Journal of Information Systems Engineering and Management
2025, 10(23s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Enhancing Intrusion Detection Systems with Ensemble Models and Hybrid Feature Selection Techniques

Fiona Lawrence [1], Rajesh Nigam[2]

[1]Research Scholar, Assistant Professor, Department of Computer Science and Engineering, Oriental University, Indore, India

[2]Associate Professor, Department of Computer Science and Engineering, Oriental University, Indore, India

Corresponding author : Fiona Lawrence , fionalawrence2908@gmail.com

ARTICLE INFO

Received: 15 Dec 2024

Revised: 05 Feb 2025

Accepted: 24 Feb 2025

ABSTRACT

Detection Systems (IDS) play a critical role in safeguarding networks against cyber attacks. However, selecting the most effective machine learning model for intrusion detection is challenging due to varying dataset characteristics. This research investigates the performance of multiple machine learning models, including SVM (Linear, Poly, RBF, and Sigmoid), LightGBM, XGBoost, and CatBoost, across two widely used datasets: CICIDS2017 and NF-UNSW-NB15. The primary problem is the inconsistency in model performance across different datasets, affecting the reliability of IDS solutions. To address this, we used SMOTE for balancing class distributions and PCA for dimensionality reduction. Each model was evaluated based on accuracy, precision, sensitivity, and F-measure. The results show that LightGBM, XGBoost, and CatBoost consistently outperform the SVM models across both datasets, with accuracy levels above 98%. In contrast, the SVM models exhibited significant variation, performing better on the NF-UNSW-NB15 dataset than on CICIDS2017. Ensemble models are more suitable for intrusion detection due to their higher and more stable performance across different datasets, making them preferable for real-world applications.

Keywords: Detection Systems , Machine Learning , SVM , XGBoost , CatBoost , CICIDS2017 , NF-UNSW-NB15.

1. INTRODUCTION

In today's increasingly interconnected world, cyberattacks are more prevalent than ever, with organizations facing constant threats from various malicious actors. Intrusion Detection Systems (IDS) have become a critical component in the security infrastructure of modern networks. IDSs are designed to monitor network traffic and detect unauthorized access or malicious activities, allowing organizations to protect sensitive data and infrastructure. While IDS solutions have been around for decades, advancements in machine learning (ML) have enabled the development of more intelligent and efficient IDSs capable of identifying novel attacks in real time.

Selecting the most appropriate ML model for an IDS remains a challenging task, as the performance of these models can vary significantly depending on the characteristics of the dataset used.

Traditional ML models, such as Support Vector Machines (SVM), have long been used for intrusion detection due to their robustness in handling classification tasks. However, the advent of **ensemble models** such as **LightGBM**, **XGBoost**, and **CatBoost** has revolutionized the field, offering superior performance through a combination of multiple weak learners into a stronger predictive model. These ensemble models are particularly effective at handling complex, high-dimensional datasets, making them well-suited for intrusion detection tasks. This study aims to explore the efficacy of ensemble models for IDS by incorporating **Principal Component Analysis (PCA)** for dimensionality reduction and **Synthetic Minority Over-sampling Technique (SMOTE)** to address the common issue of imbalanced datasets in network traffic data.

1.1.Novel Contributions of this paper

The novel contributions of this paper are:

1) This research applies PCA and SMOTE in con-junction with various ML models to evaluate their performance on two widely-used intrusion detection datasets: CICIDS2017 and NF-UNSW-NB15. PCA is employed to reduce the dimensionality of the data while retaining the most important features that capture the majority of the variance. This not only simplifies the dataset but also helps in mitigating the risk of over fitting. SMOTE, on the other hand, is used to balance the class distribution by generating synthetic samples for the minority class, ensuring that the model receives a more balanced representation of both normal and malicious traffic.
2) We compare the performance of several models,including traditional SVM variants (Linear, Poly,RBF, Sigmoid) and more modern ensemble modelssuch as LightGBM, XGBoost, and CatBoost. Theperformance metrics used for evaluation are accuracy, precision, sensitivity, and F-measure, all of which provide a comprehensive understanding ofhow well each model performs in identifying andclassifying intrusions.
3) The CICIDS2017 and NF-UNSW-NB15 datasetsare used for this research. CICIDS2017 is a comprehensive dataset that includes both normal network traffic and various types of attacks such asDoS, brute force, and botnet attacks. NF-UNSW-NB15 is a modern dataset that captures real-worldnetwork traffic with a variety of attack scenarios,making it an excellent benchmark for evaluatingIDS models. Both datasets include a wide range ofnetwork features, making them suitable for dimensionality reduction techniques like PCA.
4) Our initial findings show that ensemble models,particularly LightGBM, XGBoost, and CatBoost,outperform traditional SVM models across all keyperformance metrics. When PCA and SMOTE areapplied, the ensemble models consistently achieveaccuracy levels above 98%, with strong performance in precision, sensitivity, and F-measure aswell. In contrast, the performance of the SVM models improves significantly when applied to the NF-UNSW-NB15 dataset, though they still lag behindthe ensemble models.
5) In this research paper, divided into seven sections, the study begins with a Literature Reviewon intrusion detection techniques, followed by theProposed Architecture and Proposed Algorithmfor the IDS. Implementation details are provided,leading to a thorough Result analysis, and the paper concludes with key insights in the Conclusionsection.

2. RELATED PRIOR RESEARCH

Logeswari et al. (2023): Software Defined Networking (SDN) enhances network flexibility and management but also increases vulnerability to attacks. To address this, the authors propose a novel Hybrid Feature Selection-LightGBM (HFS-LGBM) Intrusion Detection System (IDS) for SDN. Their method applies a two-phase feature selection process followed by LightGBM for attack classification, achieving superior performance on the NSL-KDD dataset in terms of accuracy, precision, recall, and F-measure [1].

Musleh et al. (2023): With the rise of IoT devices, securing these networks has become a challenge. This study introduces a machine learning-based IDS for IoT using feature extraction methods and various ML models. Combining VGG-16 with a stacking model yielded the best results, achieving 98.3% accuracy on the IEEE Dataport dataset, demonstrating the importance of effective feature extraction and model selection [2].

Chaganti et al. (2023): As IoT devices become more widespread, so do IoT-based attacks. This paper proposes an LSTM-based intrusion detection system (IDS) for SDN-IoT networks. The proposed system effectively detects and classifies attacks with an accuracy of 0.971, demonstrating the efficacy of LSTM in handling multiclass classification for IoT-based attacks [3].

Kasongo (2023): The increasing data flow in modern networks has heightened security vulnerabilities. This study implements an IDS framework using various Recurrent Neural Networks (RNNs) combined with XGBoost for feature selection. The best results were achieved using XGBoost-LSTM, with a test accuracy of 88.13% on NSL-KDD, demonstrating the framework's effectiveness compared to traditional methods [4].

Verma and Ranga (2023): In an era of massive data generation, securing networks is crucial. The authors emphasize the importance of high-quality datasets for effective intrusion detection. They analyze the CIDDS-001 dataset using various machine learning techniques, showing its complexity and the importance of modern datasets for better intrusion detection, as older datasets like NSL-KDD and KDD 99 are outdated for modern attack scenarios [5].

Alotaibi and Rassam (2023): The paper discusses how traditional IDS approaches struggle with novel attacks, leading to high false alarms. To address this, machine learning (ML) techniques are recommended. However, adversarial machine learning (AML) can exploit IDS vulnerabilities. The paper surveys AML strategies, highlighting attack types and defense mechanisms, and outlines future research directions [6].

Pinto et al. (2023): Industrial control systems (ICS), SCADA, and DCS, which are critical infrastructures (CI), face increasing cyber threats. This paper surveys ML-based IDS techniques used to protect CI, focusing on zero-day attacks. It reviews the security datasets and explores recent advancements in IDS for CI protection [7].

Henry et al. (2023): This study addresses the cybersecurity challenges posed by the growing number of IoT devices, particularly focusing on zero-day attacks. The authors propose a combined CNN-GRU IDS model using the CICIDS-2017 dataset. The results show high detection accuracy of 98.73%, with an improved False Positive Rate (FPR) of 0.075, indicating the model's efficacy [8].

Azam et al. (2023): The paper reviews IDS techniques, discussing challenges such as false positives and detecting new threats. ML and DL techniques are explored as potential solutions for improving IDS. The decision tree model is proposed for detecting anomalies, and the paper emphasizes the need for robust methodologies and dataset selection [9].

Awajan (2023): With the rise in IoT-based attacks, the author proposes a deep learning-based IDS for IoT. The system uses a four-layer fully connected architecture to detect various attacks, achieving an average accuracy of 93.74% and demonstrating effective real-time intrusion detection for IoT devices [10].

