

Enhancing Intrusion Detection on Imbalanced Network Traffic Using SMOTETomek with Machine Learning and Deep Learning Classifiers

Dhanasak Bhumichai^{1*}

¹Mathematics and Computer Science Department, Navaminda Kasatriyadhiraj Royal Air Force Academy

Corresponding Author Email : dhanasak_b@rtaf.mi.th

Received 11 December 2025

Revised 25 August 2026

Accepted 12 May 2026

Abstract

The growing sophistication of cyberattacks required intrusion detection systems (IDS) capable of accurately identifying malicious network traffic, even under highly imbalanced and overlapping data conditions. Traditional machine learning models often struggled in such scenarios due to biased learning toward majority classes. This study presented a comparative evaluation of traditional machine learning and deep learning classifiers, including Decision Tree, Random Forest, K-Nearest Neighbors, XGBoost, Convolutional Neural Network, and Bidirectional Long Short-Term Memory (Bi-LSTM), for binary intrusion detection. The NF-UNSW-NB15-V2 dataset was used for training, and NF-UNSW-NB15-V3 was used for testing to ensure unbiased cross-version validation mitigate data leakage associated with random partitioning. To address class imbalance and overlapping decision boundaries, the SMOTETomek resampling technique was applied and compared against the baseline setting. Performance was assessed using standard classification metrics. Experimental results revealed that SMOTETomek-enhanced Bi-LSTM and XGBoost models achieve superior Recall and F1-score. With SMOTETomek, XGBoost achieves a Recall of 92.18% and an F1-score of 90.87%, improving by 12.87% and 2.42%, respectively. Similarly, Bi-LSTM reached a Recall of 96.87% and an F1-score of 96.65%, reflecting gains of 2.65% and 1.15%. Additionally, SMOTETomek notably improved most models, highlighting its importance in cybersecurity classification tasks. This study provided guidance on model selection for IDS under imbalanced data and demonstrated that combining deep learning with data-level balancing improved intrusion detection robustness in real-world networks.

Keywords: Machine Learning, Intrusion Detection System, Imbalanced and Overlapping Dataset, SMOTETomek Link

1. Introduction

Network intrusion detection systems (NIDS) are a fundamental component of cyber defense strategies, designed to identify unauthorized or malicious activities within network traffic. Traditional signature-based detection techniques struggle to detect novel or evolving threats, which has led to increasing adoption of machine learning (ML) and deep learning (DL) for anomaly- and behavior-based intrusion detection [1], [2]. However, a common challenge in applying ML and DL to NIDS is the class imbalance inherent in real-world network traffic datasets. Attack traffic often accounts for only a small proportion of total traffic, while benign connections dominate. This imbalance can significantly bias classifiers toward the majority class, resulting in poor detection rates for the minority (attack) class, which is the most critical to identify [3]. Moreover, overlapping feature distributions between benign and malicious flows further complicate model training, leading to ambiguous decision boundaries and reduced generalization [4].

To address these issues, recent studies have employed various resampling techniques like SMOTE (Synthetic Minority Over-sampling Technique), undersampling, and hybrid methods to mitigate imbalance [5], [6]. Among these, SMOTETomek Link has emerged as a promising hybrid approach that both boosts minority class representation and removes noisy or borderline majority samples, thereby enhancing class separability and improving classifier performance [7]. In addition to data-level techniques, choice of classifier is also critical. Ensemble methods such as Random Forest and XGBoost have demonstrated strong performance in network security tasks due to their robustness and ability to handle high-dimensional data [8], [9]. Meanwhile, deep learning models, particularly Convolutional Neural Networks (CNN) and Bi-directional Long Short-Term Memory (LSTM), have shown promise by learning intricate spatial and temporal features from network flows [10], [11].

In this paper, we propose a comprehensive experimental framework to evaluate six classifiers—Random Forest, Decision Tree, K-Nearest Neighbors (KNN), XGBoost, CNN, and Bi-directional LSTM—on the NF-UNSW-NB15-V2 and NF-UNSW-NB15-V3 datasets. Such an evaluation strategy ensures unbiased cross-version validation while mitigating potential data leakage introduced by random splitting approaches. We perform two experiments: a baseline without any resampling and an enhanced setup using SMOTETomek Link to address imbalance and feature overlap. Our evaluation metrics include Accuracy, Precision, Recall, F1-score, and execution time, providing a balanced assessment between detection performance and computational feasibility.

The contributions of this work are as follows:

1. A systematic comparison of classical machine learning and deep learning models on the NF-UNSW-NB15 dataset under both balanced and imbalanced conditions, using a cross-version training–testing protocol to ensure robust and unbiased evaluation.
2. The implementation and empirical evaluation of a hybrid SMOTETomek technique for intrusion detection, analyzing its impact on classification performance, sensitivity to minority classes, and overall model robustness.

3. A quantitative analysis of the trade-off between detection performance and computational cost, providing practical insights for real-world deployment of network intrusion detection systems. A practical analysis of trade-offs between performance and computational cost, informing deployment choices for real-world NIDS systems.

