.

Alashhab et al. [24] presented a method to improve DDoS detection using online learning to detect expected attacks, and an ensemble (Passive-Aggressive, BernoulliNB, SGD Classifier, and MLP Classifier) to classify attacks. The researchers also used CICDDoS2019, slow-read-DDoS, and InSDN datasets for training and evaluating the model on SDN (Mininet and Ryu) simulators. The accuracy is 99.2%.

To build advanced artificial intelligence-based machine-learning techniques, Raza et al. [25] proposed a machine learning method to identify threats quickly and efficiently. The author used the Class Probability Random Forest (CPRF) technique, and a k-fold strategy was used to verify the performance on the CICIDS2017 dataset. The results indicated that the performance was high (99.9%).

Li et al. [26] presented a framework based on the comparison between selection and extraction feature techniques in an IoT network and used the decision tree algorithm for binary and multiclass classification to classify attacks based on the F1-score and runtime. The authors used the TON-IoT dataset to test and estimate the framework.

Suad et al. [2] introduced an ID model using ML and big data based on the SVM algorithm and the Spark Big Data Platform. The authors used the ChiSqSelector algorithm for the feature selection. The KDD dataset was used to test and train the proposed model. Table 1 shows a summary of related work based on the dataset used in the research, the feature selection algorithm used, and the classification algorithm.

The research [27] developed a model that combines the PCA algorithm and SVM to detect attacks, and applied optimization to improve the model's performance. The model is tested using the NSLKDD dataset. Despite the high performance achieved by the model, it did not address pre-processing steps such as data encoding, which are necessary when dealing with different types of data. The results of the model before optimization were lower than those of the proposed model, indicating that the model is unable to detect all types of attacks. The proposed model differs from previous work in that it uses a smaller number of principal components extracted by the Principal Component Analysis (PCA) algorithm; only 26 principal components were obtained compared to 31 components in the previous study.

Despite the reduction in the number of components, the model achieved high performance in attack detection, indicating a reduction in training and execution time, making the model more efficient and applicable in cloud environments that require rapid detection and processing of large amounts of data in real time.

A researcher [28] applied the SVM algorithm with PCA to detect attacks. The average accuracy of the model reached 91%, which indicates that the proposed model outperforms this model by 10%, in addition to the fact that the model used multi-classification, while our model uses binary classification.

## 3. Proposed Work

The proposed model presents an ML application for ID using the NSL-KDD dataset. The proposed model consists of seven steps, as illustrated in Figure. 1. The following sections explain the model in detail.

One of the most significant considerations in an IDS is selecting a suitable dataset. Thus, data collection is a vital task. As network attacks change, the use of old datasets may not provide the objective and required results. In this study, the author used an NSL-KDD dataset.

The NSL-KDD dataset is an enhanced version of the KDD Cup dataset, which is one of the most common datasets used for assessing performance in IDS. The NSL-KDD dataset was created to resolve the issues in the KDD Cup dataset [29].

**The NSL-KDD Dataset** contains records and a more balanced attack distribution. The NSL-KDD dataset contains **41 features** that define each network connection, with a **label** representing whether the connection is an attack or normal [30].

**Attack Categories**: Attacks in NSL-KDD are categorized into four categories, as shown in Figure 2.

In the model, the authors began collecting data in the first stage, which was the basis for training and testing the proposed model. In the second stage, the data processing step was performed, and missing or empty values were removed, followed by the data coding and normalization process. In the next stage, we used the PCA algorithm to extract features from the data. Subsequently, the authors divided the data into training and testing datasets. Finally, the training and testing stages of the model are used to classify and detect attacks using

