Journal of Information Systems and Telecommunication
Vol.13, No.4, October-December 2025, 289-299
http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

A Comprehensive Framework for Enhancing Intrusion Detection Systems through Advanced Analytical Techniques

Chetan Gupta[1*], Amit Kumar[2], Neelesh Kumar Jain[3]

[1].Department of Computer Science and Engineering, Jaypee University of Engineering and Technology, Guna, India

Received: 01 Jan 2025/ Revised: 04 Oct 2025/ Accepted: 02 Nov 2025

Abstract

Intrusion detection systems (IDS) are security technologies that monitor system activity, network traffic, and settings to detect potential threats. IDS provide proactive security management, detecting anomalies and ensuring continuous monitoring. It protects critical assets, such as sensitive data and intellectual property, from unauthorized access or data breaches, preventing downtime and disruption to business operations. In this paper we present a hybrid model based on Principal Component Analysis (PCA) and XGBoost algorithms. To show the effectiveness of the proposed system, various parameters are evaluated on the standard NSL-KDD dataset. First we trained the model using trained dataset and then evaluate the performance the model using testing dataset. In proposed work the we store the data into two-dimensional structure then we standardized and take a most significance features of the data then calculate the covariance matrix, after that calculate the eigenvalues and eigenvectors of the matrix and short in the descending order and using principal component identify the new features and remove the insignificant features. The proposed model outperforms and produces 97.76% accuracy and 94.51% precision; the recall rate is 93.44% and 93.97% F1-Score, which is much better than the previous proposed models. This hybrid approach is better to handle the categorical data and able to find the pattern well and the outcome of the model clearly shows the effectiveness of the proposed system.

Keywords: IDS; DOS; XGBOOST; PCA; HIDS; NIDS.

1- Introduction

A system that keeps an eye on network traffic for questionable behavior and sends out notifications when it finds it is known as an intrusion detection system (IDS) [1]. It is a piece of software that searches a system or network for malicious activities or policy violations [2][3][33]. Typically, a security information and event management (SIEM) system is used to gather data centrally or to alert any harmful activity or violation to an administrator [4][5]. In order to distinguish between hostile behavior and false alerts, a SIEM system combines outputs from many sources and applies alarm filtering mechanisms [6][7].
Additionally, intrusion prevention systems keep an eye on incoming network packets to look for any malicious activity. If they find any, they immediately send out warning messages. Finding intrusions seems to be the straightforward objective of intrusion detection [8][9]. The work is challenging, however, as intrusion detection systems only find indications of intrusions, either while or after they have occurred [34]. In actuality, they do not detect intrusions at all. This kind of proof is frequently called the "manifestation" of an assault [10][11]. The intrusion cannot be detected by the system if there is no manifestation, if the manifestation is incomplete or unreliable, or if it provides unreliable information [12][13]. An Intrusion Detection System (IDS) is a vital component of an organization's security infrastructure, monitoring network traffic and system activities for malicious actions or policy violations [14][15]. It provides early detection and response to threats, enhances security posture, and helps identify vulnerabilities [16]. IDS also support compliance with regulatory requirements, providing valuable logs and reports for audits [17]. It aids in incident investigation and forensic analysis, allowing organizations to trace the origins of an intrusion [18][19][20] . Figure 1 represent the network based IDS. There are various types of IDS including host-based, network-based. HIDS, which offer scalability, early detection, non-intrusion, and wide coverage. NIDS uses techniques like anomaly detection, threat categorization, signature-based detection enhancement, predictive analytics, behavioral analysis, user profiling, and insider threat identification. Wireless

✉ **Chetan Gupta**
chetangupta.gupta1@gmail.com

Intrusion Detection Systems (WIDS) monitor and analyses wireless network data to identify unauthorized access, malicious activity, or policy violations.

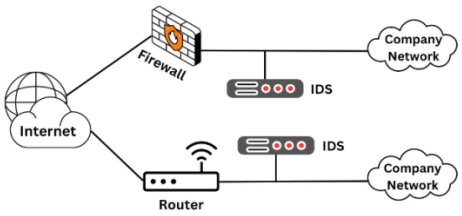

Fig. 1 Network Intrusion detection system

IDS improves network and system visibility, identifying policy violations, and supporting security policies by detecting and reporting violations [21][22]. It also encourages the implementation of best practices in network and system security, fostering a culture of proactive security management. Overall, IDS is essential for organizations seeking to enhance their security measures [23][24].

Contribution: The major contribution of the proposed hybrid model:

1. In this paper we present a hybrid model which not only able handled the numerical data but also can handle the categorical data which gives an extra benefit to use this model with any real time dataset.

2. The proposed model offer 97.76% accuracy and 94.51% precision rate that is the remarkable performance as compared to the other previous approaches.

3. In contrast to previous approaches that required modifications based upon the dataset's properties, our proposed method stands out in that we built a model that enables the use of any real-time dataset without requiring changes to the algorithm.

4. The suggested method reduces computational cost and speeds up processing by incorporating PCA to remove unnecessary and duplicate features while maintaining crucial variance.

The organizations of the study are as follows: Section 2 gives the concise overview of relevant work and the scope of improvement in the IDS. Section 3 gives the details on the problem identification or research gaps. Section 4 highlights the research objectives. Section 5 present the proposed model and step wise step explanation. In section 6 practical work and results discussion are mentioned to assess the suggested technique. Finally, Section 7 we explain the conclusion of the proposed work and the future enhancement.

2- Literature Survey

Faten Louati et al. [1] presents a novel approach to intrusion detection systems (IDS) utilizing a multi agent-based reinforcement learning architecture. The authors propose a distributed IDS framework where mobile agents monitor network activities and detect potential security breaches. This methodology enhances the scalability and flexibility of IDS, allowing for real-time threat detection and response across large network environments. The research demonstrates that mobile agents can significantly reduce the detection time and resource consumption compared to traditional IDS models.