The rest of this paper is organized as follows. Section 2 presents related work. Section 3 describes the research methodology. Section 4 reports experimental results. Section 5 discusses the findings. Finally, Section 6 concludes and outlines, while the future work is presented in Section 7.

2. Related Work

2.1 Machine learning and deep learning for intrusion detection

The adoption of ML and DL for network intrusion detection has accelerated in recent years due to the need for behavior- and anomaly-based detection capable of identifying novel attacks. Several studies demonstrated that deep architectures (CNNs, LSTMs, Bi-LSTM, and hybrid CNN–Bi-LSTM models) could learn richer representations from flow-based features and raw traffic traces, often outperforming classic ML models on large-scale, modern datasets. For example, ML algorithms like Decision Tree (DT), K-nearest neighbor (KNN), Random Forest (RF), and eXtreme Gradient Boosting (XGBoost) could be deployed to effectively detect malicious traffic [12], [13], [14], [15], [16]. In addition, convolutional and recurrent models have been applied successfully to detect IoT and general network attacks, showing strong capability to capture spatial and temporal patterns in flows [17], [18]. Another recent article described Bi-LSTM models excel at modeling temporal/sequential dependencies, which was useful when network traffic exhibited time-dependent patterns (e.g. sequential bursts, slow scanning, IoT device behavior over time). These models could thus detect anomalies that might be invisible to static-feature classifiers [19].

2.2 The UNSW / NF-UNSW-NB15 datasets and challenges

Research on modern intrusion detection frequently used the UNSW/UNSW-NB15 family of datasets because they contained realistic and contemporary attack scenarios, a large number of flows, and many flow-level features that reflected modern traffic behaviors. The NF-UNSW-NB15 variants (v2, v3) contained dozens of features and multiple attack types, but they also presented severe class imbalance and overlapping feature distributions that complicate classifier learning and evaluation. Using NF-UNSW-NB15 (and its variants) as a benchmark has therefore become common practice for evaluating both classical and deep learning IDS approaches [20], [21].

2.3 Imbalanced learning and hybrid resampling techniques

Class imbalance was a major challenge in intrusion detection research. In real-world network traffic, normal connections vastly outnumbered attack instances, and even within attack traffic, certain types-especially rare or emerging ones were significantly under-represented. Training classifiers directly on such skewed data often yielded models with seemingly high overall accuracy but poor detection of minority (attack) classes, resulting in low recall and F1 scores for the most critical intrusions. To mitigate this issue, data-level resampling strategies were widely used to balance

class distributions before training. One of the most prominent methods was the Synthetic Minority Oversampling Technique (SMOTE), which generated artificial minority samples based on feature-space similarities rather than simply duplicating existing ones. This approach improved the model's ability to learn meaningful patterns of rare attack behaviors and enhances minority-class detection performance [22]. Studies using SMOTE on the KDD99 dataset have demonstrated significant performance improvements when applied with traditional machine learning algorithms, without compromising majority-class accuracy [23].

However, oversampling alone may introduce challenges such as class overlap, noisy samples, and potential overfitting-especially near decision boundaries. To address these drawbacks, hybrid resampling approaches combined SMOTE with cleaning or undersampling methods to enhance data quality. For example, SMOTE-Tomek applied SMOTE to generate synthetic minority samples and then removed borderline or ambiguous majority-class instances using Tomek Links. This process reduced class overlap, sharpens decision boundaries, and decreases false positive rates, leading to more reliable and robust intrusion detection performance [24], [25], [26].

As a result, hybrid resampling strategies that integrated SMOTE-based oversampling with undersampling or data-cleaning techniques have gained considerable momentum in NIDS research. These methods effectively handled both class scarcity and sample noise, improving the detection of rare attack types while preserving strong performance on majority classes.

2.4 Convolutional neural network for IDS

Convolutional Neural Networks (CNNs) are a type of neural network that is originally used in image and video recognition tasks [27]. In recent years, CNNs have also been widely adopted to enhance Intrusion Detection System (IDS) solutions due to their strong capability in automatic feature extraction and pattern recognition [28]. The CNN architecture consists of input layer, convolutional layer, pooling layer, and output layer [29] that are specialized to extract different features from the input data. The CNN model structure is illustrated in Figure 1.

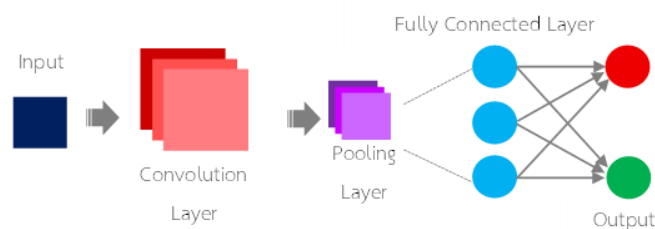


Figure 1 Fundamental Design of Convolutional Neural Network (CNN) architecture.