**Table 1.** summary of related works.

| paper | data | Features selection | classification |
|---|---|---|---|
| [14] | KDD dataset | Particle swarm optimization | AdaBoost |
| [31] | CSE CIC IDS 2018 | Firefly Algorithm | Decision Tree (DT) |
| [15] | WSN-DS and an IoMT | PCA | Stochastic Gradient Descent (SGD) and Gaussian Nave Bayes (GNB) |
| [32] | BoT-IoT | – | Random Forest (RF), An optimized gradient tree boosting system (XGBoost) |
| [17] | Bot-IoT and NSL-KDD. | A graphic visualization | random forest (RF), decision trees (DTs), and SVM algorithms |
| [18] | NSL-KDD | The correlation-based feature selection algorithm | LightGBM algorithm |
| [19] | CICDDoS2019, UNSW-NB15, and CIC Bell DNS EXF 2021 | Genetic algorithm (GA) , Grasshopper optimization algorithm (GOA). | A random forest (RF) |
| [20] | WSN-DS | PCA, and a singular value decomposition (SVD) | SVM |
| [21] | NSL-KDD | CfsSubsetEval, WrapperSubsetEval feature selection algorithms. | Naive Bayes, Random Forest, Fuzzy Logic, Decision Trees, ANN and Decision Supporting Machines, K Nearest Neighbour, and K-means machine learning algorithms |
| [5] | Real-CyberSecurity-Datasets | Cumulative Distribution Function gradient (CDF-g) algorithm | SVM, RF, DT, and ANN algorithms |
| [22] | KDD'99 cup datasets | – | Bayesian Network, Naïve Bayes classifier, RF, Decision Tree, Decision Table, ANN |
| [23] | IOTID20 dataset. | – | Decision tree and random forest, XGBoost, CatBoost, and multi-layer perception |
| [24] | CICDDoS2019, slow-read-DDoS, and InSDN | – | Passive-Aggressive, BernoulliNB, SGD Classifier, and MLP Classifier |
| [25] | CICIDS2017 | – | The class Probability Random Forest (CPRF) technique |
| [26] | TON-IoT | Correlation matrix, PCA | Decision tree algorithm |
| [2] | KDD | ChiSqSelector | SVM |
| [27] | NSL-KDD | PCA | SVM |

the SVM algorithm, evaluate the model, and determine the best performance for detecting attacks.

**Data cleaning:**

In this step, missing and infinite data were removed. For increased accuracy and model results, the authors removed irrelevant data that were non-informative or incorrect values.

**Normalization:**

Normalization is a crucial preprocessing technique used in machine learning and research to scale features to a typical range, thereby enhancing model performance. This technique is important when datasets contain features with varying amounts, units, or ranges.

In the proposed model, z-score normalization was applied to each numerical feature using the StandardScaler method provided by the scikit-learn library. This transformation shifts the distribution of each feature to have a mean of zero and a standard deviation of one, while preserving the overall distribution of the variable and ensuring that all features are on comparable scales. The normalization of each feature $\chi_i$ is mathematically performed by:

$$\acute{\chi}_i = \frac{x_i - \mu_i}{\sigma_i} \tag{1}$$

where $\mu_i$ and $\sigma_i$ are the mean and standard deviation of the feature *i*, respectively.

This normalization step improves convergence and stability in the SVM.

**Label Encoding:**

Label encoding is a method used to convert categorical data to numerical data suitable for machine learning that accepts numerical values. These values can be encoded by various techniques. Among these, one-hot encoding is the most prevalent method. Hence, the authors converted categorical data into numerical data by applying the one-hot encoding approach.

**Dataset Splitting:**

Data splitting is a necessary step in training and evaluating the models [33, 34]. To train and test the model, the authors split the dataset into training and testing subsets in a 70:30 ratio, with 70% of the data reserved for training and the remaining 30% reserved for testing. During the training phase, the model exclusively accesses the train-

