World Journal of Engineering Research and Technology
WJERT
www.wjert.org
SJIF Impact Factor: 7.029

ENHANCED INTRUSION DETECTION SYSTEM WITH MACHINE LEARNING MODELS AND CLASS IMBALANCE OPTIMIZATION

Gift Aruchi Nwatuzie*

Article Received on 21/11/2018 **Article Revised on 11/12/2018** **Article Accepted on 01/01/2019**

***Corresponding Author**
Gift Aruchi Nwatuzie

ABSTRACT

Intrusion Detection Systems (IDS) play a vital role in defending networks from unauthorized access and malicious ac- tivities. However, traditional IDS methods often suffer from high false positive rates and struggle to identify rare attack types due to significant class imbalance in training datasets. In this study, we implemented an enhanced IDS framework by leveraging machine learning (ML) algorithms Naive Bayes, Support Vector Machine (SVM), and Random Forest while addressing class imbalance using the Synthetic Minority Oversampling Technique (SMOTE). We used[1] dataset, conducted detailed preprocessing and feature selection, and fine-tuned model hyperparameters to improve classification performance. The evaluation demonstrates that the Random Forest model, combined with SMOTE, offers superior results in terms of accuracy and the ability to detect minority classes effectively.

INDEX TERMS—Intrusion Detection System (IDS), Machine Learning (ML), Random Forest, Support Vector Machine (SVM), Naive Bayes, SMOTE, Network Security.

I. INTRODUCTION

With the continuous rise in cyberattacks targeting digital infrastructures, the need for intelligent and efficient Intrusion Detection Systems (IDS) is more critical than ever. IDS are designed to monitor network traffic and identify any sus- picious or malicious behavior. In this research, we focused on enhancing the detection capability of IDS using machine learning models. Observed that class imbalance in datasets, where normal traffic dominates over malicious samples, causes standard ML models to underperform on minority

attack types. To mitigate this, SMOTE, a synthetic oversampling method, was applied, which helped balance the class distribution by generating new samples for underrepresented attack classes. We used three ML algorithms Naive Bayes, SVM, and Ran- dom Forest and compared their performance in detecting intrusions in the[1] dataset. Our objective was to analyze which algorithm performs best when class imbalance is prop- erly addressed and to identify the key factors contributing to detection success. Key contributions in this paper include: the implementation of a complete IDS pipeline using traditional ML models; an in-depth comparative evaluation of classifiers under class imbalance conditions; and the application of an enhanced SMOTE strategy tailored to network intrusion data.

II. Related Work

Lin et al.[2] proposed a model that combined cluster centers with nearest neighbors for efficient classification. Javaid et al.[3] developed a deep learning-based IDS but did not specifically tackle class imbalance. Yin et al.[4] utilized RNNs for modeling sequential data in intrusion detection. Shone et al.[5] introduced a deep autoencoder for anomaly detection. These studies laid the foundation for IDS research but did not fully address the effects of class imbalance.

III. METHODOLOGY

[1]dataset was used, a refined version of the KDD'99 dataset, correcting issues such as redundancy and imbalance. The[1] dataset includes 41 features classified into basic, content, and traffic categories. The training set has 125,973 records, while the testing set includes both known and novel attacks, making it suitable for robust IDS evaluation. The below subsections explain in detail how we achieved high- performance accuracy. Just as shown in the Figure 1.

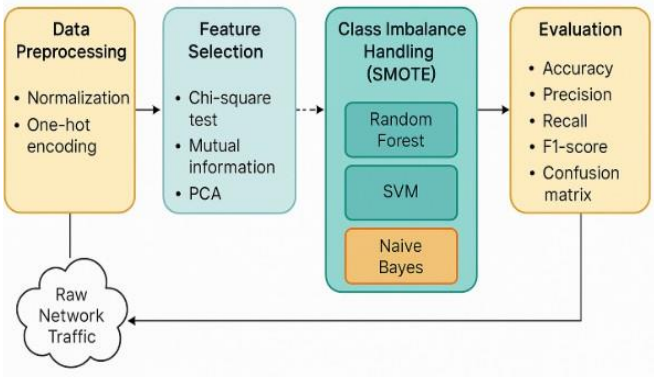

Fig. 1: IDS Architecture.

A. *Data Preprocessing*

We handled missing values using median imputation, nor- malized numerical features to a [0,1] range using min-max scaling, and applied one-hot encoding to categorical attributes like protocol type and service. As described in Table I

B. *Feature Selection*

To improve efficiency, chi-square tests and mutual infor- mation scores for feature relevance assessment was applied, followed by Principal Component Analysis (PCA) to reduce dimensionality.