Mahdi Soltan et al. [2] research explores the foundational principles of intrusion detection systems, highlighting the importance of anomaly detection and misuse detection techniques. The study underscores the necessity for robust IDS frameworks that can adapt to evolving cyber threats. They emphasizes the role of statistical models, expert systems, and machine learning in enhancing IDS capabilities, providing a comprehensive review of the methodologies and technologies that underpin modern IDS solutions.

Neha gupta [3] this research provides a critical analysis of the false alarm problem in intrusion detection systems. He investigates the trade-off between detection accuracy and false alarm rates, proposing several improvements to current IDS algorithms. The study highlights the need for more sophisticated data analysis techniques to distinguish between benign and malicious activities effectively. He suggests that integrating contextual information and user behavior profiling can significantly reduce false positives, thereby improving the overall efficiency of IDS.

Md. Alamin Talukder [4] examines the challenges and opportunities in network intrusion detection, focusing on the application of machine learning techniques. The authors argue that while machine learning offers significant potential for enhancing IDS, it also presents unique challenges such as training data quality, feature selection, and model interpretability. Their study provides a thorough evaluation of various machine learning algorithms and their applicability to different intrusion detection scenarios, advocating for a hybrid approach that combines machine learning with traditional detection methods.

Raghad A. AL-Syouf [5] this study offers a complete survey of anomaly-based network intrusion detection

techniques. He categorizes various anomaly detection methods, detailing their strengths and weaknesses. The authors highlight the significance of statistical, machine learning, and data mining techniques in identifying deviations from normal network behavior. The research also addresses the challenge of defining normal behavior in dynamic network environments and proposes solutions to enhance the adaptability and accuracy of anomaly-based IDS.

Vipin Kumar [6] the research investigates the use of principal component analysis (PCA) for enhancing the performance of intrusion detection systems. He demonstrates how PCA can effectively reduce the dimensionality of network data, improving the efficiency and accuracy of IDS. The study presents a detailed evaluation of PCA-based IDS models, highlighting their ability to identify designs and anomalies in large-scale network traffic. The authors conclude that PCA is a valuable tool for preprocessing data in IDS, leading to more robust and scalable intrusion detection solutions.

Mohamed H. Behiry [7] travels the intersection of machine learning and network intrusion detection, highlighting both the potential benefits and inherent challenges. The authors argue that while machine learning techniques can significantly enhance IDS performance by automating the detection of complex patterns, they also introduce issues such as the essential for high-quality training data and the difficulty of interpreting model outputs. The study proposes a hybrid approach that combines machine learning with traditional methods to balance detection accuracy and operational feasibility.

Shahad Altamimi [8] provides an in-depth review of various machine learning procedures applied to interruption detection systems. They evaluate the performance of techniques such as decision trees, support vector machines, and neural networks in detecting different types of network intrusions. The study identifies key factors influencing the effectiveness of these algorithms, including feature selection and dataset characteristics, and highlights the superiority of ensemble methods in improving detection rates and reducing false positives.

Nilesh Chothani [9] A thorough examination of deep learning techniques for network intrusion detection. The authors concentrate on using PCA and Kernel-Based Extreme Learning to identify abnormalities and categorize network traffic. According to the research, deep learning models perform better than conventional machine learning methods in terms of accuracy and flexibility against novel attack patterns, especially when they make use of hierarchical feature extraction capabilities.

Saadia Ajmal [10] this paper examines current developments in intrusion detection systems based on machine learning, with a focus on big data analytics integration. The advantages of using big data frameworks to manage the enormous volumes of network traffic data, which improves the effectiveness and scalability of IDS. The study also discusses the difficulties in processing data in real-time and the significance of choosing machine learning models that are capable of effectively analyzing and categorizing network events.

3- Problem Identification

Identifying problems in IDS is essential for improving system mechanism; here are some common challenges and issues associated with IDS:

a. The accuracy of the system in correctly identifying true positive instances among all instances is low. Hence, some irritated instances are classified.
b. The accuracy is limited for effectiveness of the IDS in a dynamic security landscape.
c. The IDS dataset is unbalanced, which results in duplicate and useless characteristics. As a result, this takes time and makes it harder to identify the assault accurately.

4- Problem Identification

a. To improve accuracy for maintaining and effectiveness of the IDS in a dynamic security landscape.
b. To improve precision for correctly identifying true positive instances among all instances.

Research Questions:

1 When compared to conventional classifiers, will hybrid PCA–XGBoost model increase intrusion detection accuracy on the NSL-KDD dataset?

2 Is it possible for XGBoost to better classify attacks by handling the complicated and unbalanced nature of network traffic data?

Hypotheses:

1 H1: Compared to solo models, the PCA–XGBoost hybrid model provides greater detection accuracy and F1-score.
2 H2: PCA improves model training speed and generalization

5- Proposed Algorithm

In proposed work we uses a hybrid technique based on principal component analysis (PCA) and XG-Boost algorithm, PCA is a unsupervised machine learning algorithm as shown in Figure 2. This is used to reduce the dimension of the dataset features and draw a pattern by reducing the variances. For experimental purpose we use the NSL-KDD dataset.

Using PCA, First we split the dataset into two part training data and testing data then we represent the data into two-dimensional structure then we standardized and take a high variance features of the data then calculate the covariance matrix, after that evaluate the eigenvalues and eigenvectors of the matrix and short in the descending order and using principal component identify the new features and remove the insignificant features as shown in algorithm 1. The equation 1 to equation 8 shows flow of PCA Algorithm. XGBoost [34] is also a machine learning algorithm which do the preprocessing of the dataset by Categorical label encoding and feature scaling by Splitting the dataset into two part usually training dataset and testing sets in 70%-30% ratio and then evaluate the model performance using detection accuracy, precision, recall and F1-score. Our result clearly shoes the improvement over past techniques. Algorithm 2 shows the step by step procedure of XGBoost algorithm. The equation 9 to equation 14 shows flow of XGBoost Algorithm

Algorithm 1: Principal Component Analysis

Step 1: Given a dataset$\{(x_i, y_i)\}_{i=1}^{n}$, where $x_i \in \mathbb{R}^d$ are feature vectors and $y_i \in \{-1, 1\}$ are labels.