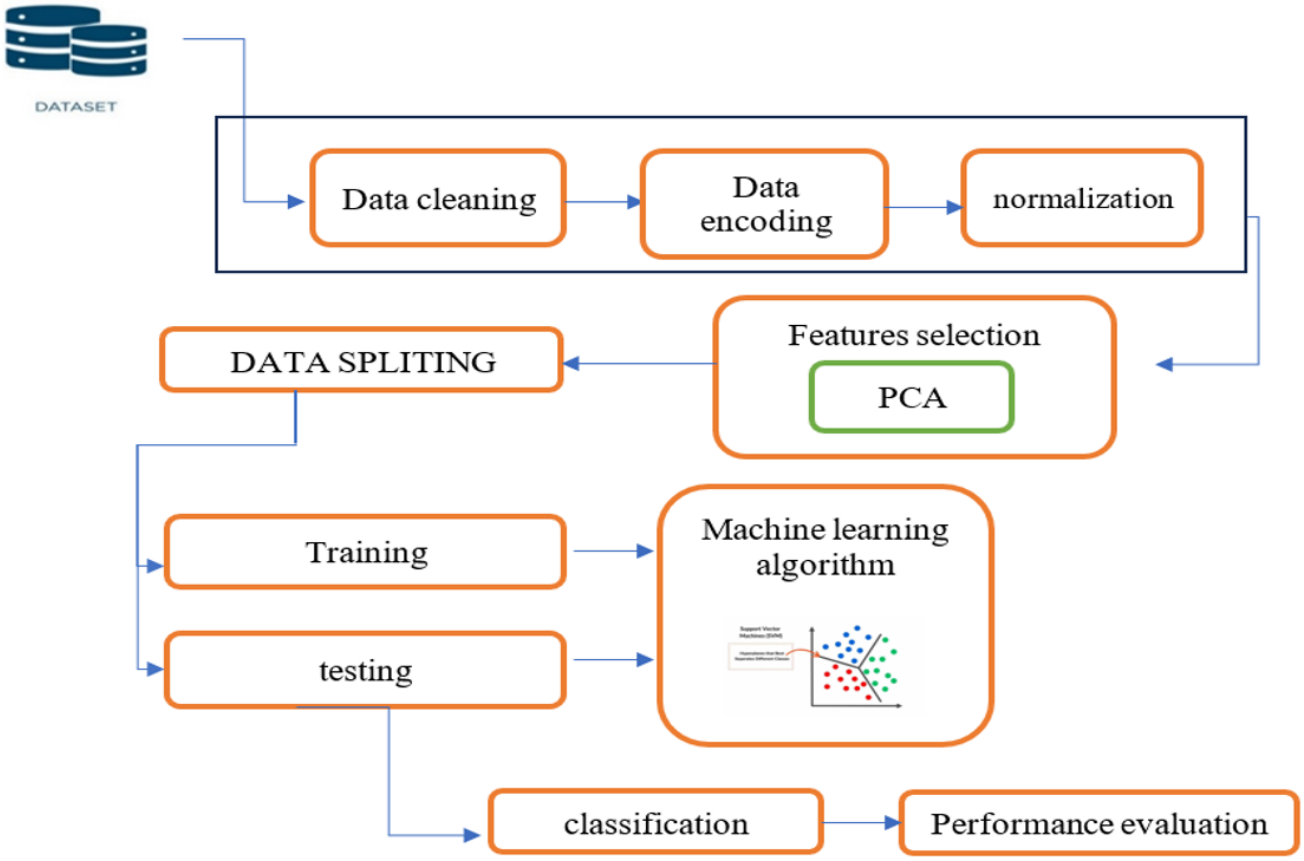


**Figure 1.** proposed model.

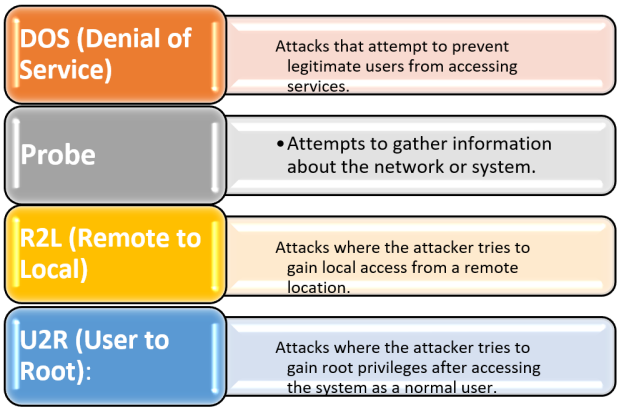


**Figure 2.** attacks categories

ing data to learn patterns and relationships, whereas the test data are introduced through machine learning only after training is complete.

**Feature Selection:**

The NSL-KDD dataset contained 41 features. Many features can considerably affect the performance of machine-learning development. Feature selection is an essential step for ensuring the success of machine learning. Another vital aspect of the effectiveness of an IDS approach in various situations is the selection of features. Generating features through deep-supervised and unsupervised learning also facilitates model debugging, making the learning outcomes more interpretable. Additionally, it significantly accelerated the model training process and improved the training accuracy.