CNNs can be integrated into Intrusion Detection Systems (IDS) by utilizing their capacity to automatically extract meaningful spatial features from raw network traffic data. In this setting, network traffic is modeled as time-series or structured data, enabling CNNs to identify underlying patterns that distinguish normal activity from malicious behavior [30], [31]. First, the input layer receives the data packet and converts it into a format suitable for the network. The convolutional layer then applies several filters to extract key features, such as patterns or anomalies. This produces

feature maps that highlight important parts of the data. Next, the pooling layer reduces the size of these feature maps, which lowers computational cost and improves robustness to noise and small variations. The fully connected layer then processes the extracted features to make a classification decision. Finally, the output layer applies an activation function, such as softmax or sigmoid, to generate class probabilities and classify the traffic as normal or malicious.

2.5 Bi-LSTM for IDS

A Bidirectional LSTM (Bi-LSTM) [32] is an advanced version of the standard LSTM model that can learn information from both forward and backward directions in a sequence. LSTM networks are widely used for sequence prediction because they can capture long-term dependencies. However, a traditional LSTM processes data in only one direction, from beginning to end, which may limit its understanding when future information is also important. Bi-LSTM overcomes this limitation by using two LSTM layers: one processes the sequence forward (start to end), and the other processes it backward (end to start). The architecture of the Bi-LSTM is shown in Figure 2. Recent work applying Bi-LSTM to NF-UNSW and other network datasets reports notable improvements in recall and F1, especially when combined with preprocessing and resampling techniques that reduce imbalance and noise. Nevertheless, these models typically demand more computational resources and longer training times than tree-based ensembles [33].

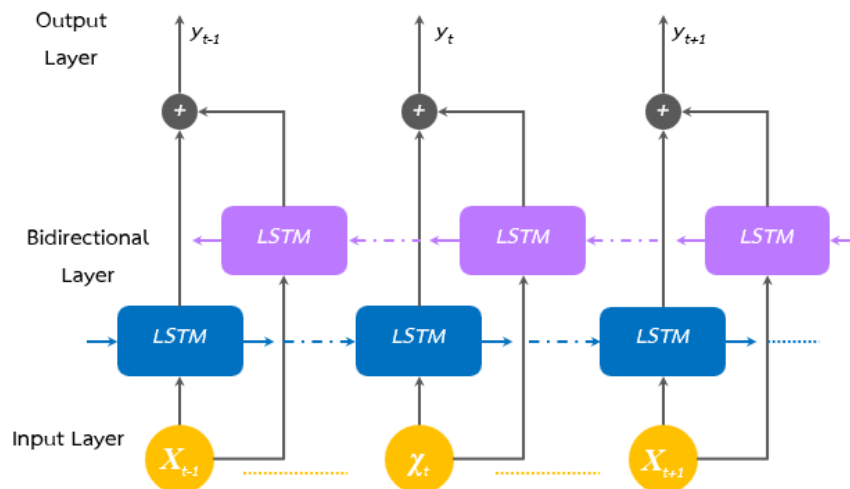


Figure 2 The architecture of Bidirectional Long Short-Term Memory (Bi-LSTM).

2.6 Gaps and how this work differs

While numerous studies have employed SMOTE, Tomek Link, or their combined variants to mitigate class imbalance, relatively few have conducted a systematic and controlled comparison across both traditional machine learning and deep learning classifiers using consistent NF-UNSW-NB15 dataset variants as training and testing splits. Moreover, detailed analyses of runtime implications are often absent, despite their importance for real-world deployment. This study addresses these gaps by: (i) performing a direct comparative evaluation of Random Forest, Decision Tree, KNN, XGBoost,

CNN, and Bi-LSTM under both baseline (no resampling) and SMOTETomek conditions on NF-UNSW-NB15-V2 and V3; (ii) reporting comprehensive class-level performance metrics—precision, recall, and F1-score—alongside execution time to inform deployment-oriented decision-making; and (iii) examining the influence of SMOTETomek on overlapping decision boundary regions and model sensitivity within a large-scale, realistic intrusion detection benchmark. This work aligns with and addresses the practical need highlighted in recent reviews for reproducible, performance- and cost-aware evaluations of resampling strategies on modern IDS datasets [34].

3. Research Methodology

This section presents the methodology for detecting malicious network traffic using traditional machine learning and deep learning classifiers. The methodology consists of five main stages: dataset preparation, preprocessing, handling data imbalance and overlapping samples, model development, and performance evaluation. Figure 3 illustrates the proposed model.

3.1 Dataset Preparation

In this study, the NF-UNSW-NB15 dataset is selected due to its comprehensive coverage of modern cyberattack types and its realistic representation of network traffic. The dataset consists of two subsets: NF-UNSW-NB15-V2, which is used for training, and NF-UNSW-NB15-V3, which is used for testing. Each network flow record includes 41 features, encompassing basic connection attributes (e.g., duration, protocol), content-level features (e.g., payload and header information), and time-based behavioral characteristics. The original dataset comprises eight classes: Analysis, Backdoor, Benign, DoS, Exploits, Fuzzers, Generic, and Reconnaissance. To align with practical intrusion detection system (IDS) scenarios, the multiclass classification problem is converted into a binary format (Benign vs. Attack). Table 1 presents the class distribution of the dataset.