Santhosh Kumar et al. (2023): The Internet of Things (IoT) enables smart objects to communicate and transmit data, revolutionizing various sectors. This study focuses on improving IoT security through an IDS based on a fuzzy CNN. The proposed system efficiently detects denial-of-service (DoS) attacks, improves detection accuracy, and reduces false positives by analyzing network security metrics[11].

Hnamte and Hussain (2023): The increasing shift to cyber environments has led to new network vulnerabilities. The authors propose an intelligent network intrusion detection system (NIDS) using deep learning, trained on CICIDS2018 and Edge_IIoT datasets. The system achieves near-perfect accuracy, with 100% and 99.64% in multiclass classification tasks, making it highly effective in network security[12].

Hossain and Islam (2023): Traditional IDSs struggle with unknown sophisticated attacks. This research proposes an ensemble-based ML technique for IDS, using algorithms like Random Forest, Gradient Boosting, and XGBoost. Evaluated on public datasets, the approach exceeds 99% accuracy and offers robust performance in metrics like precision, recall, F1-score, and Cohen's Kappa [13].

Shah et al. (2023): IoT systems are vulnerable to various security threats. The authors propose an AI-based security system that uses binary classification to detect malicious users and blockchain technology for tamper-proof data storage. Deep learning algorithms classify malicious smart contracts, offering a comprehensive security solution for IoT networks[14].

Venkatesan (2023): With the rise of cyber-attacks, this paper explores machine learning algorithms like SVM, Random Forests, and Decision Trees to detect vulnerabilities. Using the NSL-KDD dataset, the study compares the effectiveness of these algorithms in identifying attacks, aiming to determine the best-performing algorithm for intrusion detection[15].

Hidayat et al. (2023): The rise of big data and cloud technologies has increased network attack threats. This research proposes a hybrid feature selection method combining Pearson correlation and random forest, utilizing decision trees, AdaBoost, and KNN for machine learning, and MLP and LSTM for deep learning on the TON_IoT dataset. Decision trees and MLP demonstrated optimal performance with reduced false positives and negatives[16].

Jose and Jose (2023): The increasing use of intelligent devices and network systems leads to more cyberattacks. This study evaluates machine learning and deep learning models on the UNSW-NB15 and NSL-KDD datasets, achieving up to 98.6% accuracy. The results affirm ML techniques as effective for intrusion detection in both two-class and multi-class classification scenarios[17].

Issa and Albayrak (2023): With the proliferation of smart devices, securing them from intruders is critical. This study compares deep learning techniques, including deep neural networks, CNN, and LSTM, using the CIC-IDS 2017 dataset for intrusion detection. The research highlights the effectiveness of these AI models in protecting resource-constrained devices[18].

Maesaroh et al. (2022): DDoS attacks pose significant security threats. This study proposes a deep learning model combining CNN and LSTM, achieving 99.20% accuracy on the NSL-KDD dataset. The seven-layer model outperforms traditional methods, proving its effectiveness in detecting DDoS attacks[19].

Ullah et al. (2022): Denial of Service (DoS) attacks threaten network security. This study suggests a WIDS approach using Linux, Snort, and Iptables to detect and mitigate DoS attacks in wireless networks. The tests in WAN configurations show effective identification and prevention of network attacks[20].

Saba et al. (2022): The shift to online communication during the COVID-19 pandemic necessitated secure systems. This study proposes an IDS for Apache web servers using the Naive Bayes algorithm, trained on an IEEE dataset. The system achieves a cross-validation accuracy of 98.6%, ensuring secure and effective communication between vendors and customers[21].

Naseri and Gharehchopogh (2022): IoT devices enhance lives but face significant security threats. This research introduces a CNN-based anomaly detection IDS for IoT environments. Tested on the NID and BoT-IoT datasets, the proposed model achieves 99.51% and 92.85% accuracy, respectively, demonstrating deep learning's effectiveness in anomaly detection[22].

Kumar et al. (2022): As cyberattacks increase, this study presents a binary Farmland Fertility Algorithm (BFFA) for feature selection in IDS classification. Tested on NSL-KDD and UNSW-NB15 datasets, the BFFA combined with classifiers outperforms traditional methods in accuracy, precision, and recall, improving runtime in feature selection operations[23].**Lo et al. (2022)**: IoT systems face security and privacy issues, exacerbated by centralized storage architectures. This paper proposes a distributed IDS using fog computing to detect DDoS attacks in blockchain-enabled IoT networks. Evaluated on the BoT-IoT dataset, XGBoost outperforms in binary attack detection, while Random Forest excels in multi-attack detection with faster training and testing times on fog nodes[24].

Table 1. Comparison With Existing Literature

Reference no.	Technique	Accuracy (%)
Musleh [?]	Feature extraction	98.62
Chaganti[?]	Multiclass classification	97.1
Kasongo [?]	Feature Selection	88.13
Henry [?]	Feature Optimization	98.73
Awajan [?]	Feature extraction and network classification	93.74
Hossain [?]Trust-based	Ensemble-based ML technique for IDS	99
This paper	Trust-based	

3. PROPOSED ARCHITECTURE

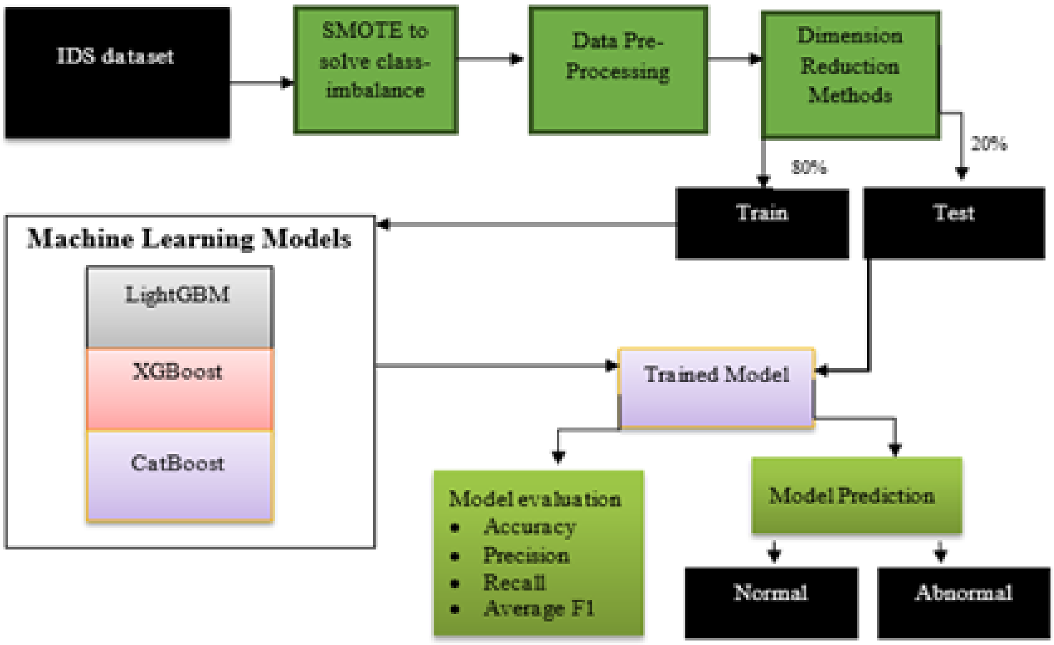

Figure 1. Comprehensive workflow for training and evaluating machine learning models on the CICIDS2017 and NF-UNSW-NB15 datasets

The figure 1 provides a comprehensive workflow for training and evaluating machine learning models on the CICIDS2017 and NF-UNSW-NB15 datasets. The process begins with **data preprocessing**, a crucial step that prepares the raw data for further analysis by cleaning and formatting it for model training. This ensures the datasets are suitable for feature extraction and selection.

Next, the workflow moves to **feature selection**, where two methods are used: **Recursive Method (RM)** and **Lasso Regression**. RM systematically selects important features by recursively eliminating the least significant ones, while Lasso Regression is used to penalize large coefficients, effectively reducing feature complexity and enhancing generalization. Both methods ensure that only the most relevant features are retained for the models, optimizing their predictive performance.

Following feature selection, **PCA (Principal Component Analysis)** is applied for **dimensionality reduction**. PCA transforms the selected features into a smaller set of uncorrelated components, reducing the overall dimensionality of the data while preserving variance. This step mitigates the risk of overfitting and accelerates the training process by simplifying the input data.