To select the features in the proposed model, component analysis was used, which is a feature selection algorithm, and the number of results or components was equal to 26.

PCA is one of the most widely applied algorithms in Machine Learning and Data Preprocessing.

This method transforms a large set of correlated features into a smaller set of uncorrelated variables called principal components while retaining most of the variance (i.e., information) present in the original dataset.

PCA essentially finds the directions in which the data varies the most and projects it on the same in a simplified manner.

Standardized network traffic data were projected onto a low-order subspace using the PCA algorithm, which retained the most informative and uncorrelated components.

The following 26 principal components captured more than 95% of the total variance and were used as inputs for the SVM classifier for attack detection.

The integration of these improves classification accuracy and reduces computational complexity.

The following algorithm defines how the PCA works in feature selection.

The following the equations used in PCA algorithm to

**Algorithm for PCA:**

**Input:** NSL-KDD dataset with numerical and categorical features

**Output:** Reduced feature set $Z$

1. Normalize numeric features using `StandardScaler`
2. Encode categorical features using One-Hot Encoding
3. Combine encoded and scaled features into matrix $X'$
4. Compute the covariance matrix $\Sigma$ of $X'$
5. Calculate eigenvalues and eigenvectors of $\Sigma$
6. Sort eigenvectors by descending eigenvalues
7. Select $k$ components that explain $\geq 95\%$ of total variance
8. Project $X'$ onto selected eigenvectors $\rightarrow Z = X' \times W$
9. Return $Z$ as the reduced dataset

select features.

The Standardized Data Matrix:

$$X' = \begin{bmatrix} x'_{11} & x'_{12} & \dots & x'_{1d} \\ x'_{21} & x'_{22} & \dots & x'_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x'_{n1} & x'_{n2} & \dots & x'_{nd} \end{bmatrix} \quad (2)$$

Where:

*n* = number of samples.

*d* = number of features.

The Compute Covariance Matrix by formula:

$$C = \frac{1}{n-1}(X')^T X' \quad (3)$$

Where:

*C* = covariance matrix *(d×d)*.

Covariance elements:

$$cov = \frac{1}{n-1}\sum_{i=1}^{n}(x'_{ij})(x'_{jk}) \quad (4)$$

Eigen Decomposition:

$$c\,\omega = \lambda\omega \quad (5)$$

Where:

$\omega$ : eigenvector.

$\lambda$: eigenvalue.

Sort Eigenvalues in Descending Order:

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d \quad (6)$$

Explained Variance of Each Component and Cumulative Explained Variance:

$$EVR_i = \frac{\lambda_i}{\sum_{j=1}^{d}\lambda_{-j}} \;, CEVR_k = \sum_{i=1}^{k} EVR_i \quad (7)$$

Where $EVR$ = Explained Variance Ratio.

Projection to Reduced Feature Space:

$$Z = X'W_k \quad (8)$$

Where:

$X'$: standardized data.

$W_k$ : Eigenvectors (selected PCs).

Z: transformed feature matrix (PC scores).

**Model training:**

At this stage, the data are passed to training, and the features obtained in the previous steps are passed to the training stage in the model. Here, the SVM algorithm is used to train on the previous data, which represents the attack data, and then passes it to the testing stage.

An SVM is a set of supervised learning methods used mainly for classification, outlier detection, and regression.

The advantages of SVM [27]:

- Deliver high efficiency for high-dimensional feature spaces.
- Performing well when the number of features is greater than the number of samples.
- Ensuring memory efficiency because the latter requires only a subset of the training points (support vectors) to make decisions.
- Flexibility by using choice kernel functions in the decision-making step: Although a standard kernel is used more often, one can design custom kernels per particular problem.

The parameter used in the model shown in Table 2.

**Table 2.** SVM Hyperparameter

| parameters | value |
|---|---|
| Kernal type | linear |
| gamma | scale |
| C | 1.0 |

A linear kernel was chosen because it can be scaled and interpreted in a high-dimensional feature space. The proposed model performs binary classification.

