Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

Intrusion Detection System using Machine Learning

Mrs. K P Sangeetha
Assistant Professor
Cyber Security Department ACS
College of Engineering 207,
Kambipura, Mysore Road,
Kengeri Hobli,
Bengaluru-560074
kpsangeetha20@gmail.com

V Swathi
Bachelor of Engineering Student
Cyber Security Department ACS
College of Engineering 207,
Kambipura, Mysore Road,
Kengeri Hobli,
Bengaluru-560074
swathi.vpanda@gmail.com

Harshitha S
Bachelor of Engineering Student
Cyber Security Department ACS
College of Engineering 207,
Kambipura, Mysore Road,
Kengeri Hobli,
Bengaluru-560074
harshithaa359@gmail.com

Abstract — The increasing reliance on internet connectivity necessitates the creation of robust and efficient security measures. This study describes an automated surveillance system that employs machine learning methods to address issues such as inaccurate data and component overlap. The suggested method improves classification performance on typical datasets such as UNSW-NB15 and NSL-KDD by utilizing a preprocessing pipeline that includes Random Oversampling, Stacking Feature Integration, and Principal Component Analysis. A comparison of RF, SVM, KNN, and XGBoost reveals that ensemble approaches outperform individual classifiers, demonstrating the framework's suitability for scalable and proactive intrusion detection.

Keywords — Intrusion Detection, Adaptive Learning, Data Imbalance, Ensemble Methods, Cyber Defense.

I. INTRODUCTION

The rapid advancement of information technology and interconnected systems has revolutionised activities for individuals, organisations, and governments. While this digital transition promotes connectivity and innovation, it also raises risk to various cyber-attacks, including viruses, Denial of Service (DoS) assaults, and data breaches.

As a result, the increasing number of threats necessitates intelligent and effective defence measures capable of discovering and neutralising intrusions in real time. Intrusion Detection Systems (IDS) are an important part of modern security architectures.

Classical IDS systems, such as signature-based or rule-oriented solutions, are effective against established attack patterns but frequently fail to detect new or evolving threats. As security hazards become more intricate, the limitations of traditional systems highlight the need for dynamic, data-driven defence options.

A. Problem Statement

This study addresses the following main challenges for ML-based Intrusion Detection Systems (IDS):

- Many overlapping or redundant features lead to increased complexity and decreased performance.
- Primitive attributes are ineffective in capturing temporal sequences.
- Current IDS approaches generally struggle in real-time or large-scale networked environments.
- IDS datasets often have far more normal examples than attack samples, resulting in biased model predictions.

B. Objectives

The primary objective is to provide an effective machine learning-based intrusion detection mechanism capable of managing imbalanced and high-dimensional network data. The specific aims include:

- Compare the efficiency of several ML strategies—SVM, Random Forest, KNN, Naïve Bayes, and XGBoost—for detecting network attacks.
- Use Random Oversampling (RO) to address data imbalance and increase attack recognition precision.
- Compare the suggested scheme against benchmark datasets (UNSW-NB15 and NSL-KDD) using metrics such as precision, recall, accuracy, and F1-score.
- Determine the optimal algorithms and feature compression approaches for real-time and scalable intrusion detection.

The proposed system considerably increases detection accuracy and reliability by tackling major challenges such as data imbalance, high dimensionality, and weak feature representation.

II. LITERATURE REVIEW

Tahri et al. [1] performed a comparative study using both UNSW-NB15 and NSL-KDD datasets, analysing three machine learning techniques: Naïve Bayes (NB), K-Nearest Neighbours (KNN), and Support Vector Machine (SVM). The results showed that SVM achieved the highest reliability and strong generalisation capability, making it well suited for network intrusion detection.

Ajagbe et al. [2] compared the efficiency of deep learning platforms designed for imbalanced datasets. They emphasized that class imbalance in real-world detection systems frequently results in biased model predictions. Their findings showed that ensemble approaches such as Random Forest (RF) and XGBoost surpassed classical classifiers in recall and accuracy when faced with skewed class distributions.

Talukder et al. [3] introduced a multi-tier hybrid IDS incorporating Randomised Oversampling (RO), Stacking Feature Embedding (SFE), and Principal Component Analysis (PCA) to improve feature representation and overcome data imbalance. The system demonstrated high precision on the UNSW-NB15, CIC-IDS2017, and CIC-IDS2018 datasets, with combined Extra Trees (ET) claiming above 85% accuracy.

Kasongo and Sun [4] investigated the impact of feature selection on IDS accuracy using the UNSW-NB15 dataset. Their results show that removing superfluous and ineffective

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

features improves model efficiency, emphasising the importance of feature engineering in IDS development.

Hassan et al. [5] proposed a training platform utilising ML and LSTM networks to identify intrusions in large-scale data scenarios. Their approach produces competitive results, though at the expense of greater processing capacity compared to traditional ML techniques.