3.2 Data Preprocessing

Several preprocessing steps are applied to enhance data quality and ensure compatibility with machine learning algorithms. First, instances containing missing or corrupted values are removed to prevent noise and potential bias. Next, categorical features such as protocol type and service are transformed using label encoding or one-hot encoding to make them interpretable by machine learning models. Numerical features are then normalized using StandardScaler to ensure consistent feature scales and improve model convergence. The NF-UNSW-NB15 V2 dataset is used exclusively for model training, while V3 is reserved for independent testing. All preprocessing steps, including categorical encoding and feature scaling, are fitted solely on the V2 training data. The learned transformation parameters are subsequently applied to both V2 and V3. SMOTETomek is performed only on the preprocessed V2 training set. The V3 dataset remains completely unseen during preprocessing and resampling to prevent data leakage. Hyperparameters are predefined and fixed throughout the experiments, and no cross-validation or grid search was conducted.

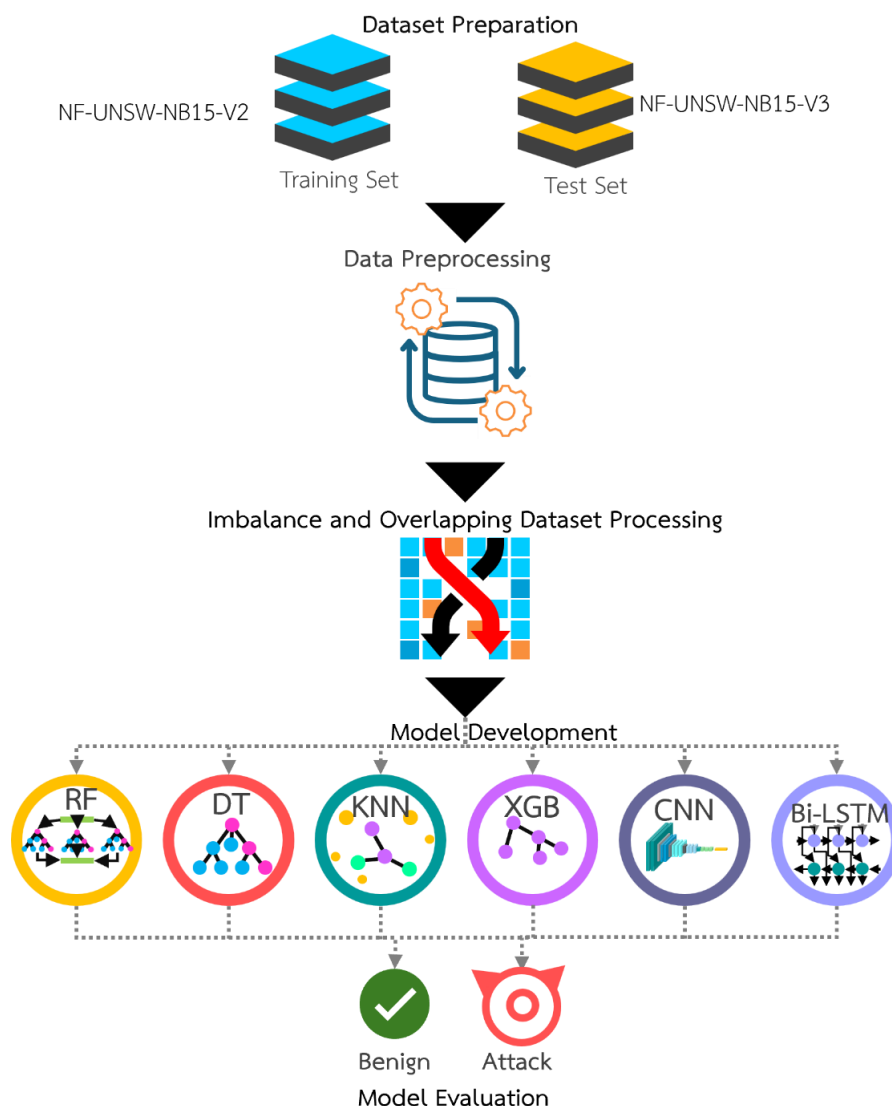


Figure 3 Balanced Intrusion Detection Framework (BIDF) Using SMOTETomek and Hybrid Classifiers.

Table 1 Class Distribution of the NF-UNSW-NB15 Dataset.