**Algorithm for SVM**

**Input:** $X_{\text{train}}$, $y_{\text{train}}$, $X_{\text{test}}$

**Output:** Predicted attack labels $y_{\text{pred}}$

1. Initialize SVM with kernel = `linear` and $C = 1.0$.
2. Train classifier: `clf.fit`$(X_{\text{train}}, y_{\text{train}})$.
3. Predict labels: $y_{\text{pred}}$ = `clf.predict`$(X_{\text{test}})$.
4. Evaluate results using accuracy, precision, recall, and F1-score.
5. Return $y_{\text{pred}}$ and performance metrics.

**Model Testing:**

The performance of the model was evaluated at this stage. After the training stage, the model's performance must be evaluated based on certain criteria to measure and improve the efficiency of the model. The model was tested on the test data and evaluated based on several criteria, such as accuracy, recall, and F-score.

**Validation:**

The goal of the validation in this research is to reveal that the proposed model reliably detects intrusions. The metrics used to validate our model are as follows.

**Accuracy:**

Accuracy measures how closely the predicted values align with the true values, which helps to assess the effectiveness of the model in detecting attacks. It is defined as the level of agreement between the predicted and actual values, according to Equation Eq. (9).

$$A = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \tag{9}$$

**Recall**

Recall refers to the ability to identify and detect intrusions accurately. It is used to calculate the number of true positives that are correctly predicted and is measured using Eq. (10).

$$S_N = \frac{T_p}{T_p + F_n} \tag{10}$$

**Precision**

Precision measures the number of correct positive predictions in relation to the total positive predictions, and is calculated using Eq. (11).

$$P = \frac{Tp}{Tp + F_p} \tag{11}$$

**F1-score**

The F1-score measurements for identifying the efficiency of detection intrusions were measured using Eq. (12).

$$\text{F-measure} = \frac{2 * T\rho}{(2 * Tp + F_p + F_N)} \tag{12}$$

## 2. RESULTS

In this section, we present the most important results of this study, using the performance criteria described above. The proposed model was implemented using Python on Anaconda on a PC with a Core i7 processor and 16 GB of memory. Table 3 illustrates the results and Figure 3. Figure 4 shows the confusion matrix of the model. The comparative analysis of the results of the proposed model with some previous works that used the same dataset, which were discussed previously, is displayed in Table 4.

**Table 3.** The results of a model

| ACCURACY | PRECISION | RECALL | F1-SCORE |
|---|---|---|---|
| 99.27 | 94.31 | 94.4 | 94.10 |

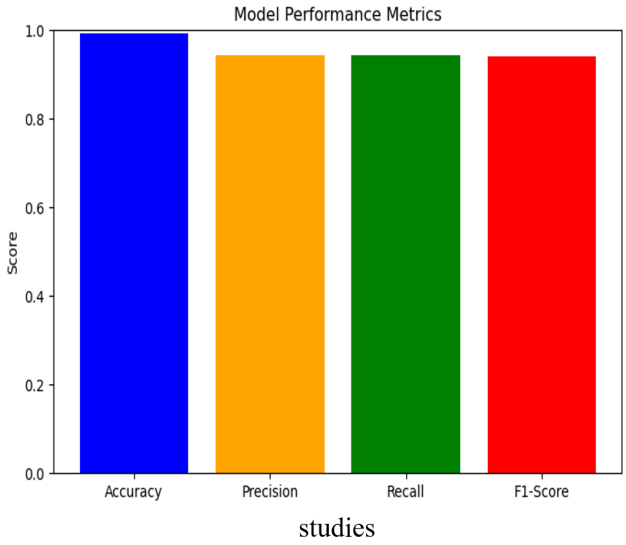


**Figure 3.** Results of the proposed model.

**Table 4.** Comparison of the results with other studies

| The study | ACCURACY | PRECISION | RECALL | F1-SCORE |
|---|---|---|---|---|
| The model | 99.27 | 94.31 | 94.4 | 94.10 |
| [5] | 97.8 | – | – | – |
| [11] | 98 | 98 | 96 | – |
| [14] | 98.3 | 96.3 | 46.0 | – |
| [15] | 98.72 | 97.45 | 97.92 | 98.23 |
| [18] | 99.8 | – | – | 96.25 |
| [19] | 94 | 99 | 93 | 97 |