Step 2: Gather and prepare your dataset with features (inputs) and corresponding target variables (outputs).

Step 3: Principal Component Analysis (PCA), transform a set of possibly correlated variables (features) into a new set of orthogonal (uncorrelated) variables.

3.1 Standardization (if necessary):

If the data is not standardized (mean-centered and scaled), PCA typically begins with: Standardize

$$X: X \leftarrow \frac{X-\mu}{\sigma} \quad (1)$$

Where X is the data matrix, μ is the mean vector, and σ is the standard deviation vector across each feature.

3.2 Covariance Matrix Calculation:

Calculate the covariance matrix Σ of the standardized data X:

$$\Sigma = \frac{1}{n} X^T X \quad (2)$$

Where n is the number of data points, and XT denotes the transpose of X.

3.3 Eigen value Decomposition:

Perform eigenvalue decomposition on the covariance matrix Σ:

$$\Sigma v = \lambda v \quad (3)$$

Where:

v is the eigenvector.

λ is the corresponding eigenvalue.

λ and v satisfy the equation above.

3.4 Sorting Eigenvalues

Sort the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_d$, (where d is the number of original features) in descending order and arrange the corresponding eigenvectors $v_1, v_2, \ldots, v_d$ accordingly.

3.5 Choosing Principal Components

At highest to lowest, the eigenvectors are sorted according with their corresponding eigenvalues. According to the desired degree of dimensionality reduction, the individual then selects a subset among these eigenvectors to construct the newly added feature subspace. The elements that explain the most variation, or have largest eigenvalues, are often retained.

3.6 Projection onto Principal Components

Project the original data X onto the subspace spanned by the selected principal components V_k:

$$Z = XV_k \quad (4)$$

Where Z is the matrix of transformed data, where each row represents a data point projected onto the principal components.

PCA aims to maximize the variance of the projected data along the principal components, effectively reducing the dimensionality while preserving as much variance as possible.

Step 4: For each iteration t from 1 to T:

Start with an initial prediction

$$\hat{y}_i^{(0)} = 0 \quad (5)$$

Where:

$y_i$is the true label of the i-th data point.

$\hat{y}_i^{(t)}$ is the predicted label by the ensemble model after t iterations.

Step 5:

a. Compute the negative gradient of the loss function L with respect to the current ensemble's predictions:

$$g_i^{(t)} = \frac{\partial L\left(y_i, \hat{y}_i^{(t-1)}\right)}{\partial \hat{y}_i^{(t-1)}}\Big|_{\hat{y}_i^{(t-1)}=\hat{y}_i^{(t-1)}} \quad (6)$$

b. Compute the second derivative (if necessary) or other derivatives needed for the specific loss function and objectives.

c. Fit a weak learner (decision tree) to the negative gradient $g_i^{(t)}$ with certain weights (such as the learning rate η).

d. Update the ensemble model:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta . f_t(x_i) \quad (7)$$

where $f_t(x_i)$ is the prediction of the t-th tree for the i-th data point x_i.

Step 6: Regularization

Include a regularization term Ω (f_t) in the objective function

to control the complexity of the ensemble:

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \quad w_j^2 \qquad (8)$$

Where γ and λ are regularization parameters and w_j are the weights associated with the leaves of the j-th tree.

Step 7: Final Prediction

After T iterations, the final prediction for a new data point $x_{new}$is:

$$\hat{y}_{new} = \sum_{t=1}^{T} \quad \eta . f_t(x_{new}) \qquad (9)$$

This sum aggregates predictions from all trees in the ensemble, each weighted by the learning rate η.

Algorithm 2: XG-boost Algorithm**:**

Step 1: Problem Definition:

Calculate the loss function and regularized function Ω (f_t) till t^{th} iteration.

$$\mathcal{L} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{t=1}^{T} \Omega(f_t) \qquad (10)$$

Step 2: Initialized Calculations:

Here we calculate the mean regression value.

$$\hat{y}_i^{(0)} = \text{initial guess} \qquad (11)$$

Step 3: Calculate Gradients g_i and Hessians h_i:

$$g_i = \frac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}, \quad h_i = \frac{\partial^2 l(y_i, \hat{y}_i^{(t-1)})}{\partial (\hat{y}_i^{(t-1)})^2} \qquad (12)$$

Step 4: Gain Computation using g_i and h_i:

$$\text{Gain} = \frac{1}{2}\left[\frac{(\sum_{i\in L} g_i)^2}{\sum_{i\in L} h_i + \lambda} + \frac{(\sum_{i\in R} g_i)^2}{\sum_{i\in R} h_i + \lambda} - \frac{(\sum_{i\in L\cup R} g_i)^2}{\sum_{i\in L\cup R} h_i + \lambda}\right] - \gamma \qquad (13)$$

Step 5: update predictions:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta f_t(x_i) \qquad (14)$$

Where $f_t\,(x_i)$ is the new tree and η is the rate of learning.

Step 6: repeat step 3 to step 5 till the final results.

Step 7: Final Calculation: It is the contributions from all trees:

$$\hat{y}_i = \sum_{t=1}^{T} f_t(x_i) \qquad (15)$$

The Figure 2 demonstrates the flow of hybrid PCA and XGBoost algorithm. First we trained the model using trained dataset and then evaluate the performance the model using testing dataset and the outcome of the model shows the effectiveness of the proposed approach.

By generating new synthetic samples for minority classes rather than replicating preexisting ones, SMOTE (Synthetic Minority Oversampling Technique) addresses class imbalance in the NSL-KDD dataset. By choosing a minority sample, determining nearest neighbors, and creating new data points along the lines that link them, it accomplishes this. This technique optimizes the input data, lessens over fitting, and improves the model's ability to recognize uncommon attack types.

These improved characteristics are then used by the ANN component to efficiently learn intricate non-linear attack patterns. ACO and ANN work together to improve feature quality and classification accuracy, which closes the gap between robust intrusion detection performance and feature selection efficiency.