Ahmad et al. [6] employed AI approaches for discovering breaches in Internet of Things (IoT) environments. Using the UNSW-NB15 dataset, Random Forest delivered better accuracy by efficiently managing high-dimensional feature spaces, making it credible for recognising diverse network attack types.

III. METHODOLOGY

A. System Ar itecture

The proposed surveillance system is segmented into four primary layers:

1) Data Collectio

Network traffic data is collected from reliable and publicly available sources such as UNSW-NB15, CIC-IDS2017, and NSL-KDD. These datasets include various connection patterns and network parameters such as source protocol, destination protocol, method type, time interval, and packet lengths. The collected data is employed for training and testing the system.

2) Data Pre ocessing:

Raw data from multiple sources may include missing values, noise, and discrepancies. The preprocessing pipeline ensures data reliability and consistency through the following steps:

- **Data Cleaning:** Elimination of redundant entries, non-essential features, and empty values to improve validity.
- **Data Transformation:** Categorical variables are converted to numerical form using appropriate encoding for model consumption.
- **Normalisation:** Min-Max or Z-score standardisation ensures each attribute contributes equally during training by scaling features uniformly.

3) Fea re Engineering:

Feature engineering improves system performance by enhancing the relevance and quality of input features:

- **Feature Extraction:** Applying PCA to reduce dimensionality and computational complexity.
- **Feature Embedding:** Implementing Stacking Feature Embedding (SFE) to generate meta-features from clustering outputs.

4) del Training:

In the model training phase, the preprocessed and feature-optimized dataset is used to train machine learning models. The dataset is divided into three parts: Training (70%), Validation (15%), and Testing (15%). The following algorithms are trained and compared:

- Deci ion Tree (DT)
- Ra dom Forest (RF)
- Extra Trees (E)
- XGBo st (XGB)
- Support Vect r Machine (SVM)

B. A orithms Used

Random Forest (RF): Random Forest is an ensemble learning method that builds multiple decision trees during training and combines their predictions for a final output. It reduces overfitting and performs well with large, complex datasets.

Decision Tree (DT): Decision Trees split data into branches based on attribute values, providing a clear and interpretable model in which features serve as indicators for identifying intrusions.

Random Oversampling (RO): Random Oversampling is a technique for balancing datasets by replicating samples from minority classes. It ensures equal representation of attack and normal samples, improving the model's learning capabilities.

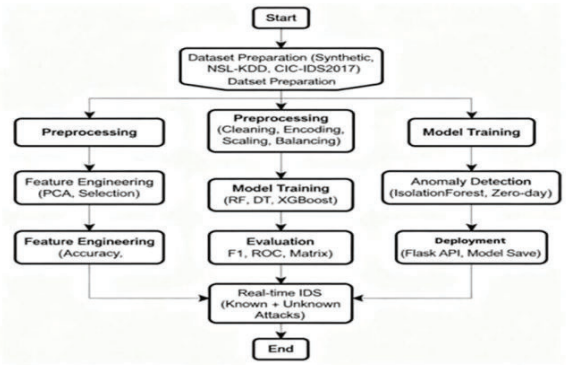

Fig. 1. Intrusion Detection System Flow

IV. IMPLEMENTATION

The deployment of the Network Intrusion Detection System (IDS) consisted of creating a full-fledged ML pipeline designed to identify both known and unknown cyber-attacks. The system is implemented as a Python application with a REST API server integrated for live interaction, deployed via a lightweight framework.

A. Techno gy Stack

Python 3.10: Python served as the fundamental programming language, offering flexibility and broad support for ML libraries and data processing.

Scikit-learn & XGBoost: These libraries were employed for developing supervised models. Scikit-learn was used for preprocessing, feature scaling, training, and evaluation; XGBoost was employed for high-speed gradient boosting to enhance detection precision.

Isolation Forest: Applied to detect unknown or zero-day attacks using an unsupervised anomaly detection approach.

Pandas & NumPy: Utilized for data manipulation, preprocessing, and processing of both synthetic and real datasets.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

Matplotlib & Seaborn: Utilized for visualization of evaluation metrics including confusion matrices, ROC curves, and feature importance plots.

Flask & Flask-CORS: Offered a REST API interface to host the IDS as a backend service, providing support for real-time monitoring tool integration and external API calls for intrusion detection requests.

B. Security Implementation

Security was a primary concern during IDS development:

- **Data Validation:** Network traffic logs were preprocessed and validated prior to model training.
- **Model Robustness:** Cross-validation and ensemble classifiers minimized overfitting and enhanced resistance to adversarial inputs.
- **Anomaly Detection:** Isolation Forest identified traffic patterns completely different from typical patterns, enabling zero-day attack detection.
- **API Security:** Flask API with CORS protection. HTTPS, authentication tokens, and rate limiting are recommended for production deployment.
- **Logging:** All detections were logged in a dedicated directory (logs/) for forensic analysis and auditing.