The data is then split into **train** and **test** subsets. The training phase utilizes three machine learning models: **LightGBM**, **XGBoost**, and **CatBoost**. Each model undergoes independent training on the processed and reduced data to learn the underlying patterns.

After the training phase, the trained models proceed to the **model evaluation** stage. The performance of each model is evaluated using several key metrics: **accuracy**, **precision**, **recall**, and **F1-score** (both the individual and average F1-score). These metrics provide insight into the effectiveness of each model in identifying true positives, minimizing false positives, and achieving a balance between precision and recall.

Principal Component Analysis (PCA) and **Synthetic Minority Over-sampling Technique (SMOTE)** are two powerful techniques that can significantly improve the performance of machine learning models, particularly when dealing with high-dimensional data and class imbalance. Here's how each contributes to improving model performance:

1. Principal Component Analysis (PCA)

PCA is a dimensionality reduction technique that transforms the input data into a set of uncorrelated variables called principal components. The goal of PCA is to reduce the number of input features while retaining as much variance in the data as possible. Here's how PCA improves model performance:

- **Reduces Overfitting**: High-dimensional datasets often lead to overfitting, where the model learns noise and irrelevant patterns in the data rather than the true underlying relationships. PCA mitigates this risk by reducing the number of input features, helping the model generalize better to unseen data.
- **Improves Computational Efficiency**: High-dimensional datasets require more computational resources, which can slow down model training and inference. By reducing the number of features through PCA, the model becomes faster and more efficient to train, especially for algorithms that do not scale well with large feature sets (e.g., Support Vector Machines).
- **Decreases Multicollinearity**: In many datasets, features are highly correlated, which can negatively impact the performance of models, particularly linear models like Logistic Regression. PCA eliminates this multicollinearity by converting correlated features into uncorrelated principal components, leading to better model performance.
- **Focuses on Important Variance**: PCA ranks the new components by the amount of variance they capture, allowing the model to focus on the most informative features. This helps improve predictive accuracy by retaining only the most important parts of the data.

2. Synthetic Minority Over-sampling Technique (SMOTE)

SMOTE is a technique used to handle class imbalance in datasets, where one class (usually the minority class) has significantly fewer instances than the other(s). Imbalanced datasets often lead to biased models that perform poorly on the minority class. Here's how SMOTE helps improve model performance:

- **Balances the Dataset**: SMOTE generates synthetic samples for the minority class by interpolating between existing minority class samples. This helps balance the dataset, reducing the bias toward the majority class and allowing the model to learn better decision boundaries for both classes.
- **Improves Recall for the Minority Class**: Models trained on imbalanced data often perform poorly on the minority class, resulting in low recall (i.e., the ability to correctly identify positive instances). By balancing the classes, SMOTE improves recall, leading to better detection of minority class instances, which is especially important in applications like fraud detection, medical diagnosis, and anomaly detection.

- **Prevents Overfitting**: Unlike random over-sampling (which duplicates minority class instances), SMOTE generates synthetic samples, reducing the likelihood of overfitting to repeated samples. This helps the model generalize better to unseen data and increases its robustness.
- **Works Well with Decision Trees and Ensembles**: Algorithms like Decision Trees, Random Forests, and Gradient Boosting (e.g., XGBoost, LightGBM) can benefit significantly from SMOTE, as balanced datasets allow them to create more balanced splits and improve overall model performance.

4. PROPOSED ALGORITHM

4.1 LightGBM Pseudocode for Intrusion Detection System

Input: Preprocessed IDS dataset D with N samples, M features

Output: Trained LightGBM model for Intrusion Detection

1. Initialize model parameters:

- Number of trees T
- Learning rate α
- Maximum tree depth d
- Minimum data in leaf l_min
- Objective: Binary classification (Normal vs Anomalous)

2. For each tree t from 1 to T do:

- Compute gradient (first derivative) and hessian (second derivative) of loss function for each data point in the dataset.
- Build a decision tree:
- For each feature f, sort the data points and calculate the optimal split point.
- (Split is chosen to maximize the information gain or minimize loss).
- Grow the tree by recursively splitting nodes until:
- Maximum depth d is reached, or
- Number of samples in leaf node < l_min
- Use a leaf-wise (best-first) approach, splitting the leaf with the largest loss reduction.
- Update the model with the new tree's predictions:
- Update each data point's predicted value using the learning rate α.

3. Output the final ensemble model of T trees.

4. Use the trained model to classify new samples (Normal vs Anomalous) based on the learned decision rules.

4.2 XGBoost Pseudocode for Intrusion Detection System

Input: Preprocessed IDS dataset D with N samples, M features

Output: Trained XGBoost model for Intrusion Detection

- Initialize model parameters:
- Number of trees T
- Learning rate α
- Maximum tree depth d
- Minimum data in leaf l_min
- Regularization parameters (λ, γ)
- Initialize predictions as the base value (log odds of the majority class).
- For each tree t from 1 to T do:
- Compute pseudo-residuals (negative gradients) for each data point:
- residual_i = (true_label_i - predicted_label_i)
- Construct a new decision tree:

- For each feature f, compute the best split based on:
- Maximizing the reduction in loss function (binary log loss).
- Apply regularization (λ, γ) to control model complexity.
- Grow the tree until:
- Maximum depth d is reached, or
- Number of samples in leaf node < l_min
- Add the new tree's contribution to the overall model prediction:
- predicted_label_i = previous_predicted_label_i + α * tree_output_i
- Apply regularization to prune the tree if necessary to avoid overfitting.
- Output the ensemble of T trees.
- Use the model to classify new network events into "Normal" or "Anomalous" based on learned patterns.

4.3 CatBoost Pseudocode for Intrusion Detection System

Input: Preprocessed IDS dataset D with N samples, M features (including categorical features)

Output: Trained CatBoost model for Intrusion Detection

- Initialize model parameters:
- Number of trees T
- Learning rate α
- Maximum tree depth d
- Categorical feature handling strategy
- Objective: Binary classification (Normal vs Anomalous)
- Preprocess categorical features:
  - Apply CatBoost's ordered target statistics or one-hot encoding to handle categorical data:
- Calculate statistics for each categorical feature based on the order of data points.
- For each tree t from 1 to T do:
  - Compute the gradient (first derivative) of the loss function for each data point.
  - Build a symmetric decision tree:
- For each feature (including transformed categorical features), calculate the best split.
- ii. Grow the tree by recursively splitting nodes to minimize loss:
- Maximum depth d is used to limit tree complexity.
  - Update the model:
- Add the tree's predictions to the overall model predictions using learning rate α.
- Apply boosting with permutation-driven categorical feature handling to reduce overfitting.
- Output the final model consisting of T trees.
- Classify new network events or logs as "Normal" or "Anomalous" using the trained model.

5. IMPLEMENTATION

5.1 Dataset

CICIDS2017 Dataset: The **CICIDS2017 dataset** is a widely used dataset for intrusion detection system (IDS) research. It was created by the Canadian Institute for Cybersecurity to reflect real-world network traffic and attack scenarios. This dataset includes normal traffic as well as a variety of attack types such as brute force, Denial of Service (DoS), Distributed Denial of Service (DDoS), infiltration, web attacks, and botnets, among others. The data captures several days of network traffic, featuring over 80 network traffic features, including basic network information (e.g., source and destination IP addresses, port numbers, and protocols), time-based statistics (e.g., packet count, flow duration), and advanced traffic metrics related to packet flow, payload size, and more.The **CICIDS2017** dataset is structured to represent real-world traffic by simulating a typical corporate network with various users and services running simultaneously, including FTP, HTTP, HTTPS, SSH, and email. The dataset is designed to evaluate the performance of machine learning algorithms for anomaly detection, where models must distinguish between benign network behavior and malicious intrusions. The complexity and diversity of the attack types in the dataset make it ideal for developing IDSs that can detect known attacks while being adaptable enough to identify new and evolving threats. Furthermore, its balanced representation of both normal and malicious traffic ensures that machine learning models trained on this dataset do not suffer from data imbalance issues.

Source : https://www.unb.ca/cic/datasets/ids-2017.html

NF-UNSW-NB15 Dataset: The **NF-UNSW-NB15 dataset** is another prominent dataset used in network intrusion detection research. Created by the University of New South Wales (UNSW), this dataset is a more recent and modern representation of network traffic, capturing both contemporary network traffic and attack types. It was designed to address some of the limitations present in previous IDS datasets by introducing new types of modern-day attacks that were not present in older datasets. The attacks in this dataset include Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms, making it a comprehensive dataset for studying both traditional and novel attack vectors.The dataset was generated in a controlled network environment using the IXIA PerfectStorm tool, which generated realistic traffic flows from a variety of users and applications. **NF-UNSW-NB15** consists of a combination of both labeled normal and malicious traffic records, with over 49 features capturing network-level details such as protocol type, service type, source and destination IP addresses, TCP flags, and traffic volume. These features include both basic traffic data (e.g., flow size, number of packets) and more advanced features that can be used to analyze traffic behavior over time.