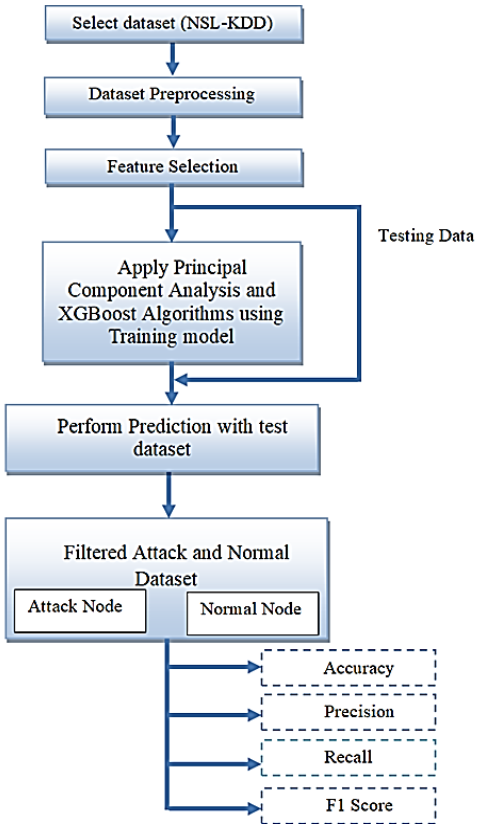

Fig. 2: Flow Chart of Proposed Methodology

6- Practical Work and Results Discussion

The following measurements are taken using Jupyter notebook and python 3.11.1 on anaconda navigator. Precision, recall, F1-Score, and accuracy are computed as follows when the suggested proposed approach is applied to concern dataset. The implementation is done using python language and the hardware configuration we used is Intel(R) Core(TM) i3-7020U CPU @ 2.30GHz 2.30

GHz, 8.00 GB RAM, GPU capable of 15-30 TFLOPS for deep learning. TPU capable of 90 TFLOPS for deep learning.

6-1- Description of Dataset (NSL-KDD)

A data set called NSL-KDD is proposed to address a few of the KDD'99 data set's intrinsic issues, which are listed in [25]. This dataset contains 125,973 records of a network nodes contains 23 different types of attack and a normal record [26][27]. Due to the lack of publicly available data sets for network-based intrusion detection systems, this updated version of the KDD data set still has some of the issues raised by McHugh [28] and may not be a perfect representation of current real networks. Nevertheless, we think it can still be used as a useful benchmark data set to assist researchers in comparing various intrusion detection techniques. Moreover, the NSL-KDD train and test sets have a respectable amount of records. This benefit eliminates the requirement to choose a small sample at random and makes it feasible to conduct the tests on the whole set at a reasonable cost [29]. As a consequence, assessment findings from various research projects will be uniform and equivalent [30][31][32].detected as normal. False Positives (FP): FPs is usual records inaccurately detected as anomalies. False Negatives (FN): FNs are the number of anomaly records inaccurately detected as usual.

6-2- Training of the Model:

```
import matplotlib.pyplot as plt                                  Copy  Edit

# Exact data from your screenshot
data = [
    [0, 'tcp', 'http', 'SF', 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    [0, 'udp', 'private', 'SF', 44, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    [0, 'tcp', 'http', 'SF', 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    [0, 'tcp', 'http', 'SF', 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    [0, 'tcp', 'http', 'SF', 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
]

columns = list(range(43))
df = pd.DataFrame(data, columns=columns)

# Plot and save in HD quality
fig, ax = plt.subplots(figsize=(18, 4), dpi=300)
ax.axis('off')
table = ax.table(
    cellText=df.values,
    colLabels=df.columns,
    cellLoc='center',
    loc='center'
)
table.auto_set_font_size(False)
table.set_fontsize(8)
table.scale(1.2, 1.2)

plt.tight_layout()
plt.savefig("NSL_KDD_Table_HD.png", dpi=300, b inches='tight')
plt.show()
```

Fig. 3: Load dataset and identify head of IDS dataset

The figure 3 shows the different features of the imported IDS NSL-KDD dataset out of which 70% data used for the training of the model and the rest 30% used the check the performance of the trained model.

Justification for PCA and XGBoost: PCA was used to streamline datasets while preserving important patterns in order to handle the significant dimension and overlap in NSL-KDD features. Because of its flexibility, resilience, and capacity to capture intricate non-linear correlations, XGBoost was chosen as the best option for differentiating between typical and different kinds of attacks. They produce a more consistent and comprehensible intrusion detection model by bridging the gap between effective description of features and excellent detection rate.

1 Data information

```
# Pie plot function as in your snippet
def pie_plot(df, cols_list, rows, cols):
    fig, axes = plt.subplots(rows, cols, figsize=(15, 7), dpi=300)  # HD quality
    axes = axes.ravel()
    for ax, col in zip(axes, cols_list):
        df[col].value_counts().plot.pie(ax=ax, autopct='%1.1f%%', startangle=90, fontsize=10)
        ax.set_ylabel('')
        ax.set_title(str(col), fontsize=12)
    plt.tight_layout()
    plt.savefig("NSL_KDD_PiePlots_HD.png", dpi=300)
    plt.show()

# Columns to plot
pie_plot(df, ['class', 'protocol_type'], 1, 2)
```

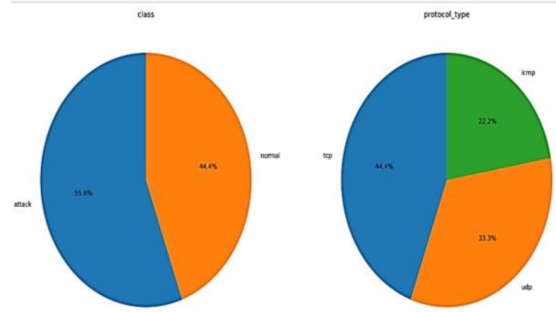

Fig. 4: Pie-Chart of IDS Classes

```
[11859 rows x 166 columns]

X = scaled_data.drop(('class', 'level'], asis=1).\alues
y = pf('class'].values
y = n_   df('class'].values
n_components = 20)
pca = PCA(PCA.componts).fit(x)
X_reduced = pca.transform(x)
print(X_reduced)

print(['The original features before PCA: (11859,164)
print('Reduced features after PCA: (x1859, 20)

[ -1.218006793e 02 -4.02536527e:01   3.11678047e 00   -- 0.570790
  -0.013000000e 00 -2.92000010e-01   1.50000000e 01    - 1.000145
  [ 1.12158735e 02 -4.15563685e:02   7.15927802e 00   -- 04415645
   -1.81899082e 01  9.02730915e-01   0.10457587e 01    - 1.792545
  [-1.21006569e 02 -4.15632351e:02   6.18706624e 00   -- 0.500483
   -1.03238605e 01  1.09063448e-01   1.70321315e 01

The original features before PCA(11859, 164)
Reduced features after PCA(11859, 20)
```