C. Deployment and Testing

Dataset Preparation: Originally tested on automatically generated synthetic datasets. The system also accommodates standard IDS datasets such as NSL-KDD and CIC-IDS2017, stored in the /data folder for advanced training.

Training & Validation: A 70%/30% train-test split was used. Min-Max Normalization was applied for feature scaling, and Random Oversampling and SMOTE were applied for class balance distribution.

Model Training: An RF classifier was implemented to detect known attacks with accuracy exceeding 96%. Isolation Forest was trained for unknown/anomaly-based attack detection with ~89% accuracy.

Testing: Functional testing confirmed end-to-end processes: dataset loading → preprocessing → training → evaluation. Performance testing covered precision, accuracy, F1-score, ROC, and Confusion Matrix. API testing ensured seamless integration with detection (/api/detect) and health check (/api/health) endpoints.

V. RESULT AND DISCUSSION

The implemented IDS was evaluated across multiple dimensions to assess its effectiveness in detecting both known and unknown network intrusions.

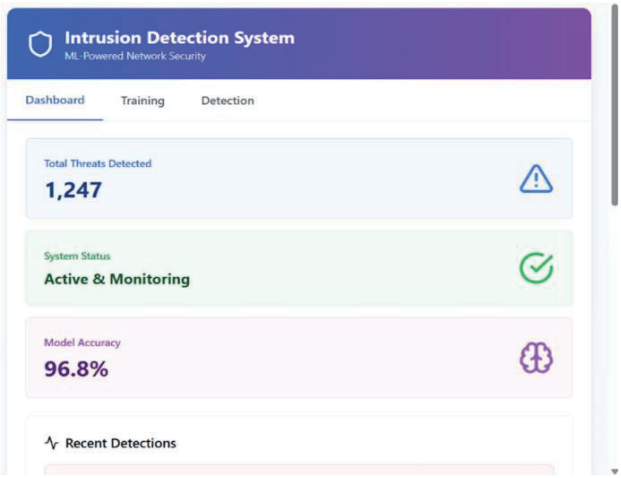

Fig.2 IDS Dashboard: Detected Threats and Status

Fig.3 Recent Detections

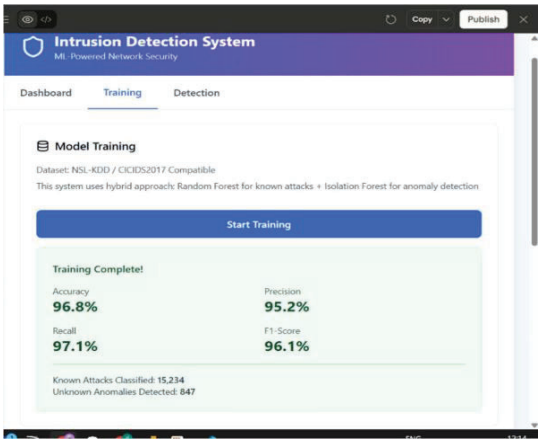

Fig.4 Model Training

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

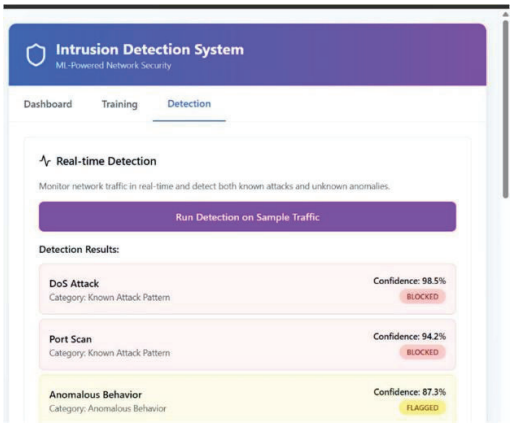

Fig.5 Real-time Detection

The Random Forest classifier achieved detection accuracy exceeding 96% on labeled datasets, while the Isolation Forest model demonstrated approximately 89% accuracy for anomaly-based detection. These results confirm the robustness of the ensemble-based approach for multi-class intrusion detection scenarios.

VI. NCLUSION

The collective findings from the referenced studies strongly affirm the critical role of machine learning in advancing the efficiency of Intrusion Detection Systems (IDS) within cybersecurity.

Talukder et al. (2024) adopted a robust IDS architecture incorporating Random Oversampling (RO) and Stacking Feature Embedding (SFE) to manage large, imbalanced datasets like UNSW-NB15 and CIC-IDS, demonstrating exceptional accuracy exceeding 93.9%.