Original Classes	Final Class
Benign	Benign (Normal Traffic)
Analysis, Backdoor, DoS, Exploits, Fuzzers, Generic, Reconnaissance	Attack (Malicious Traffic)

3.3 Addressing Data Imbalance and Overlapping Classes

The NF-UNSW-NB15 dataset exhibits significant class imbalance, where benign traffic vastly outnumbers malicious traffic. Additionally, certain attack types, especially rare or emerging ones, are severely under-represented. Furthermore, the presence of overlapping samples between benign and malicious traffic introduces ambiguity near the decision boundary. These two issues cause machine learning models to become biased toward the majority (benign) class and increasing misclassification intrusion detection reliability. To mitigate these issues, the SMOTE-Tomek Link resampling technique

is employed in the second experiment. This hybrid method integrates both oversampling and undersampling to improve class representation and refine data quality. Specifically, the SMOTE generates synthetic minority class samples by interpolating between existing minority instances within feature space, thereby increasing the diversity of attack patterns without duplicating data. However, oversampling alone may introduce noisy or overlapping instances, especially near the class boundary. To address this, Tomek Link undersampling is applied immediately after SMOTE. Tomek Links identify pairs of samples from different classes that are very close to each other, indicating potential overlap or noise. By removing the majority-class samples from these pairs, Tomek Links effectively clean the dataset, reduce class overlap, and sharpen the decision boundary. This hybrid approach not only balances the dataset but also enhances data separability, ensuring that classifiers can better learn the distinguishing characteristics of minority attack types. As a result, models trained on SMOTE-Tomek processed data demonstrate improved detection performance, particularly in terms of recall and F1-score for minority intrusions, while still maintaining strong accuracy for the majority class.

3.4 Model Development

To comprehensively evaluate intrusion detection performance across varying algorithmic paradigms, both traditional machine learning and deep learning models are implemented in this study. The selected models offer complementary strengths, ranging from low computational complexity and ease of deployment to enhanced capability in learning complex patterns and sequential dependencies. Table 2 outlines the algorithms employed, categorized by model type and associated characteristics.

Table 2 Implemented Classification Models and Their Key Attributes.

Model Type	Algorithms Used	Notes
Traditional Machine Learning	Random Forest, Decision Tree, KNN, XGBoost	Low computational cost, easier deployment
Deep Learning	CNN, Bi-LSTM	Better for feature learning and sequential data

- Random Forest: An ensemble model that avoids overfitting through feature averaging.
- Decision Tree: Interpretable, but sensitive to noisy and overlapping data.
- KNN: Instance-based learner, affected by class distribution.
- XGBoost: Gradient boosting-based model optimized for speed and accuracy.
- CNN: Captures spatial and relational features from structured data.
- Bi-LSTM: Handles temporal dependency and sequential network flow behavior.

Hyperparameters are selected based on preliminary testing. Several values are evaluated, and the configuration that achieves the best Recall and F1-score is chosen and fixed for the final experiments. Table 3 presents the key hyperparameters configured for each model in the experiment.

Table 3 Hyperparameter Configuration of All Implemented Models.

Model	Key Hyperparameters	Value
Decision Tree	Criterion	Entropy
	Max Depth	None
	Min Samples Split	2
	Min Samples Leaf	1
	Max Features	None
	Random State	Not specified
Random Forest	Number of Trees (n_estimators)	860
	Criterion	Entropy
	Max Depth	None
	Min Samples Split	2
	Min Samples Leaf	1
	Max Features	sqrt
	Random State	42
KNN	Number of Neighbors (k)	5
	Weights	Uniform
	Distance Metric	Minkowski (p=2)
	Effective Distance	Euclidean
	Algorithm	Auto
XGBoost	n_estimators	100
	max_depth	6
	learning_rate	0.3
	subsample	1.0
	colsample_bytree	1.0
	gamma	0
	reg_lambda	1
	eval_metric	mlogloss
	use_label_encoder	False
CNN	Conv1D-1 Filters	32
	Kernel Size	3
	Padding	Same
	Conv1D-2 Filters	64
	Kernel Size	2
	Dense Units	128
	Output Units	1
	Activation (Hidden)	ReLU
	Activation (Output)	Sigmoid

Model	Key Hyperparameters	Value
Bi-LSTM	Trainable Parameters	Depends on input size
	LSTM Units	50
	Bidirectional	Yes
	Dense Units	128
	Output Units	1
	Activation (Hidden)	ReLU
	Activation (Output)	Sigmoid
	Total Trainable Parameters	33,857

Although NF-UNSW-NB15 consists of tabular flow-based records, each sample can be represented as a one-dimensional ordered feature vector of 41 attributes. 1D-CNN architecture is able to capture local feature interactions through convolutional filters operating along the feature dimension. Unlike traditional machine learning models that treat features independently, the convolutional operation enables the model to learn local dependency patterns and higher-order feature combinations automatically. This design allows the network to extract discriminative representations from correlated traffic features without requiring manual feature engineering.