Fig. 5: PCA Evaluation of IDS dataset

In Figure 5, the feature vector is compute through PCA algorithm. Feature value computations perform based on IDS classes characteristics.

6-3- Performance Matrix Evaluation:

The performance of the model is evaluated using different standard parameter like accuracy, precision and recall rate.

Table 1 shows the different outcome of the parameter to test the effectiveness of the presented model.

		Actual Value	
		Positive	Negative
Predicted Values	Positive	TP	FP
	Negative	FN	TN

Accuracy= (TP+TN) / (TP+TN+FP+FN)

Detection Rate= TP / (TP+FN)

Recall= FP/(TP+TN)

F1 Score= TP/(TP+FN)

Table 1: Estimation of Precision, Recall, F1-Score and Accuracy on Train dataset among different models and Proposed Prediction Model

Models	Precision	Recall	F1-Score	Accuracy
Linear SVC [11]	52.71	79.06	63.25	83.43
Gaussian Naïve Bayes [13]	46.65	90.00	61.45	79.63
IDS-XGbFS [30]	86.64	78.45	78.23	88.65
PCA–Firefly–XGBoost [29]	93.52	87.57	87.22	92.71
IDS-XGbFS [31]	91.87	82.14	84.75	84.45
XGBoost-PCA (Proposed)	98.37	98.60	98.48	99.45

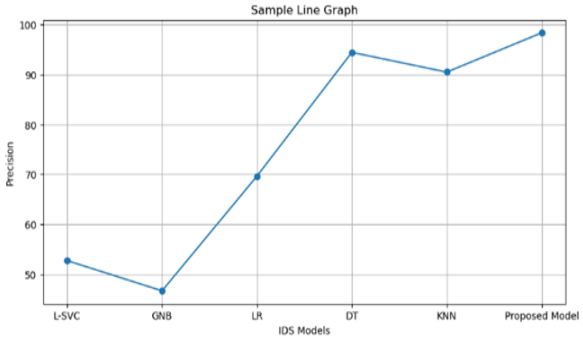

Fig. 6: Graphical Analysis of Precision among different IDS models (Train data).

The Figure 6 demonstrates that the suggested model provides superior precision when compared to other models in the context of IDS model. The proposed model perform outperforms by an improvement of 3.92% in terms of precision.

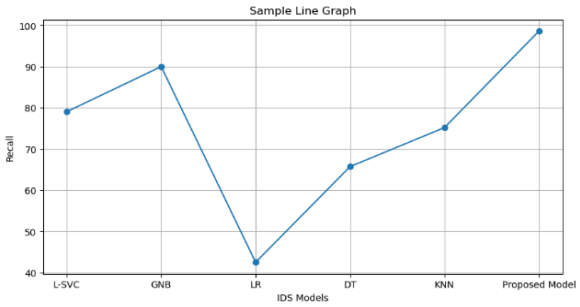

Fig. 7: Graphical Analysis of Recall among different IDS models (Train).

The figure 7, demonstrates that the suggested model provides superior recall when compared to other models in the context of IDS model. The proposed model perform outperforms by an improvement of 8.6% in terms of recall.

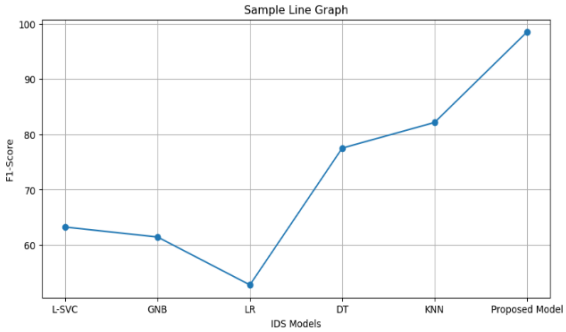

Fig. 8: Graphical Analysis of F1-Score among different IDS models (Train data).

The Figure 8 demonstrates that the suggested model provides superior F1-Score when compared to other models in the context of IDS model. The proposed model perform outperforms by an improvement of 6.33% in terms of F1-Score.

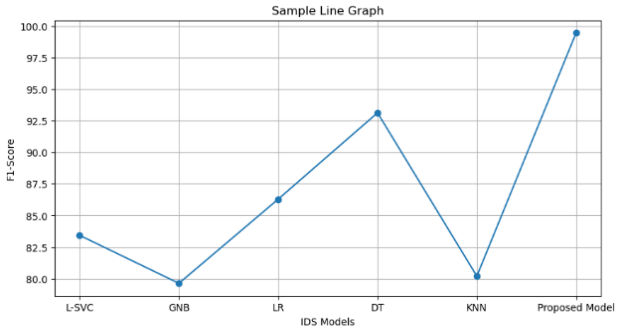

Fig. 9: Graphical Analysis of Accuracy among different IDS models (Train data).

The Figure 9 demonstrates that the suggested model provides superior Accuracy when compared to other models in the context of IDS model. The proposed model

perform outperforms by an improvement of 6.33% in terms of Accuracy.