Source : https://research.unsw.edu.au/projects/unsw-nb15-dataset

5.2 Illustrative example

5.2.1 Illustrative example of LightGBM

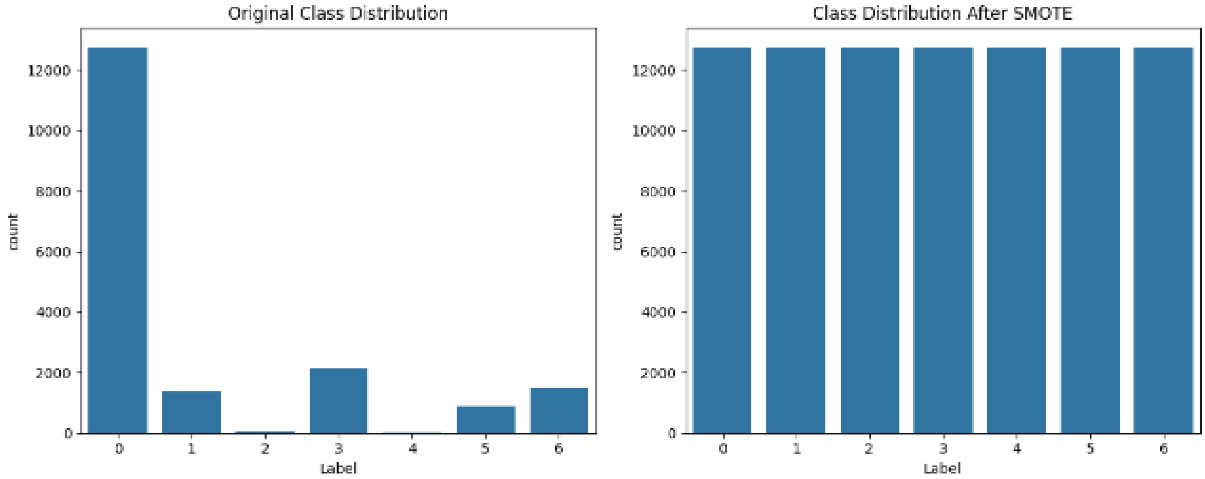

Figure 2. The class distribution before and after applying SMOTEof LightGBM

The figure 2 illustrates the class distribution before and after applying SMOTE (Synthetic Minority Over-sampling Technique). Initially, there is a significant imbalance with class "0" dominating, while after applying SMOTE, all classes are balanced with an equal number of samples, improving the dataset's suitability for training machine learning models.

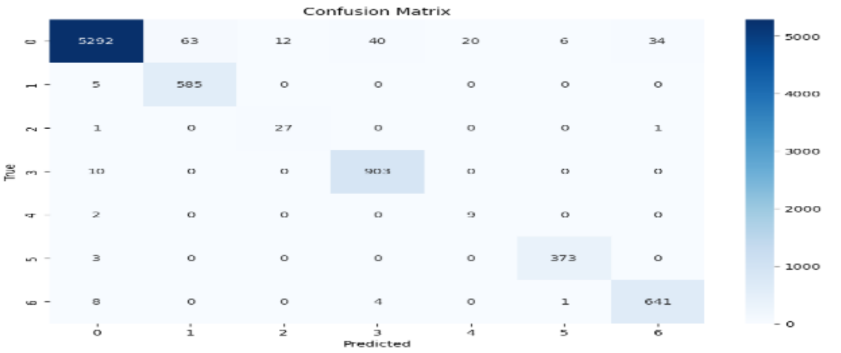

Figure 3. Confusion matrix shows strong model performanceof LightGBM

The figure 3 shows confusion matrix shows strong model performance with the majority of true classes correctly classified, especially for class "0" (5292 correct predictions). However, some misclassifications are evident, such as minor errors across classes like "1," "3," and "6," where a few instances are mispredicted as other classes.

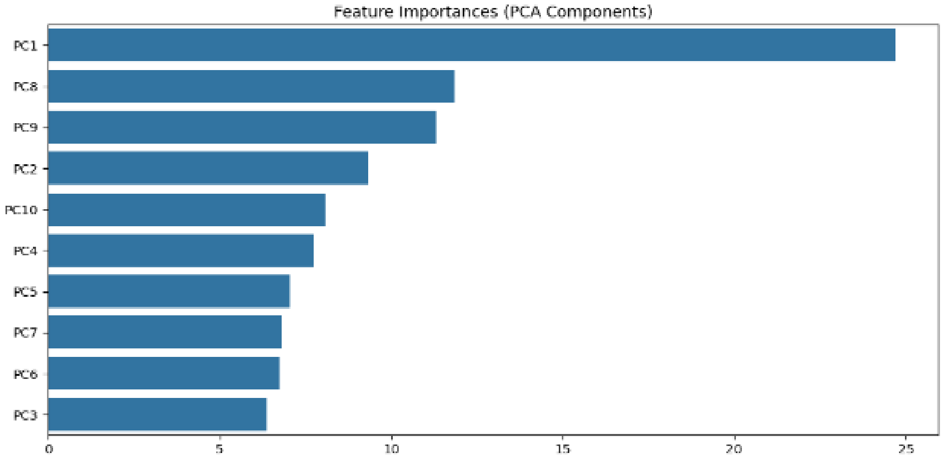

Figure 4. The feature importance of principal components (PCA components) in the datasetof LightGBM

The figure 4 illustrates the feature importance of principal components (PCA components) in the dataset. PC1 has the highest importance, significantly contributing to model performance, followed by PC8, PC9, and PC2. These components capture the most variance, making them crucial for reducing dimensionality while retaining important information for predictions.

5.2.2 Illustrative example of XGBoost

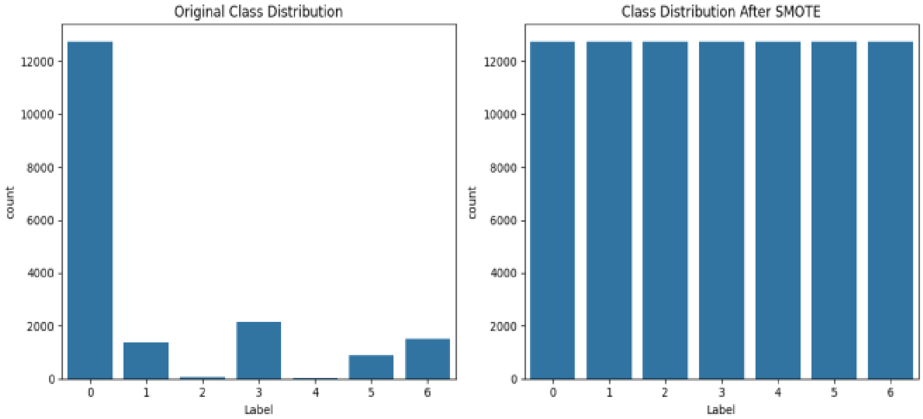

Figure 5. The class distribution before and after applying SMOTEof XGBoost

The figure 5 shows the class distribution before and after applying SMOTE. Initially, class "0" dominates, with other classes significantly underrepresented. After applying SMOTE, the class distribution is balanced across all labels, ensuring that each class has an equal number of samples, improving the fairness and performance of machine learning models.

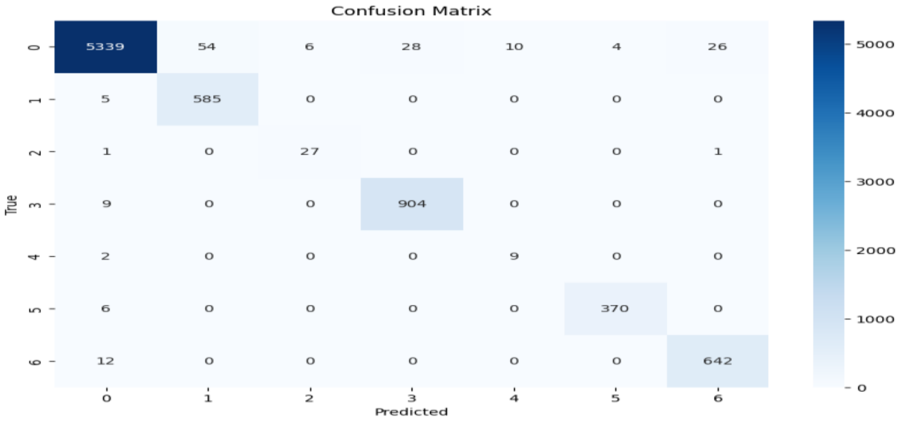

Figure 6. Confusion matrix shows strong model performanceof XGBoost

The figure 6 shows confusion matrix shows the model's prediction performance across various classes. Class "0" has the highest correct predictions (5339), but some misclassifications occur across other classes, such as minor errors in classes "1," "3," and "6." Overall, the model performs well but has room for improvement in specific classes.