The proposed 1D Convolutional Neural Network (CNN) architecture proposed in this study consists of sequential layers designed for binary classification of network traffic data. The model begins with an input layer of shape $(L, 1)$, where L represents the number of input features. The input is passed to the first Conv1D layer with 32 filters, kernel size of 3, ReLU activation, and same padding, followed by a MaxPooling1D layer with a pool size of 2 and same padding. Next, a second Conv1D layer with 64 filters and kernel size of 2 (ReLU activation, same padding) is applied, followed by another MaxPooling1D layer with pool size 2. The extracted feature maps are then flattened using a Flatten layer and fed into a fully connected (Dense) layer with 128 neurons and ReLU activation. Finally, the network ends with a Dense output layer containing 1 neuron with a sigmoid activation function, producing the probability for binary classification (normal or malicious traffic). Figure 4. Architecture of the Proposed 1D-CNN Model

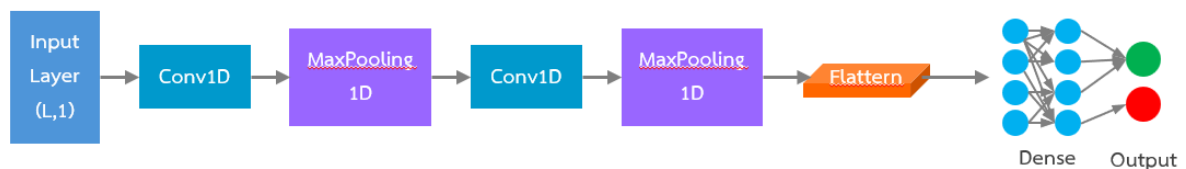


Figure 4 Proposed 1D Convolutional Neural Network (CNN) architecture.

The proposed Bi-LSTM model is designed for binary classification of network traffic data. It starts with an input layer of shape $(L, 1)$, where L is the number of features. The input is then processed by a Bidirectional LSTM layer with 50 units, which analyzes the sequence in both forward and backward directions. Only the final hidden states are used as output. This output is passed to a fully connected layer with 128 neurons and ReLU activation. Finally, a single neuron with a sigmoid activation function produces the probability of the traffic being normal or malicious. Figure 5 illustrates the architecture of the Proposed Bi-LSTM model.

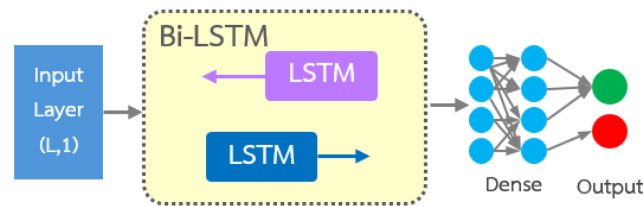


Figure 5 Proposed Bi-LSTM architecture.

While NF-UNSW-NB15 is not inherently a time-series dataset, each flow instance is represented as a sequence of ordered features (sequence length = 41). This pseudo-sequential representation enables the Bi-LSTM to model inter-feature dependencies by treating the feature vector as an ordered signal. The bidirectional structure allows the model to capture dependencies in both forward and backward directions across the feature space, thereby learning complex feature interactions beyond linear correlations. We acknowledge that this representation does not correspond to temporal dynamics; rather, it serves as a mechanism for structured feature dependency modeling.

3.5 Representation of CNN and Bi-LSTM

Each network flow instance in the NF-UNSW-NB15 dataset consists of 41 numerical features and is represented as a fixed-length feature vector $\mathbf{x} \in \mathbb{R}^{41}$. For the 1D-CNN model, the input is reshaped into a one-dimensional tensor of shape $(41, 1)$, where convolution is performed along the feature dimension. No 2D reshaping is applied, as the dataset does not possess spatial structure. The convolutional filters operate as sliding windows over adjacent features, enabling the model to learn localized interaction patterns among correlated traffic attributes (e.g., flow duration, packet size statistics, byte rates). This design allows hierarchical extraction of higher-order feature combinations while preserving the original feature ordering.

For the Bi-LSTM model, each instance is reshaped into a pseudo-sequential representation of length 41, where each feature dimension is treated as one step in the sequence. No grouping by session, flow ordering is applied. Thus, the sequence does not represent temporal evolution across multiple flows but rather an ordered traversal of the feature vector. The bidirectional recurrent structure enables modeling of inter-feature dependencies across the entire feature space, capturing nonlinear contextual relationships in both forward and backward directions. We acknowledge that this formulation constitutes a pseudo-sequence rather than a true time-series structure. Therefore,

the Bi-LSTM should be interpreted as modeling structured feature dependencies rather than temporal dynamics of network traffic.

3.6 Model Evaluation

The confusion matrix is a fundamental evaluation tool in supervised machine learning. It provides a structured representation of the relationship between predicted labels and their corresponding ground-truth labels [35]. By organizing classification outcomes into a tabular format, the confusion matrix enables the derivation of various quantitative performance metrics. The primary terminologies associated with the confusion matrix include:

True Positive (TP): Data samples correctly predicted as positive.

True Negative (TN): Data samples correctly predicted as negative.