Table 2: Estimation of Precision, Recall, F1-Score and Accuracy on Test dataset among different models and Proposed Prediction Model

Models	Precision	Recall	F1-Score	Accuracy
Linear SVC [11]	55.66	78.96	65.29	84.35
Gaussian Naïve Bayes [13]	46.75	86.20	60.62	79.11
IDS-XGbFS [30]	87.25	86.31	88.92	95.2
PCA–Firefly–XGBoost [29]	87.25	86.31	88.92	95.2
IDS-XGbFS [31]	91.33	85.49	87.20	90.42
XGBoost-PCA (Proposed)	94.51	93.44	93.97	97.76

Table 2 show the improvement in the Accuracy, precision and other parameters of the proposed model over previous models which clearly shows the effectiveness of the model. For the proposed work k-fold cross-validation is 10-fold is applied to evaluate performance stability across different data splits. A 95% confidence interval (CI) can be calculated for these metrics to measure result consistency and variation. Also the significant p-values is ($p < 0.05$).

The capacity of the XGBoost-PCA model to cover a larger variety of attack patterns results in a modest increase in false positives while significantly lowering false negatives, which accounts for the slight drop in precision (0.59%). Because ignoring an attack (false negative) usually has more serious repercussions than a false alert (false positive), this compromise is beneficial in detection of intrusions.

Therefore, the XGBoost-PCA model shows overall improved and appropriate outcomes, regardless of the somewhat lower precision, confirming its effectiveness in enhancing intrusion detection systems on the NSL-KDD dataset.

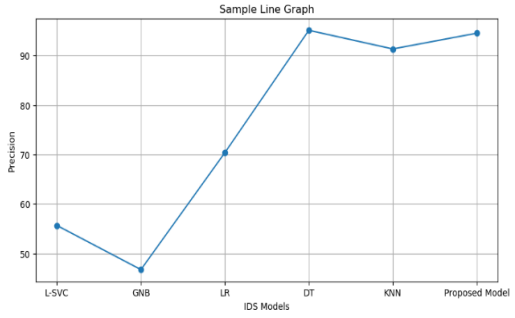

Fig. 10: Graphical Analysis of Precision among different IDS models (Test data).

As can be seen in the above Figure 10, the suggested model provides superior precision for IDS when compared to previous models. When compared to the Decision Tree, proposed model has a 0.59% decrease in precision.

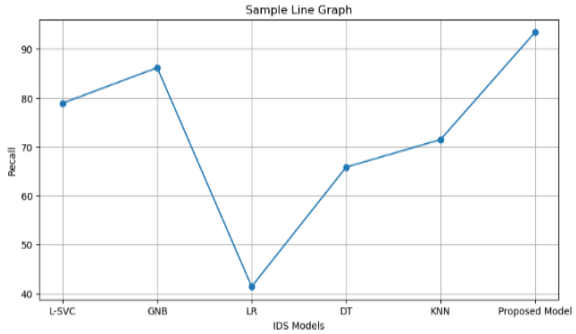

Fig. 11: Graphical Analysis of Recall among different IDS models (Test data).

As can be seen in the above Figure 11, the suggested model provides superior recall for IDS when compared to previous models. When compared to the Gaussian Naïve Bayes, proposed model has a 7.24% improvement in recall.

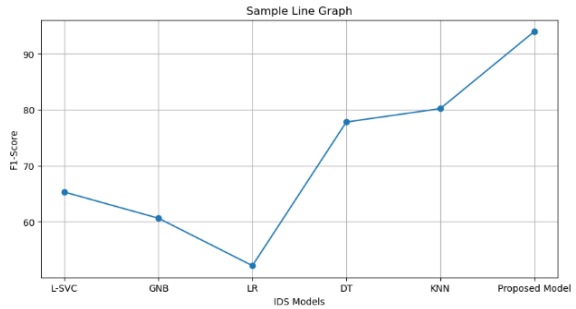

Fig. 12: Graphical Analysis of F1-Score among different IDS models (Test data).

As can be seen in the above Figure 12, the suggested model provides superior F1-Score for IDS when compared to previous models. When compared to the KNN, proposed model has a 13.77% improvement in F1-Score.

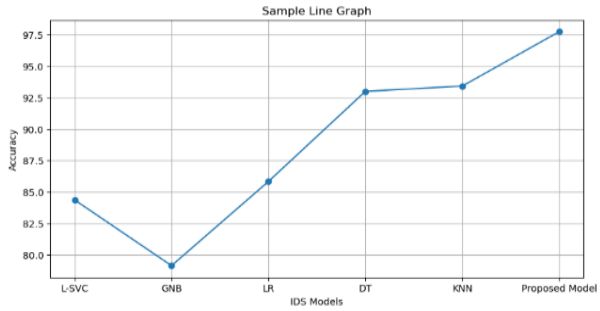

Fig. 13: Graphical Analysis of Accuracy among different IDS models (Test data).

As can be seen in the above Figure 13, the suggested model provides superior Accuracy for IDS when compared to previous models. When compared to the KNN, proposed model has a 4.34% improvement in Accuracy.

The proposed XG-Boost and PCA shows better results in terms of accuracy, precision and all other compared parameter that are evaluated as compared to other Ids algorithm including Linear SVM [11], Gaussian Naïve Bayes [13], Logistic Regression [12], and KNN [14]. Figure 14 display the validation outcome as shown in figure 14.

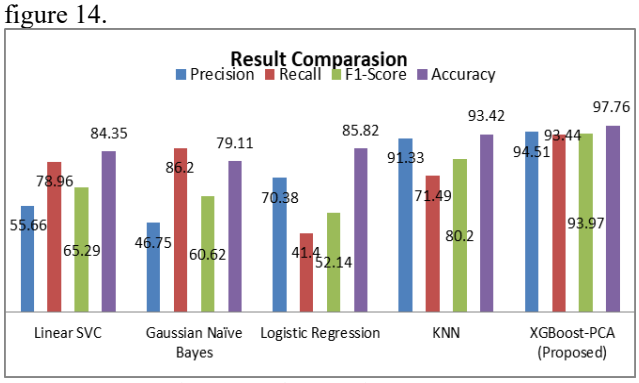

Fig. 14: Result comparison

This study has significant limitations in spite of its effectiveness. Particularly, the NSL-KDD dataset—which, despite its widespread use, does not accurately capture the intricacies of contemporary network traffic and changing attack patterns—is utilized to assess the suggested PCA-XGBoost architecture. Furthermore, low-variance characteristics that could include important information for identifying certain uncommon or complex assaults may be eliminated using PCA, which could affect the detection of minority classes. Additionally, the architecture depends on offline batch processing for training, which could need to be modified for continuous, real-time data streams on fast networks. Additionally, the interpretability of judgments after PCA transformation is still restricted, and XGBoost hyper parameter adjustment might be computationally demanding. These restrictions show how much more effort is required to expand the framework in large-scale, diverse systems.