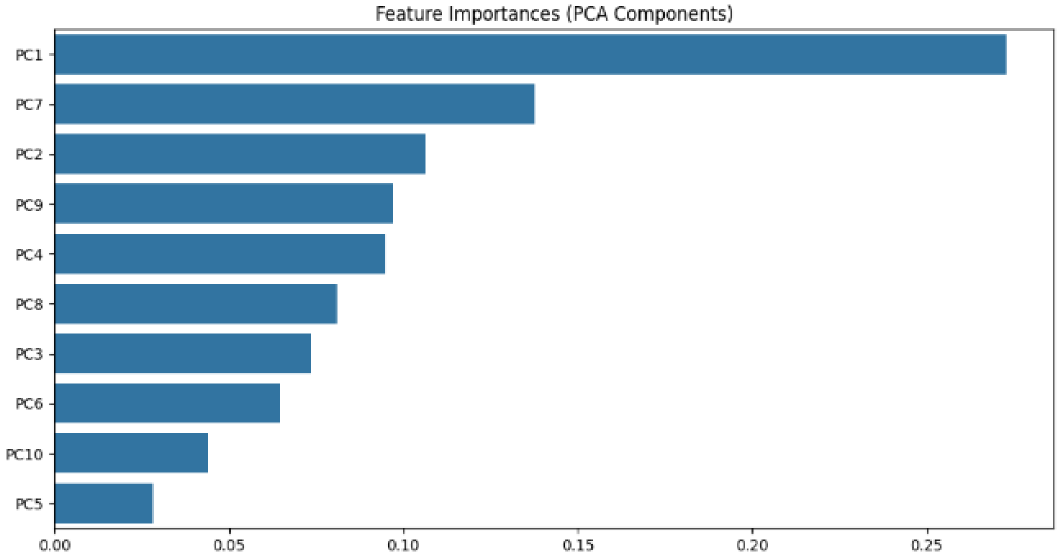

Figure 7. The feature importance of principal components (PCA components) in the datasetof XGBoost

The figure 7 highlights the importance of PCA components in the model. PC1 has the highest influence, followed by PC7, PC2, and PC9. These principal components capture the most variance in the data, playing a crucial role in improving model performance by reducing dimensionality while retaining significant information.

5.2.3 Illustrative example of CatBoost

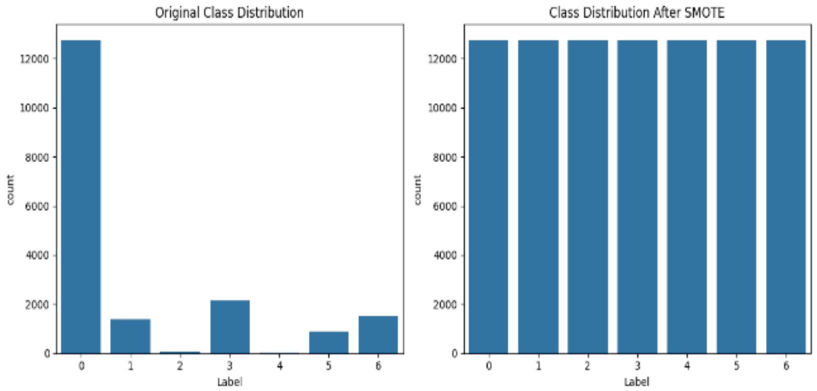

Figure 8. The class distribution before and after applying SMOTEof CatBoost

The figure 8 compares class distributions before and after applying SMOTE. Initially, there is a significant imbalance, with class "0" heavily dominating. After applying SMOTE, all classes are evenly distributed, effectively addressing class imbalance and providing a balanced dataset for better machine learning model training and evaluation.

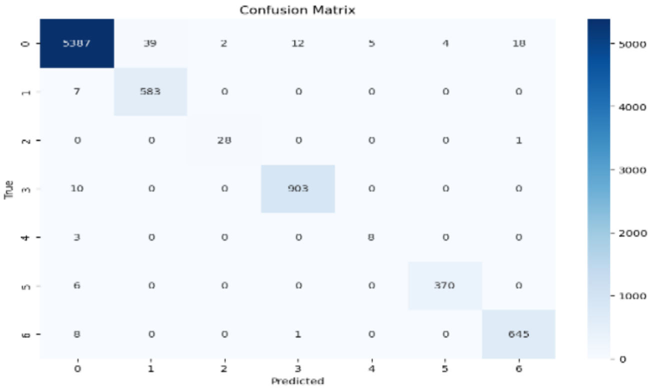

Figure 9. Confusion matrix shows strong model performanceof CatBoost

The figure 9 shows confusion matrix shows the model's performance across multiple classes. Class "0" has the highest correct predictions (5387), while other classes like "3" and "6" also perform well. Some minor misclassifications occur, especially between classes "0" and others, indicating areas where model accuracy could be improved.

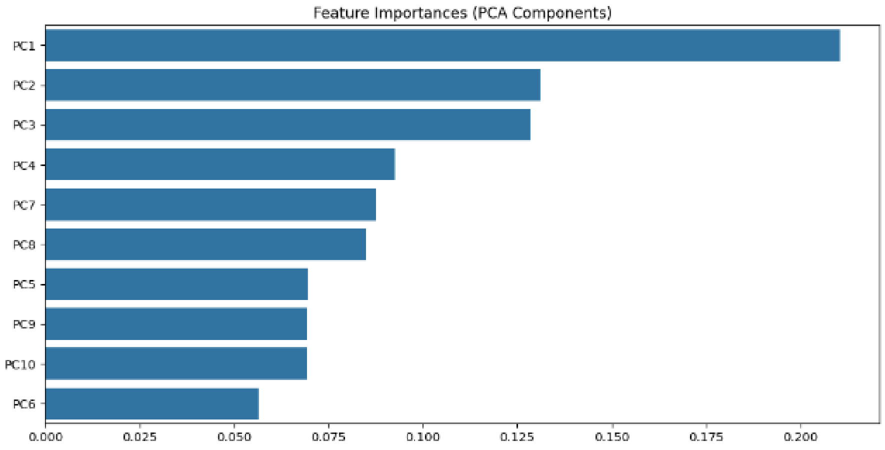

Figure 10. The feature importance of principal components (PCA components) in the datasetof CatBoost

The figure 10 illustrates the importance of various PCA components in the model. PC1 has the highest significance, followed by PC2 and PC3. These components capture most of the data's variance, playing a crucial role in reducing dimensionality while preserving key information, thereby enhancing model efficiency and performance.

6. RESULT

Table 2. Performance comparison of various models on the CICIDS2017 dataset

	CICIDS2017 dataset			
	Accuracy (%)	**Precision (%)**	**Sensitivity (%)**	**F-measure (%)**
SVM-Linear	32.84	61	33	25
SVM-Poly	17.65	18	18	15
SVM-rbf	80.71	88	81	83
SVM-Sigmoid	53.38	75	53	57
LightGBM	98.56	98	98	98
XGBoost	97.96	98	98	98
CatBoost	97.39	98	97	97