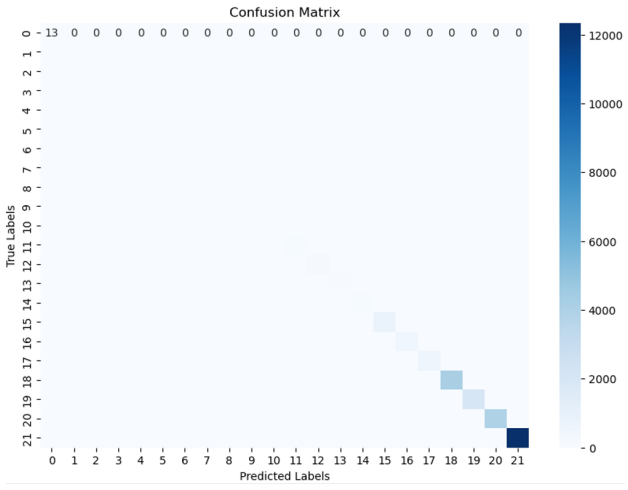


**Figure 4.** The proposed work confusion matrix

## 3. DISCUSSION

The results obtained from the experiments showed that the proposed system achieved high performance across all evaluation metrics. The accuracy was 99.27%, indicating that the system correctly classified most samples. This high level of accuracy reflects the efficiency of the

model in distinguishing between classes to detect attacks. Precision, which reached 94.31%, indicates that a high percentage of the items classified as positive were correct. This implies that the model can reduce the number of false positives, making it useful for attack detection. The recall was 94.4%, indicating that the system can identify most of the actual positive items, reducing the number of false negatives. This is important for attack-detection systems. Most of the false negatives were dominated by low-frequency or minority attack classes, such as the U2R and R2L classes, where most benchmark NSL-KDD datasets normally did not have enough representative samples. Therefore, classes with abundant training samples, such as DoS and Probe classes with more salient representative feature patterns, reached higher true-positive rates.

Most were due to the feature overlap between normal and probing traffic, which showed that benign packets shared similar statistical patterns with lightweight reconnaissance flows. PCA helped reduce many redundant correlations, but attacks with subtle signatures were not easily separated completely.

Finally, the F1-Score achieved by the proposed model was 94.10%, showing that the model offers a good balance between reducing false positives and increasing the number of true positives detected. Overall, these results indicate that the proposed system is effective in attack detection.

Through the comparison shown in Table 3, Study [5] only presented the accuracy performance measure, and its result was 97%, indicating that the proposed model outperformed this study in terms of the accuracy of ID. In [11], good results were presented in the performance measures used, but they were relatively lower than the proposed model in terms of the accuracy measure, as the proposed model outperformed this Study in terms of performance. In [14], it is noted that the recall is very low, and therefore, this model suffers from a major problem in identifying real cases of intrusion. Therefore, the proposed model shows high superiority over this model. In [15], the results demonstrated superiority in all measures, with the proposed model outperforming the others in accuracy. In [18], only the accuracy measure was presented, and the rest of the criteria were not used, making the comparison incomplete. Finally, the study [19] presented a high specific accuracy, but the proposed model outperforms this model in the rest of the criteria, which makes the proposed system more balanced.

The proposed model is characterized by its ability to maintain a high performance compared to previous studies. Applying the Principal Component Analysis (PCA) algorithm reduced the number of features to 26, leading to lower computational complexity and improved model efficiency. This renders the proposed model suitable for practical applications that require fast response times.

## 4. CONCLUSION

With the complexity and increase in cyber-attacks at present, the need for an effective IDS has increased. This paper presented an ID model based on machine learning algorithms and feature selection to reduce the data dimensions and speed up the detection time of attacks. The results demonstrated the effectiveness of the proposed model, as the accuracy of the model was 99%. In future work, we will present models for ID using deep learning and different datasets. The proposed PCA–SVM model exhibits good accuracy on NSL-KDD. However, its actual deployment in real-world cloud infrastructure has several limitations.

- Distribution shifts in data: Real network traffic is more dynamic and imbalanced than benchmark datasets.
- Scalability constraints: First, PCA transformation and SVM classification require computational costs related to the size of incoming data streams.
- Adaptability: The nature of SVM is static; retraining should be conducted when new types of attacks appear.