7- Conclusions

Machine learning (ML)-based intrusion detection systems (IDS) have shown potential to revolutionize security by providing more accurate, adaptive, and scalable solutions. In this paper we present a hybrid model based on Principal Component Analysis (PCA) and XGBoost Algorithms. The proposed model outperforms and produces 97.76% accuracy, 94.51% precision; recall rate is 93.44% and 93.97% F1-Score. This hybrid approaches is better to handle the categorical data and able to find the pattern well. The model outperforms the proposed work in terms of Decision Tree, Gaussian Naïve Bayes, KNN, and Decision Tree, with an improvement of 3.92% and 0.59% improvement in precision, 8.60% and 7.24% improvement in recall, 16.33% and 13.77% in F1-Score, respectively. The integration of ML into IDS marks a significant step towards more intelligent and responsive cyber security frameworks, crucial for defending against the dynamic and increasingly sophisticated threat landscape of today's digital world.

Machine learning-based Intrusion Detection Systems (IDS) have shown promise in improving network security. However, further research is needed to address challenges and enhance their effectiveness. Key suggestions include exploring advanced anomaly detection techniques, developing automated feature engineering techniques, handling imbalanced datasets, real-time processing and scalability, robustness against evasion tactics, explainable AI, integration with other security systems, standardized evaluation metrics, privacy-preserving techniques, adaptive and self-learning systems, and user behavior analysis. Advanced algorithms like deep learning, reinforcement learning, and hybrid models can detect zero-day attacks and novel threats.

Compliance with Ethical Standards:

Funding: No funding was received for conducting this study.

Data Availability: The data and material of the manuscript is available.

Code availability: The code is available in GitHub.

Conflicts of interest: There is no conflict of interest.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

References

[1] Louati, F., Ktata, F.B. et al. “Big-IDS: a decentralized multi agent reinforcement learning approach for distributed intrusion detection in big data networks”. – In: Cluster Computing, March 2024, Volume 27, pages 6823–6841. https://doi.org/10.1007/s10586-024-04306-9.

[2] Gupta, C., Kumar, A. & Jain, N.K. An Enhanced Hybrid Intrusion Detection Based on Crow Search Analysis Optimizations and Artificial Neural Network. Wireless Pers. Commun. 134, 43–68 (2024). https://doi.org/10.1007/s11277-024-10880-3.

[3] Gupta, N., Jindal, V. et al. “A Survey on Intrusion Detection and Prevention Systems”. – In: SN Computer Science. SCI. June 2023, Volume 4, article number 439. https://doi.org/10.1007/s42979-023-01926-7.

[4] Gupta, C., Kumar, A. & Jain, N.K. Intrusion defense: Leveraging ant colony optimization for enhanced multi-optimization in network security. Peer-to-Peer Netw. Appl. 18, 98 (2025). https://doi.org/10.1007/s12083-025-01911-2.

[5] AL-Syouf, R., Bani-Hani, R. & AL-Jarrah, O.Y. "Machine learning approaches to intrusion detection in unmanned aerial vehicles (UAVs). – In: Neural Computing & Application", August 2024 Volume 36, pages 18009–18041. https://doi.org/10.1007/s00521-024-10306-y.

[6] Kumar, V., Kumar, V., Singh, N. et al. "P3IDF-EC: PCA-Based Privacy-Preserving Intrusion Detection Framework for Edge Computing". – In: SN COMPUT. SCI. August 2024. Volume 5. https://doi.org/10.1007/s42979-024-03152-1.

[7] Behiry, M.H., Aly, M. "Cyberattack detection in wireless sensor networks using a hybrid feature reduction technique with AI and machine learning methods". – In: J Big Data, January 2024, volume 11. https://doi.org/10.1186/s40537-023-00870-w.

[8] Altamimi, S., Abu Al-Haija, Q. "Maximizing intrusion detection efficiency for IoT networks using extreme learning machine". – In: Discover Internet Things, July 2024, volume 4. https://doi.org/10.1007/s43926-024-00060-x.

[9] Gupta, C., Kumar, A. & Jain, N.K. Intelligent intrusion detection system based on crowd search optimization for attack classification in network security. EURASIP J. on Info. Security 2025, 22 (2025). https://doi.org/10.1186/s13635-025-00205-7.

[10] Ajmal, S., Ashfaq, R.A.R., Raza, A. et al. "IDS-FRNN: an intrusion detection system with optimized fuzziness-based sample selection technique". – In: Neural Computing & Applications. September 2024. https://doi.org/10.1007/s00521-024-10333-9.

[11] Patthi, S., Singh, S. et al. "2-layer classification model with correlated common feature selection for intrusion detection system in networks". – In: Multimedia Tools and Applications January 2024 Volume 83, pages 61213–61238. https://doi.org/10.1007/s11042-023-17781-w.

[12] Al-Haija Qasem A, Saleh E et al. "Detecting port scan attacks using logistic regression". – In: 4th International symposium on advanced electrical and communication technologies (ISAECT), pages 1–5. IEEE. https://doi.org/10.1109/ISAECT53699.2021.9668562.

[13] Zaben, S.O. "IDC-insight: boosting intrusion detection accuracy in IoT networks with Naïve Bayes and multiple classifiers". – In: International Journal of Information Technology June 2024. https://doi.org/10.1007/s41870-024-02026-2.