False Positive (FP): Data samples incorrectly predicted as positive.

False Negative (FN): Data samples incorrectly predicted as negative.

To assess the performance and practical applicability of the evaluated classifiers, multiple evaluation metrics are employed, each capturing different aspects of intrusion detection capability. These metrics not only reflect predictive accuracy but also emphasize the ability to correctly identify minority attack classes and the computational efficiency crucial for real-time IDS deployment. Table 4 summarizes the definitions and relevance of each metric used in this study.

Recall and F1-score are emphasized due to their importance in intrusion detection, where false negatives (missed attacks) are more critical than false positives.

Table 4 Evaluation metrics used in this study.

Metric	Description
Accuracy	Overall correctness of predictions
Precision	Correctly identified attack samples among predicted attacks
Recall	Correctly detected attack samples among actual attacks (most critical for IDS)
F1-score	Harmonic mean of precision and recall
Training & Testing Time	Computation cost and real-time applicability

4. Experimental Result

This section presents the results of two experiments conducted to classify malicious network traffic using six machine learning and deep learning classifiers. The objective is to evaluate their effectiveness in detecting cyberattacks, both before and after applying the SMOTETomek Link technique to address data imbalance. The experiment consists of Baseline and SMOTETomek Link experiment

4.1 Baseline Experiment (Without SMOTETomek)

In the first experiment, the models are trained on the original imbalanced dataset. The results in Table 5 show that the Random Forest, CNN, and Bi-LSTM models achieve high accuracy,

with CNN achieving the highest overall accuracy of 99.41%, followed by Random Forest (98.97%) and XGBoost (98.15%). However, the imbalance in the dataset affects the performance of some models, particularly those sensitive to class imbalance. While accuracy remained high, a closer inspection of the recall for the minority (attack) class reveals significant differences. For instance, CNN achieves a recall of 94.13%, whereas Random Forest achieves 88.50%, indicating that CNN is more effective in detecting malicious traffic. Conversely, Decision Tree and KNN achieve relatively lower recall rates (62.00% and 92.00%, respectively), suggesting underperformance in detecting attacks despite their reasonable accuracy. The deep learning models, CNN and Bi-LSTM, outperformed traditional machine-learning models, particularly in recognizing complex attack patterns. Bi-LSTM achieved 99.15% accuracy with strong precision and F1-scores, demonstrating its ability to capture temporal dependencies in network traffic.

Table 5 Summary of Classifier Performance without SMOTETomek Link.

Model	Accuracy Rate (%)	Precision	Recall	F1-Score	Training Time (ms)	Testing Time (ms)
Random Forest	98.97	99.99	88.50	93.90	1801467	42297
Decision Tree	94.92	75.97	62.94	68.84	10124	215
KNN	96.02	71.63	91.71	80.44	629	5962701
XGBoost	98.15	99.97	79.31	88.45	17261	935
CNN	99.41	99.21	94.13	97.42	428	89
Bi-LSTM	99.15	96.30	94.22	95.50	2625	387

Among the six classifiers, CNN demonstrates the most effective performance. To further evaluate its classification capability, the confusion matrix of the best-performing model (CNN) is presented in Figure 6. The model correctly classified 954,364 negative samples and 88,022 positive samples, with only 701 false positives and 5,488 false negatives. The low number of false positives confirms the high precision of the model, while the relatively higher number of false negatives explains why recall is slightly lower than precision. Overall, the results indicate stable and reliable classification performance.

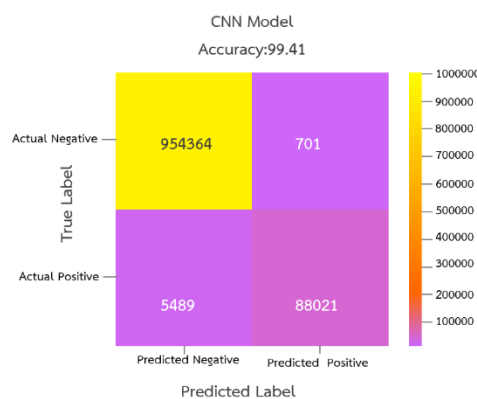


Figure 6 Confusion matrix of CNN model on imbalanced NF-UNSW-NB15 dataset.

4.2 SMOTETomek Experiment (With Imbalanced Handling)

In the second experiment, the SMOTETomek Link mechanism is introduced to address data imbalance by combining oversampling and noise reduction. The results in Table 6 demonstrate significant improvement, especially in the attack class recall and F1-scores. Random Forest improves notably, with accuracy increasing from 98.97% to 99.43%, and attack-class recall increasing from 88.50% to 94.00%. Similarly, XGBoost shows substantial improvement, with recall increasing from 79.31% to 92.18%, and overall accuracy rising to 98.35%. This indicates that SMOTETomek effectively reduces the minority class bias and enhances attack detection capabilities.