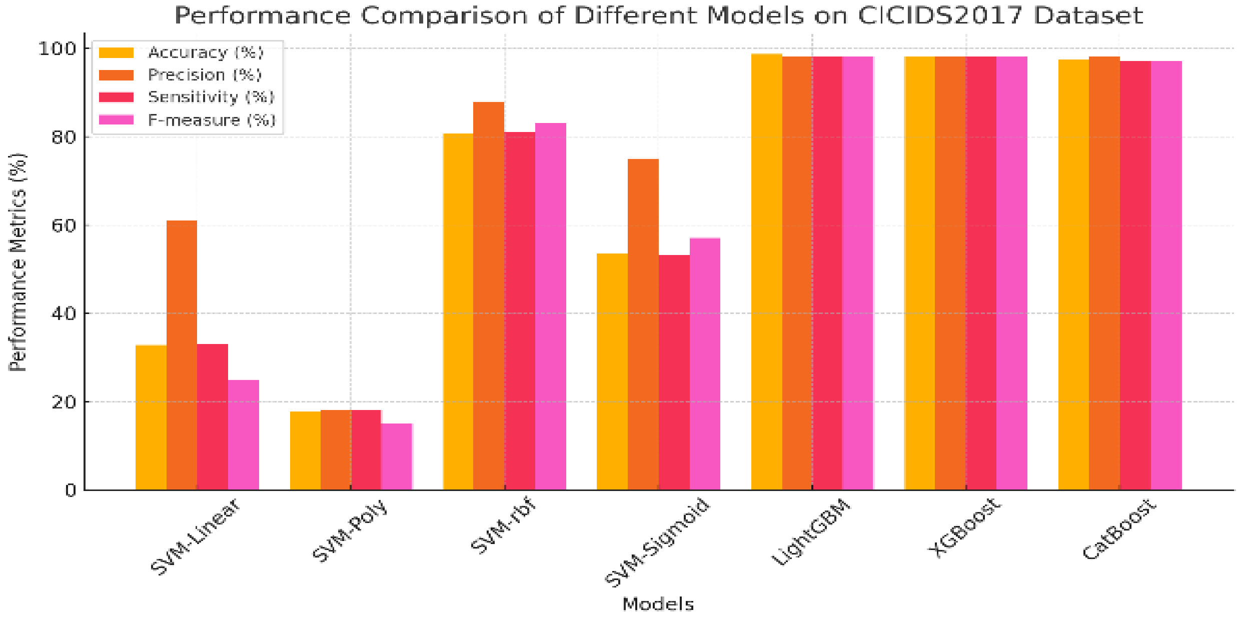

Figure 11. Performance comparison of various models on the CICIDS2017 dataset

The table 1 and figure 11presents a performance comparison of various models SVM (Linear, Poly, RBF, and Sigmoid), LightGBM, XGBoost, and CatBoost on the CICIDS2017 dataset, evaluating metrics such as accuracy, precision, sensitivity, and F-measure. The performance of the **SVM-Linear** model shows lower accuracy (32.84%), sensitivity (33%), and F-measure (25%), though it achieves a precision of 61%. **SVM-Poly** performs poorly across all metrics, with accuracy (17.65%) being the lowest among the models. **SVM-RBF** shows significant improvement, with accuracy reaching 80.71% and strong performance across other metrics (precision: 88%, sensitivity: 81%, F-measure: 83%).**SVM-Sigmoid** offers moderate performance with accuracy at 53.38%, precision at 75%, and lower sensitivity and F-measure scores. The **LightGBM** model outperforms the SVM variants with high accuracy (98.56%), precision, sensitivity, and F-measure, all reaching around 98%, making it a top performer. Similarly, **XGBoost** and **CatBoost** follow closely, both achieving high and balanced performance across all metrics, with accuracy, precision, sensitivity, and F-measure hovering around 98%.Overall, LightGBM, XGBoost, and CatBoost outperform the SVM variants in all key performance indicators, especially in accuracy and F-measure, making them more suitable for the CICIDS2017 dataset in intrusion detection applications.

Table 3. Performance comparison of various models on the NF-UNSW-NB15 dataset.

NF-UNSW-NB15				
	Accuracy (%)	**Precision (%)**	**Sensitivity (%)**	**F-measure (%)**
SVM-Linear	91.78	91.5	93.94	73.28
SVM-Poly	93.28	94.16	95.67	75.7
SVM-rbf	91.71	90.18	93.23	75.93
SVM-Sigmoid	92.49	92.13	94.44	75.82
LightGBM	99.34	99	99	99
XGBoost	98.75	98	98	98
CatBoost	98.28	98	98	98

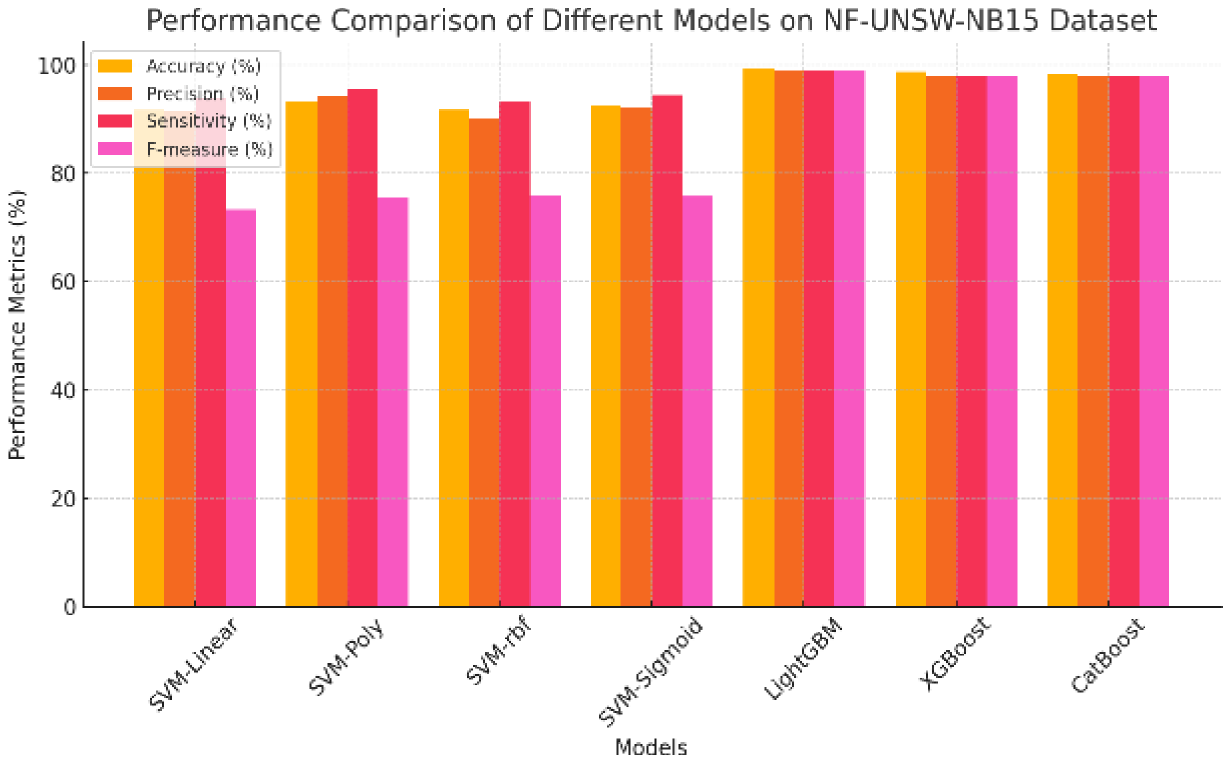

Figure 12. Performance comparison of various models on the NF-UNSW-NB15 dataset

The table 3 and figure 12compares the performance of different models SVM (Linear, Poly, RBF, and Sigmoid), LightGBM, XGBoost, and CatBoost on the NF-UNSW-NB15 dataset using key metrics like accuracy, precision, sensitivity, and F-measure. The **SVM-Linear** model shows strong performance with an accuracy of 91.78%, precision at 91.5%, and sensitivity at 93.94%, though its F-measure is notably lower at 73.28%. **SVM-Poly** outperforms the linear variant in all aspects, achieving 93.28% accuracy, 94.16% precision, and a higher F-measure of 75.7%. **SVM-RBF** performs similarly, with slight dips in precision and accuracy, but still maintains solid metrics. **SVM-Sigmoid** shows moderate performance with accuracy at 92.49%, precision at 92.13%, and lower sensitivity and F-measure scores.In comparison, the ensemble models **LightGBM**, **XGBoost**, and **CatBoost**outshine the SVM models, with all three models reaching high accuracies (above 98%), and achieving near-perfect precision, sensitivity, and F-measure scores, hovering around 98-99%. These results demonstrate that the ensemble-based models (LightGBM, XGBoost, CatBoost) are better suited for intrusion detection tasks on the NF-UNSW-NB15 dataset due to their superior performance across all evaluated metrics. The SVM models, while competitive, are outperformed by these more advanced machine learning algorithms.