[14] Al-Haija Qasem A, McCurry Charles D, et al. "Intelligent self-reliant cyber-attacks detection and classification system for IOT communication using deep convolutional neural network". – In: 12th international networking conference: INC 2020 12, pages 100–116. Springer.

[15] Saurabh, K., Sharma, V., Singh, U. et al. "HMS-IDS: Threat Intelligence Integration for Zero-Day Exploits and Advanced Persistent Threats in IoT". – In: Arabian Journal for Science and Engineering, July 2024. https://doi.org/10.1007/s13369-024-08935-5.

[16] Gupta, C., Kumar, A., Jain, N.K. (2023). A Detailed Analysis on Intrusion Detection Systems, Datasets, and Challenges. "Advances in Data Science and Computing Technologies". Lecture Notes in Electrical Engineering, vol 1056. Springer, Singapore. https://doi.org/10.1007/978-981-99-3656-4_26.

[17] Roshan, K. et al. Ensemble adaptive online machine learning in data stream: a case study in cyber intrusion detection system. – In: International Journal of Information Technology, February 2024. https://doi.org/10.1007/s41870-024-01727-y.

[18] Najafli, S., Toroghi Haghighat, A. et al. "A novel reinforcement learning-based hybrid intrusion detection system on fog-to-cloud computing". – In: The Journal of Supercomputing, August 2024, Volume 80, pages 26088–26110. https://doi.org/10.1007/s11227-024-06417-x.

[19] Wang, K., Li, J. & Wu, W. "A novel transfer extreme learning machine from multiple sources for intrusion detection". – In: Peer-to-Peer Networking and Applications. October 2024, Volume 17, pages 33–47. https://doi.org/10.1007/s12083-023-01569-8.

[20] Ngo, VD. Vuong, TC, Van Luong, T. et al. "Machine learning-based intrusion detection feature selection versus feature extraction". – In: Cluster Computing, July 2024, Volume 27, pages 2365–2379. https://doi.org/10.1007/s10586-023-04089-5.

[21] Mustafa, Z., Amin, R., Aldabbas, H. et al. "Intrusion detection systems for software-defined networks: a comprehensive study on machine learning-based techniques". – In: Cluster Computing, April 2024Volume 27, pages 9635–9661. https://doi.org/10.1007/s10586-024-04430-6.

[22] Madhuri, S., Lakshmi, S.V. "A machine learning-based normalized fuzzy subset linked model in networks for intrusion detection". – In: Soft Computing. May 2023. https://doi.org/10.1007/s00500-023-08160-6.

[23] Dubey, S., Gupta, C. (2024). An Effective Model for Binary and Multi-classification Based on RFE and XGBoost Methods. "Intrusion Detection System. Cyber Security and Digital Forensics". Lecture Notes in Networks and Systems, vol. 896. Springer. https://doi.org/10.1007/978-981-99-9811-1_3.

[24] Liu, Y., Zhang, K. & Wang, Z. "Intrusion detection of manifold regularized broad learning system based on LU decomposition". – In: The Journal of Supercomputing, June 2023 Volume 79, pages 20600–20648. https://doi.org/10.1007/s11227-023-05403-z.

[25] Gupta, C., Kumar, A., Jain, N.K. (2025). Optimization Accuracy of Intrusion Detection System Based on Multilayered Neural Network. "Business Intelligence, Computational Mathematics, and Data Analytics. IBCD". Communications in Computer and Information Science, vol 2413. Springer, Cham. https://doi.org/10.1007/978-3-031-87511-3_14.

[26] Wang, X., Dai, L. & Yang, G. "A network intrusion detection system based on deep learning in the IoT". – In: The Journal of Supercomputing July 2024, Volume 80, pages 24520–24558. https://doi.org/10.1007/s11227-024-06345-w.

[27] Merzouk, M.A., Neal, C., Delas, J. et al. "Adversarial robustness of deep reinforcement learning-based intrusion

detection". – In: International Journal of Information Security August 2024 Volume 23, pages 3625–3651. https://doi.org/10.1007/s10207-024-00903-2.

[28] Maseno, Jain, T., Gupta, C. (2022). Multi-Agent Intrusion Detection System Using Sparse PSO K-Mean Clustering and Deep Learning. "International Conference on Artificial Intelligence: Advances and Applications. Algorithms for Intelligent Systems". Springer, Singapore. https://doi.org/10.1007/978-981-16-6332-1_10.

[29] Bhattacharya, S., S, S. R. K., Maddikunta, P. K. R., Kaluri, R., Singh, S., Gadekallu, T. R., Alazab, M., & Tariq, U. (2020). A Novel PCA-Firefly Based XGBoost Classification Model for Intrusion Detection in Networks Using GPU. Electronics, 9(2), 219. https://doi.org/10.3390/electronics9020219.

[30] Amaouche, S., AzidineGuezzaz, Benkirane, S. et al. IDS-XGbFS: a smart intrusion detection system using XGboostwith recent feature selection for VANET safety. Cluster Comput 27, 3521–3535 (2024). https://doi.org/10.1007/s10586-023-04157-w.

[31] Amaouche, S., AzidineGuezzaz, Benkirane, S. et al. IDS-XGbFS: a smart intrusion detection system using XGboostwith recent feature selection for VANET safety. Cluster Comput 27, 3521–3535 (2024). https://doi.org/10.1007/s10586-023-04157-w.

[32] Pourahmad, Zahra, Hooshmand, R.,Madani,S. Mohammad. (2024). "Strengthening of Power Grid Protection Systems Against Cyber-Attacks: A Comprehensive Review" Iranian Journal of Electrical and Computer Engineering.

[33] Abolfazl Sajadi,Bijan Alizadeh, (2024). "SQ-PUF: A Resistant PUF-Based Authentication Protocol against Machine-Learning Attack" Iranian Journal of Electrical and Computer Engineering.

[34] Boshra Pishgoo, Ahmad akbari azirani. (2022). "Improving IoT Botnet Anomaly Detection Based on Dynamic Feature Selection and Hybrid Processing", Iranian Journal of Electrical and Computer Engineering, B- Computer Engineering, Issue 2.