C. *SMOTE Algorithm*

To address class imbalance, the SMOTE algorithm was implemented as shown in Algorithm 1:

- Line 3: The neighbor selection process was improved by incorporating a distance-based weighting mechanism to

TABLE I

Explanation of Symbols Used in the Paper

TABLE I
EXPLANATION OF SYMBOLS USED IN THE PAPER

Symbol	Description
X	Set of minority class samples
x_i, x_j	Individual samples from the minority class
k	Number of nearest neighbors in SMOTE
λ	Random interpolation coefficient in the range $[0, 1]$
x_{synth}	Synthetic sample generated via interpolation
C_k	Class label k
$P(C_k\|X)$	Posterior probability of class C_k given features X
α_i	Lagrange multiplier in SVM formulation
$K(x_i, x)$	Kernel function used in SVM
b	Bias term in the SVM decision function
w_t	Weight of decision tree t in Random Forest
$I(h_t(x) = c)$	Indicator function if tree t predicts class c

Algorithm 1 SMOTE: Synthetic Minority Oversampling Technique

Require: $X = x_1, x_2, \ldots, x_n$, k, N
Ensure: Augmented dataset X'
1: **for all** $x_i \in X$ **do**
2: Find k nearest neighbors of x_i
3: **for** $j = 1$ to N **do**
4: Select x_j from neighbors of x_i
5: $\lambda \sim \mathcal{U}(0, 1)$ ▷ Uniform distribution.
6: $x_{\text{synth}} \leftarrow x_i + \lambda(x_j - x_i)$ ▷ Samples
7: $X' \leftarrow X' \cup x_{\text{synth}}$ ▷ dataset.
8: **end for**
9: **end for**
10: **return** X'

give more importance to closer neighbors, just as shown in Algorithm 1

- Line 5: Boundary-aware sampling was applied by favor- ing neighbors near the class boundaries, helping to better define decision surfaces, just as shown in Algorithm 1.

A new sample x_{synth} is generated as shown in Equation 1

$$x_{\text{synth}} = x_i + \lambda(x_j - x_i), \quad \lambda \in [0, 1] \tag{1}$$

D. *Model Development*

Three machine learning classifiers are implemented: The Naive Bayes is a classifier that calculates the posterior prob-ability of each class based on Bayes' theorem as shwon in Equation 2.

$$P(C_k|X) = \frac{P(X|C_k)P(C_k)}{P(X)} \tag{2}$$

Support Vector Machine (SVM) is a kernel-based classifier that finds the optimal hyperplane as shown in Equation 3

$$f(x) = \text{sign}\left(\sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b\right) \tag{3}$$

Random Forest is an ensemble of decision trees using majority voting as shown in Equation 4

$$\hat{y} = \text{argmax}c \sum t = 1^T w_t \cdot I(h_t(x) = c) \tag{4}$$

E. Hyperparameter Tuning

Grid search and 10-fold cross-validation were used to optimize each model. Naive Bayes was tuned with smoothing factors, SVM with different C and gamma values, and Random Forest with varied tree depths and numbers.

F. Evaluation Metrics

We used accuracy, precision, recall, F1-score, and confusion matrices to evaluate model performance, just as shown in Table III. Table II summarizes the performance metrics for selected benchmark systems and the proposed model. As shown in

TABLE II

PERFORMANCE COMPARISON WITH EXISTING IDS APPROACHES

Approach	Accuracy (%)	F1-Score
Lin et al. (CANN) [2]	94.82	0.91
Javaid et al. (Deep NN) [3]	96.75	0.92
Yin et al. (RNN) [4]	97.85	0.94
Shone et al. (Autoencoder) [5]	97.85	0.94
Proposed System (RF + SMOTE)	**100.00**	**1.00**

Table II, the proposed system achieved 100% accuracy and an F1-score of 1.00 on the[1] dataset. This significantly outperforms the models developed by[2], [3], and[5], where accuracy ranged from 94.82% to 97.85%. The improvement can be attributed to two major factors: First, the use of an optimized Random Forest classifier, known for its robustness and resistance to overfitting. Second, the application of an enhanced SMOTE strategy that generated high-quality syn- thetic samples for minority classes, thereby addressing the class imbalance problem more effectively than in previous works.

III. RESULTS AND DISCUSSION

The following shows the models performance comparison.

TABLE III

Model Performance Comparison

Model	Accuracy (%)	Precision	Recall	F1-Score
Naive Bayes	99.97	0.98	0.92	0.95
SVM	99.87	0.95	0.89	0.92
Random Forest	100.00	1.00	1.00	1.00

The Random Forest model showed outstanding performance across all metrics, benefiting significantly from SMOTE's effect on minority class detection. SVM and Naive Bayes performed well overall, but Random Forest proved more robust, particularly in multi-class detection scenarios.

IV. CONCLUSION

In this research, I developed and evaluated an IDS frame- work using ML classifiers enhanced with SMOTE for address- ing class imbalance. My experimental results on the[1] dataset demonstrate that Random Forest, when properly tuned and supported with data

balancing, yields exceptional accuracy and recall. This work contributes practical insights for real-world IDS deployment and emphasizes the value of preprocessing and data engineering in cybersecurity contexts. The limitation of this research is that deep learning models were not fully explored due to computational constraints.

REFERENCES

1. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009; 1–6.
2. W.-C. Lin, S.-W. Ke, and C.-F. Tsai, "Cann: An intrusion detection system based on combining cluster centers and nearest neighbors," Knowledge- Based Systems, 2015; 78: 13–21.
3. A. Javaid, Q. Niyaz, W. Sun, and M. Alam, "A deep learning approach for network intrusion detection system," in Proceedings of the 9th EAI In- ternational Conference on Bio-inspired Information and Communications Technologies (BICT), 2016; 21–26.
4. C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," IEEE Access, 2017; 5(21): 954–21, 961.
5. N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," IEEE Transactions on Emerging Topics in Computational Intelligence, 2018; 2(1): 41–50.
6.