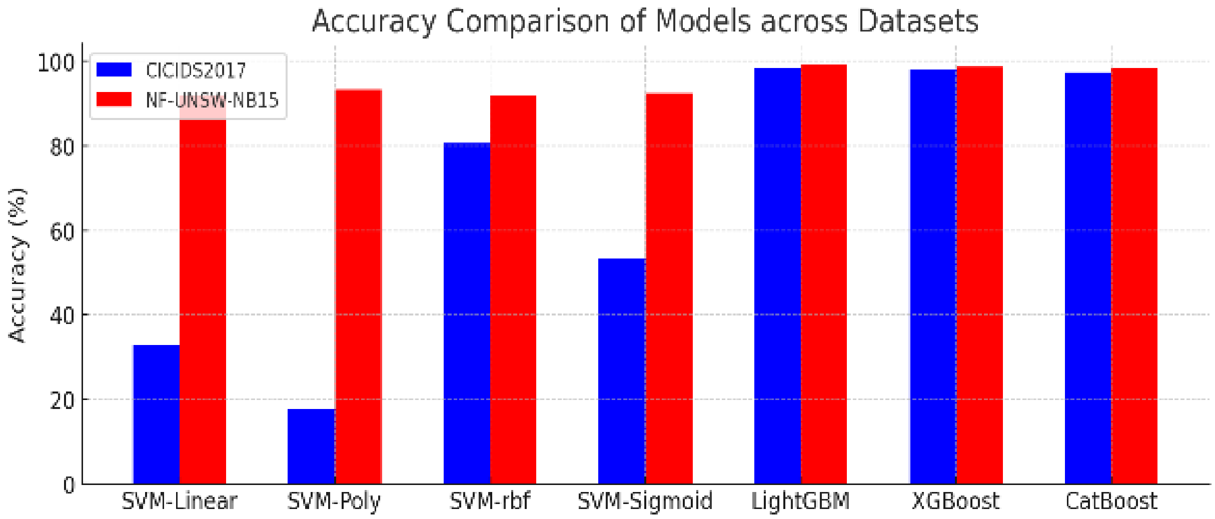

Figure 13. Compares the accuracy of various models.

The figure 13compares the accuracy of various models SVM (Linear, Poly, RBF, Sigmoid), LightGBM, XGBoost, and CatBoost across two datasets: CICIDS2017 and NF-UNSW-NB15. The ensemble models (LightGBM, XGBoost, CatBoost) perform consistently well on both datasets, achieving accuracy above 97%. In contrast, the SVM models show a significant disparity in performance between the datasets, especially **SVM-Linear** and **SVM-Poly**, which perform much better on NF-UNSW-NB15. Overall, the models perform better on NF-UNSW-NB15, with clear accuracy gains compared to CICIDS2017.

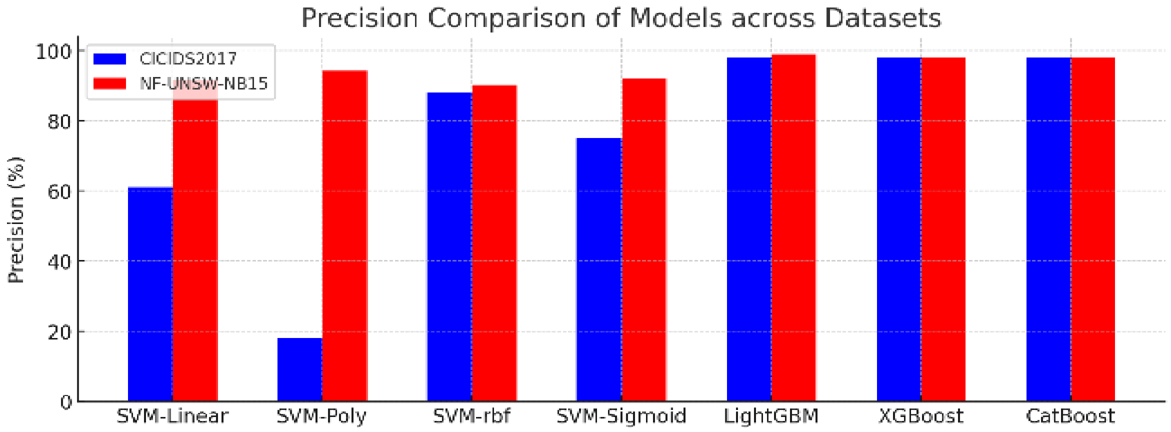

Figure 14. Compares the precision of various models.

The figure 14compares the precision of different models SVM (Linear, Poly, RBF, Sigmoid), LightGBM, XGBoost, and CatBoost across the CICIDS2017 and NF-UNSW-NB15 datasets. The ensemble models (LightGBM, XGBoost, and CatBoost) maintain high precision (around 98%) on both datasets. SVM models show significant variation, particularly **SVM-Linear** and **SVM-Poly**, which achieve much higher precision on NF-UNSW-NB15 compared to CICIDS2017. Overall, the precision values are consistently higher for the NF-UNSW-NB15 dataset across all models, indicating better performance on this dataset.

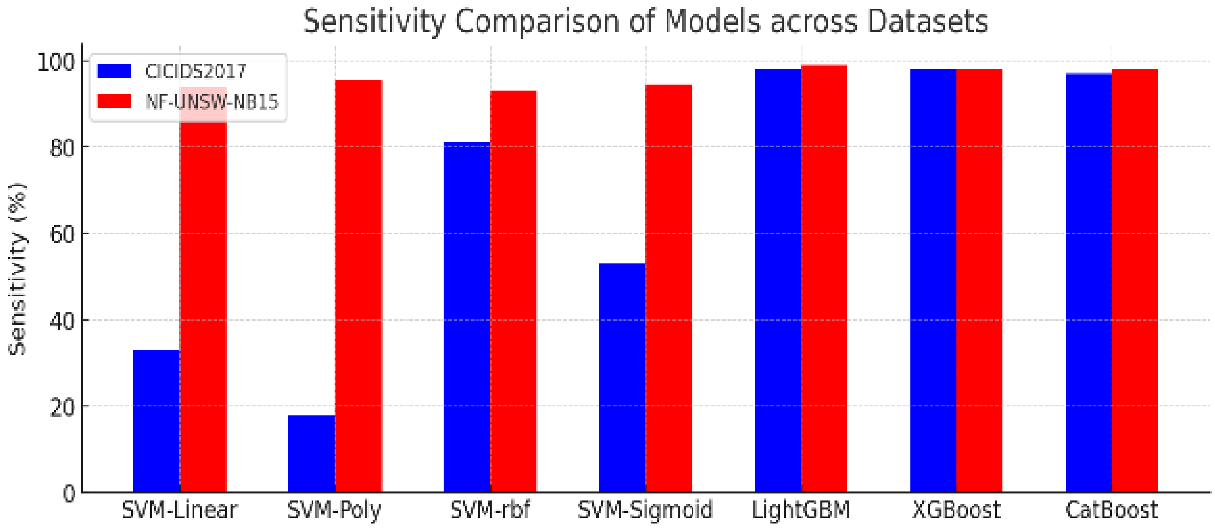

Figure 15. Compares the sensitivity of various models.

The figure 15compares the sensitivity of various models SVM (Linear, Poly, RBF, Sigmoid), LightGBM, XGBoost, and CatBoost across the CICIDS2017 and NF-UNSW-NB15 datasets. Ensemble models like **LightGBM**, **XGBoost**, and **CatBoost** show consistently high sensitivity (around 98%) on both datasets. The SVM models exhibit substantial variability, with **SVM-Linear** and **SVM-Poly** performing significantly better on NF-UNSW-NB15 than on CICIDS2017. Overall, sensitivity values are higher for NF-UNSW-NB15, indicating that models are better at detecting true positives in this dataset compared to CICIDS2017.

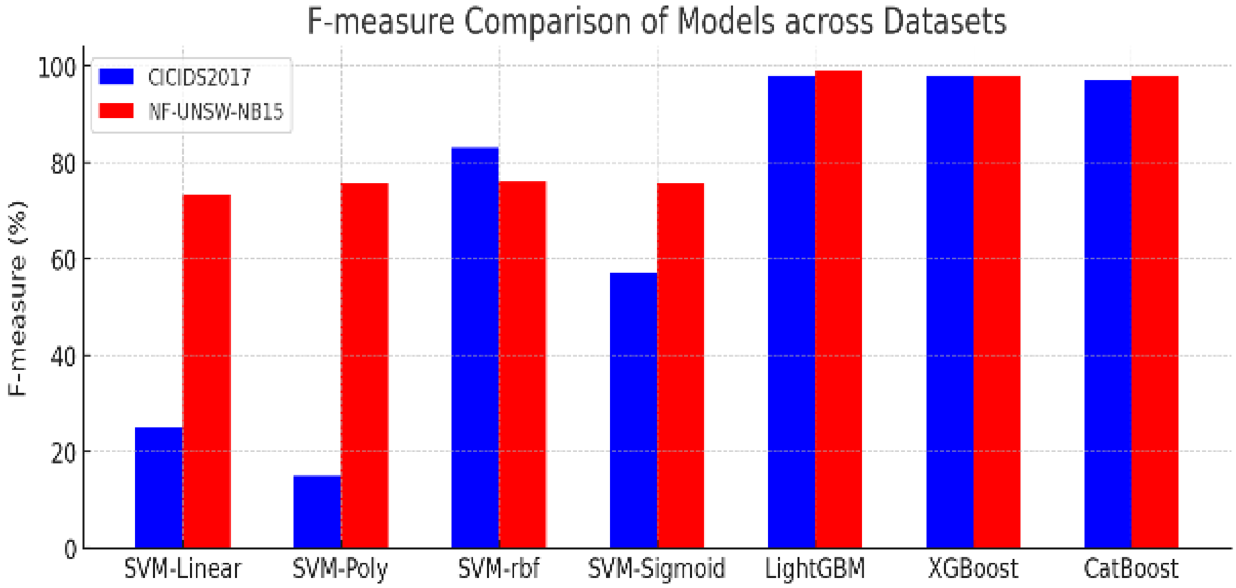

Figure 16. Compares the F-measure of various models.