To overcome these problems, in future work, we will implement the proposed model using real cloud computing data.

## REFERENCES


[1] M. A. Talukder, S. Sharmin, M. A. Uddin, M. M. Islam, and S. Aryal, "Mlstl-wsn: Machine learning-based intrusion detection using smotetomek in wsns," *Int. J. Inf. Secur.*, vol. 23, pp. 2139–2158, 2024.

[2] S. M. Othman, F. M. Ba-Alwi, N. T. Alsohybe, and A. Y. Al-Hashida, "Intrusion detection model using machine learning algorithm on big data environment," *J. Big Data*, vol. 5, pp. 1–12, 2018.

[3] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. A. Khan, "Performance analysis of machine learning algorithms in intrusion detection system: A review," *Procedia Comput. Sci.*, vol. 171, pp. 1251–1260, 2020.

[4] A. R. Muhammad, P. Sukarno, and A. A. Wardana, "Integrated security information and event management (siem) with intrusion detection system (ids) for live analysis based on machine learning," *Procedia Comput. Sci.*, vol. 217, pp. 1406–1415, 2023.

[5] R. Jayaraj, A. Pushpalatha, K. Sangeetha, T. Kamaleshwar, S. U. Shree, and D. Damodaran, "Intrusion detection based on phishing detection with machine learning," *Meas. Sensors*, vol. 31, p. 101 003, 2024.

[6] M. Sajid, K. R. Malik, A. Almogren, T. S. Malik, A. H. Khan, and J. Tanveer, "Enhancing intrusion detection: A hybrid machine and deep learning approach," *J. Cloud Comput.*, vol. 13, p. 123, 2024.

[7] A. Pakmehr, A. Aßmuth, N. Taheri, and A. Ghaffari, "Ddos attack detection techniques in iot networks: A survey," *Clust. Comput.*, vol. 27, pp. 14 637–14 668, 2024.

[8] S. M. Othman, A. Y. Al-Mutawkkil, and A. M. Alnashi, "Survey of intrusion detection techniques in cloud computing," *Sana'a Univ. J. Appl. Sci. Technol.*, vol. 2, pp. 363–374, 2024.

[9] S. Santhosh Kumar, M. Selvi, and A. Kannan, "A comprehensive survey on machine learning-based intrusion detection systems for secure communication in internet of things," *Comput. Intell. Neurosci.*, vol. 2023, p. 8 981 988, 2023.

[10] S. M. Othman, N. T. Alsohybe, F. M. Ba-Alwi, and A. T. Zahary, "Survey on intrusion detection system types," *Int. J. Cyber-Security Digit. Forensics*, vol. 7, pp. 444–463, 2018.

[11] A.-R. Al-Ghuwairi, Y. Sharrab, D. Al-Fraihat, M. AlElaimat, A. Alsarhan, and A. Algarni, "Intrusion detection in cloud computing based on time series anomalies utilizing machine learning," *J. Cloud Comput.*, vol. 12, p. 127, 2023.

[12] F. Nabi and X. Zhou, "Enhancing intrusion detection systems through dimensionality reduction: A comparative study of machine learning techniques for cyber security," *Cyber Secur. Appl.*, p. 100 033, 2024.

[13] R. Alshamy, M. Ghurab, S. Othman, and F. Alshami, "Intrusion detection model for imbalanced dataset using smote and random forest algorithm," in *International Conference on Advances in Cyber Security*, 2021, pp. 361–378.

[14] Z. Sun, G. An, Y. Yang, and Y. Liu, "Optimized machine learning enabled intrusion detection system for internet of medical things," *Frankl. Open*, vol. 6, p. 100 056, 2024.

[15] H. M. Saleh, H. Marouane, and A. Fakhfakh, "Stochastic gradient descent intrusions detection for wireless sensor network attack detection system using machine learning," *IEEE Access*, 2024.

[16] M. Ali, M. Shahroz, M. F. Mushtaq, S. Alfarhood, M. Safran, and I. Ashraf, "Hybrid machine learning model for efficient botnet attack detection in iot environment," *IEEE Access*, 2024.