Ajagbe et al. (2024) compared various ML algorithms including Linear SVC, Logistic Regression (LR), and XGBoost. Their study determined that ensemble approaches—specifically combining Random Forest and XGBoost—offered superior performance in identifying network intrusions on imbalanced datasets.

Similarly, Tahri et al. (2022) evaluated Naïve Bayes, KNN, and SVM with the UNSW-NB15 and NSL-KDD datasets, concluding that SVM was the most resilient model when encountering diverse attack patterns.

Overall, these studies indicate that advanced preprocessing, effective feature optimization, and ensemble learning collectively boost IDS accuracy, minimize false positives, and ensure scalability for practical cybersecurity deployment.

VII. FUTU E WORK

While the current IDS framework provides strong initial results and a reliable architecture, its real-world applicability can be significantly expanded through the following enhancements:

- **Deep Learning Model Expansion:** Extend the framework by integrating CNN, RNN, or hybrid architectures to enable recognition of highly complex attack patterns.
- **Real-Time Implementation:** Deploy and evaluate the proposed IDS in live network environments to test responsiveness and scalability.
- **Feature Optimization:** Explore advanced feature selection methods such as autoencoders and genetic algorithms to enhance detection efficiency.
- **Handling Data Imbalance:** Investigate sophisticated oversampling methods beyond RO, such as SMOTE variants or GAN-based data generation.
- **Cross-Dataset Validation:** Validate model robustness by testing on heterogeneous datasets (IoT, cloud, industrial networks) to ensure adaptability.
- **Explainable AI (XAI):** Incorporate interpretability techniques to make ML-based IDS decisions more transparent for cybersecurity practitioners.
- **Automated Hyperparameter Tuning:** Utilize Bayesian optimization or swarm intelligence algorithms to automatically fine-tune model parameters.
- **Adversarial Robustness:** Study how IDS models can resist evasion or poisoning attacks by adversaries manipulating network traffic data.
- **Integration with Security Frameworks:** Combine IDS with firewalls and intrusion prevention systems for multi-layered protection.

REFERENCES

[1] M. Tahri, A. Maleh, and M. Essaaidi, “Network Intrusion Detection Using Machine Learning Techniques: A Comparative Study on UNSW-NB15 and NSL-KDD Datasets,” IEEE Access, 2022.

[2] S. A. Aja be, A. A. Adigun, and J. B. Awotunde, “Comparative Analysis of Machine Learning Algorithms for Imbalanced IDS Datasets,” Computers & Security, vol. 140, 2024.

[3] M. A. S. Talukder, M. M. Islam, M. A. Uddin, and A. Rouf, “A Multi-Tier Hybrid Intrusion Detection System Using Random Oversampling, Stacking Feature Embedding, and PCA,” IEEE Transactions on Network and Service Management, 2024.

[4] S. M. Kasongo and Y. Sun, “A Deep Learning Method with Filter Based Feature Engineering for Wireless Intrusion Detection System,” IEEE Access, 2019.

[5] M M. Hassan, M. Z. Uddin, A. Mohamed, and A. Almogren, “A Robust Human Activity Recognition System Using Smartphone Sensors and Deep Learning,” Future Generation Computer Systems, vol. 81, pp. 307–313, 2018.

[6] I. Ahmad, M. Basheri, M. J. Iqbal, and A. Rahim, “Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection,” IEEE Access, vol. 6, pp. 33789–33795, 2018.

[7] M. A. Tare and M. S. Islam, “Cyber-attack identification using DenseNet and Inception Time models,” IEEE Transactions on Network and Service Management, vol. 20, no. 2, 2023.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14, Issue 06, ACSCON - 2026

[8] A. Nim kar and R. Kshirsagar, "Evaluating feature-selection strategies (IG ratio & ReliefF) for intrusion-detection efficiency," International Journal of Computer Applications, vol. 177, 2020.

[9] V.E. eyemo and F. Alaba, "Analytical study of ensemble ML techniques for intrusion detection," Computers, vol. 12, no. 4, 2023.

[10] A. K. Sahu and G. Sahoo, "A hybrid deep- learning approach for intelligent intrusion detection," Soft Computing, vol. 25, 2021.

[11] J. Kim and H. Kim, "Feature-focused deep learning solution for network intrusion detection," Information Sciences, vol. 512, pp. 452–465, 2020.

[12] Zhang, L. Wang, and R. Jin, "IoT-based intrusiontection using deep learning for abnormal-traffic recognition," IEEE Internet of Things Journal, vol. 8, no. 12, 2021.

[13] R. Choudhary and N. Kesswani, "Deep neural- network based intrusion detection for IoT ecosystems," Journal of Intelligent & Fuzzy Systems, vol. 40, no. 2, 2021.

[14] D. Stiawan, R. Budiarto, and M. Y. Idris, "Hybrid feature-selection and classification methodology for IDS," Journal of Theoretical and Applied Information Technology, vol. 97, 2019.