The figure 16compares the F-measure of different models SVM (Linear, Poly, RBF, Sigmoid), LightGBM, XGBoost, and CatBoost across the CICIDS2017 and NF-UNSW-NB15 datasets. The ensemble models (LightGBM, XGBoost, and CatBoost) achieve consistently high F-measure scores (around 98%) on both datasets. In contrast, the SVM models show notable variation, particularly **SVM-**

Linear and **SVM-Poly**, which perform significantly better on NF-UNSW-NB15 compared to CICIDS2017. Overall, the models demonstrate higher F-measure values on NF-UNSW-NB15, indicating better balance between precision and recall on this dataset.

7. CONCLUSION

The analysis of various machine learning models for intrusion detection systemsacross two datasets, **CICIDS2017** and **NF-UNSW-NB15**, highlights the differences in model performance in terms of key metrics such as accuracy, precision, sensitivity, and F-measure. **LightGBM**, **XGBoost**, and **CatBoost**, which are ensemble-based models, consistently outperformed the traditional **SVM** variants in all evaluated metrics on both datasets.From the **accuracy** comparison, the ensemble models exhibited high and stable performance across both datasets, achieving accuracy levels above 97%. In contrast, the SVM models showed significant variation, with **SVM-Linear** and **SVM-Poly** performing far better on the NF-UNSW-NB15 dataset compared to the CICIDS2017 dataset. This trend continued with **precision**, where the ensemble models maintained nearly perfect precision across both datasets, while the SVM models showed better precision on NF-UNSW-NB15.The **sensitivity** analysis further reinforced this finding. While the ensemble models demonstrated consistently high sensitivity on both datasets, the SVM models, particularly **SVM-Linear** and **SVM-Poly**, exhibited a marked improvement on NF-UNSW-NB15. This pattern also extended to the **F-measure**, where the ensemble models achieved high scores, while the SVM models saw their performance increase on the NF-UNSW-NB15 dataset.The ensemble-based models—**LightGBM**, **XGBoost**, and **CatBoost**—are highly effective for intrusion detection tasks on both CICIDS2017 and NF-UNSW-NB15 datasets. While the **SVM** models demonstrate potential, especially on NF-UNSW-NB15, their performance is less consistent across different datasets. This analysis suggests that ensemble models provide better generalization and stability across varying data distributions, making them more suitable for real-world intrusion detection systems.

REFERENCES

[1] Logeswari, G., Bose, S., & Anitha, T. (2023). An intrusion detection system for sdn using machine learning. *Intelligent Automation & Soft Computing*, *35*(1), 867-880.

[2] Musleh, D., Alotaibi, M., Alhaidari, F., Rahman, A., & Mohammad, R. M. (2023). Intrusion Detection System Using Feature Extraction with Machine Learning Algorithms in IoT. *Journal of Sensor and Actuator Networks*, *12*(2), 29.

[3] Chaganti, R., Suliman, W., Ravi, V., & Dua, A. (2023). Deep learning approach for SDN-enabled intrusion detection system in IoT networks. *Information*, *14*(1), 41.

[4] Kasongo, S. M. (2023). A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework. *Computer Communications*, *199*, 113-125.

[5] Verma, A., & Ranga, V. (2023). On evaluation of network intrusion detection systems: Statistical analysis of CIDDS-001 dataset using machine learning techniques. *Authorea Preprints*.

[6] Alotaibi, A., & Rassam, M. A. (2023). Adversarial machine learning attacks against intrusion detection systems: A survey on strategies and defense. *Future Internet*, *15*(2), 62.

[7] Pinto, A., Herrera, L. C., Donoso, Y., & Gutierrez, J. A. (2023). Survey on Intrusion Detection Systems Based on Machine Learning Techniques for the Protection of Critical Infrastructure. *Sensors*, *23*(5), 2415.

[8] Henry, A., Gautam, S., Khanna, S., Rabie, K., Shongwe, T., Bhattacharya, P., ... & Chowdhury, S. (2023). Composition of hybrid deep learning model and feature optimization for intrusion detection system. *Sensors*, *23*(2), 890.

[9] Azam, Z., Islam, M. M., & Huda, M. N. (2023). Comparative analysis of intrusion detection systems and machine learning based model analysis through decision tree. *IEEE Access*.

[10] Awajan, A. (2023). A novel deep learning-based intrusion detection system for IOT networks. *Computers*, *12*(2), 34.

[11] Santhosh Kumar, S. V. N., Selvi, M., & Kannan, A. (2023). A comprehensive survey on machine learning-based intrusion detection systems for secure communication in internet of things. *Computational Intelligence and Neuroscience*, *2023*.

[12] Hnamte, V., & Hussain, J. (2023). DCNNBiLSTM: An efficient hybrid deep learning-based intrusion detection system. *Telematics and Informatics Reports*, *10*, 100053.

[13] Hossain, M. A., & Islam, M. S. (2023). Ensuring network security with a robust intrusion detection system using ensemble-based machine learning. *Array*, *19*, 100306.

[14] Shah, H., Shah, D., Jadav, N. K., Gupta, R., Tanwar, S., Alfarraj, O., ... & Marina, V. (2023). Deep learning-based malicious smart contract and intrusion detection system for IoT environment. *Mathematics*, *11*(2), 418.

[15] Venkatesan, S. (2023). Design an intrusion detection system based on feature selection using ML algorithms. *Mathematical Statistician and Engineering Applications*, *72*(1), 702-710.

[16] Hidayat, I., Ali, M. Z., & Arshad, A. (2023). Machine Learning-Based Intrusion Detection System: An Experimental Comparison. *Journal of Computational and Cognitive Engineering*, *2*(2), 88-97.

[17] Hidayat, I., Ali, M. Z., & Arshad, A. (2023). Machine Learning-Based Intrusion Detection System: An Experimental Comparison. *Journal of Computational and Cognitive Engineering*, *2*(2), 88-97.

[18] Jose, J., & Jose, D. V. (2023). Deep learning algorithms for intrusion detection systems in internet of things using CIC-IDS 2017 dataset. *International Journal of Electrical and Computer Engineering (IJECE)*, *13*(1), 1134-1141.

[19] Issa, A. S. A., & Albayrak, Z. (2023). Ddos attack intrusion detection system based on hybridization of cnn and lstm. *Acta Polytechnica Hungarica*, *20*(2), 1-19.

[20] Maesaroh, S., Kusumaningrum, L., Sintawana, N., Lazirkha, D. P., & Dinda, R. (2022). Wireless Network Security Design And Analysis Using Wireless Intrusion Detection System. *International Journal of Cyber and IT Service Management*, *2*(1), 30-39.

[21] Ullah, M. U., Hassan, A., Asif, M., Farooq, M. S., & Saleem, M. (2022). Intelligent Intrusion Detection System for Apache Web Server Empowered with Machine Learning Approaches. *International Journal of Computational and Innovative Sciences*, *1*(1), 21-27.

[22] Saba, T., Rehman, A., Sadad, T., Kolivand, H., & Bahaj, S. A. (2022). Anomaly-based intrusion detection system for IoT networks through deep learning model. *Computers and Electrical Engineering*, *99*, 107810.

[23] Naseri, T. S., & Gharehchopogh, F. S. (2022). A feature selection based on the farmland fertility algorithm for improved intrusion detection systems. *Journal of Network and Systems Management*, *30*(3), 40.

[24] Kumar, R., Kumar, P., Tripathi, R., Gupta, G. P., Garg, S., & Hassan, M. M. (2022). A distributed intrusion detection system to detect DDoS attacks in blockchain-enabled IoT network. *Journal of Parallel and Distributed Computing*, *164*, 55-68.