[17] H. Attou, A. Guezzaz, S. Benkirane, M. Azrour, and Y. Farhaoui, "Cloud-based intrusion detection approach using machine learning techniques," *Big Data Min. Anal.*, vol. 6, pp. 311–320, 2023.

[18] G. Logeswari, S. Bose, and T. Anitha, "An intrusion detection system for sdn using machine learning," *Intell. Autom. & Soft Comput.*, vol. 35, pp. 867–880, 2023.

[19] M. Bakro, R. R. Kumar, M. Husain, Z. Ashraf, A. Ali, and S. I. Yaqoob, "Building a cloud-ids by hybrid bio-inspired feature selection algorithms along with random forest model," *IEEE Access*, 2024.

[20] O. Ahmed, "Enhancing intrusion detection in wireless sensor networks through machine learning techniques and context awareness integration," *Int. J. Math. Stat. Comput. Sci.*, vol. 2, pp. 244–258, 2024.

[21] Ü. Çavuşoğlu, "A new hybrid approach for intrusion detection using machine learning methods," *Appl. Intell.*, vol. 49, pp. 2735–2761, 2019.

[22] H. Alqahtani, I. H. Sarker, A. Kalim, S. M. M. Hossain, S. Ikhlaq, and S. Hossain, "Cyber intrusion detection using machine learning classification techniques," in *Computing Science, Communication and Security (COMS2 2020)*, 2020, pp. 121–131.

[23] Z. Fan, S. Sohail, F. Sabrina, and X. Gu, "Sampling-based machine learning models for intrusion detection in imbalanced dataset," *Electronics*, vol. 13, p. 1878, 2024.

[24] A. A. Alashhab, M. S. Zahid, B. Isyaku, A. A. Elnour, W. Nagmeldin, and A. Abdelmaboud, "Enhancing ddos attack detection and mitigation in sdn using an ensemble online machine learning model," *IEEE Access*, 2024.

[25] A. Raza, K. Munir, M. S. Almutairi, and R. Sehar, "Novel class probability features for optimizing network attack detection with machine learning," *IEEE Access*, 2023.

[26] J. Li, M. S. Othman, H. Chen, and L. M. Yusuf, "Optimizing iot intrusion detection system: Feature selection versus feature extraction in machine learning," *J. Big Data*, vol. 11, p. 36, 2024.

[27] S. T. Ikram and A. K. Cherukuri, "Improving accuracy of intrusion detection model using pca and optimized svm," *J. Comput. Inf. Technol.*, vol. 24, pp. 133–148, 2016.

[28] F. E. Heba, A. Darwish, A. E. Hassanien, and A. Abraham, "Principal components analysis and support vector machine based intrusion detection system," in *2010 10th International Conference on Intelligent Systems Design and Applications*, 2010, pp. 363–367.

[29] M. Ghurab, G. Gaphari, F. Alshami, R. Alshamy, and S. Othman, "A detailed analysis of benchmark datasets for network intrusion detection system," *Asian J. Res. Comput. Sci.*, vol. 7, pp. 14–33, 2021.

[30] M. Ghurab, R. Alshamy, and S. Othman, "Performance evaluation for attack detection in intrusion detection system," *Int. J. Sci. Res. Eng. Dev.*, vol. 4, 2021.

[31] P. Rana, I. Batra, A. Malik, I.-H. Ra, O.-S. Lee, and A. S. Hosen, "Efficacious novel intrusion detection system for cloud computing environment," *IEEE Access*, 2024.

[32] R. Kumar, P. Kumar, R. Tripathi, G. P. Gupta, S. Garg, and M. M. Hassan, "A distributed intrusion detection system to detect ddos attacks in blockchain-enabled iot network," *J. Parallel Distributed Comput.*, vol. 164, pp. 55–68, 2022.

[33] A. A. Shujaaddeen, F. M. Ba-Alwi, A. T. Zahary, and A. S. Alhegami, "A model for measuring the effect of splitting data method on the efficiency of machine learning models: A comparative study," in *2024 4th International Conference on Emerging Smart Technologies and Applications (eSmarTA)*, 2024, pp. 1–13.

[34] H. A. Alsaeedi and A. S. Alhegami, "An incremental interesting maximal frequent itemset mining based on fp-growth algorithm," *Complexity*, vol. 2022, p. 1 942 517, 2022.