Machine learning models that previously struggled-particularly Decision Tree and KNN-also benefit from SMOTETomek. KNN recall improves from 91.71% to 93.97%, indicating stronger attack detection. Although Decision Tree's overall accuracy slightly declines, its recall improves dramatically from 62.94% to 86.54%, highlighting improved sensitivity to malicious traffic. The deep learning models (CNN and Bi-LSTM) continue to show superior performance. CNN achieves the highest accuracy of 99.53%, with precision, recall, and F1-score all above 97.00%, confirming its robustness. Bi-LSTM maintains high detection performance with 99.21% accuracy, and a balanced recall of 96.00% for the attack class.

Table 6 Summary of Classifier Performance with SMOTETomek Link.

Model	Accuracy Rate (%)	Precision	Recall	F1-Score	Training Time	Testing Time
Random Forest	99.43	99.95	93.62	96.68	17370896	60202
Decision Tree	94.10	62.17	86.54	72.36	168816	869
KNN	95.68	68.92	93.97	79.52	925	11861653
XGBoost	98.34	89.59	92.18	90.87	31964	657
CNN	99.53	99.21	95.43	97.52	601	69
Bi-LSTM	99.21	97.56	96.87	96.65	4346	426

These results confirm that addressing both class imbalance and class overlapping is essential for network intrusion detection. SMOTETomek Link successfully enhances both minority representation and class separability, leading to better overall detection of benign traffic - a crucial factor in real-world scenarios where false positives can disrupt normal operations.

Among the six evaluated classifiers, CNN achieved the best overall performance. To further assess its classification effectiveness, the confusion matrix of the top-performing model is presented in Figure 7. The results show that the model correctly classified 954,354 normal samples and 89,256 attack samples. Only 711 normal instances were misclassified as attacks (false positives), while 4,254 attack samples were incorrectly identified as normal (false negatives).

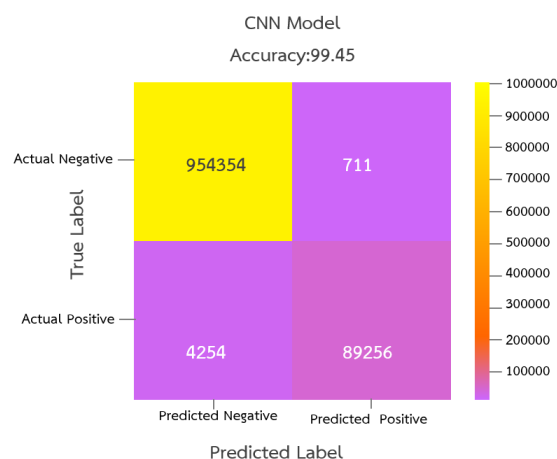


Figure 7 Confusion matrix of CNN model on balanced NF-UNSW-NB15 dataset.

4.3 Comparative Analysis

The comparative results presented in Figure 8 demonstrates the substantial impact of applying SMOTETomek Link on the performance of various classifiers when dealing with imbalanced and overlapping datasets. As indicated in Table 1, the original dataset suffers from class skew, causing several models to achieve relatively high accuracy while underperforming in Recall and F1-Score, particularly for minority classes. This behavior is consistent with observations in prior studies, which established that accuracy is often misleading in imbalanced classification tasks due to the dominance of majority classes [7]. Furthermore, overlapping class distributions significantly degraded classifier sensitivity and increased false negatives, resulting in poor generalization.

Upon applying SMOTETomek Link, the performance of all classifiers improves notably, especially in metrics sensitive to class imbalance such as Recall and F1-Score. SMOTE effectively generates synthetic minority samples to balance class representation, while Tomek Link removes borderline and noisy samples, enhancing class separability and reducing data-level overlap [7]. These improvements are especially evident in the performance of ensemble-based techniques. XGBoost exhibits the most significant enhancement, achieving the highest F1-Score, Accuracy, and Recall across all experiments. This confirms its capability to learn complex patterns and effectively handle noise-reduced and well-balanced data [36]. Similarly, Random Forest shows enhanced robustness and consistency due to its ability to aggregate diverse decision trees trained on cleaner and better-distributed samples. In contrast, KNN and Decision Tree demonstrated only moderate improvements. Since KNN relies heavily on local proximity, the removal of boundary points sometimes eliminated useful discriminative information, slightly affecting its generalization. Meanwhile, Decision Tree remained prone to overfitting, although its Recall improved due to increased minority class exposure after SMOTE.

Overall, the results confirm that SMOTETomek Link not only enhances minority class recognition but also improves classifier stability and robustness in overlapping and imbalanced settings. Ensemble-based models, particularly XGBoost and Random Forest, emerge as the most

effective classifiers due to their strong capacity to handle synthetic and noise-reduced data while maintaining structural generalizability.

The results confirm that handling data imbalance is crucial for improving attack detection in cybersecurity datasets. While traditional machine-learning models show significant improvements with SMOTETomek, deep learning models like CNN and Bi-LSTM consistently outperform all other classifiers, both before and after using the imbalance-handling technique.

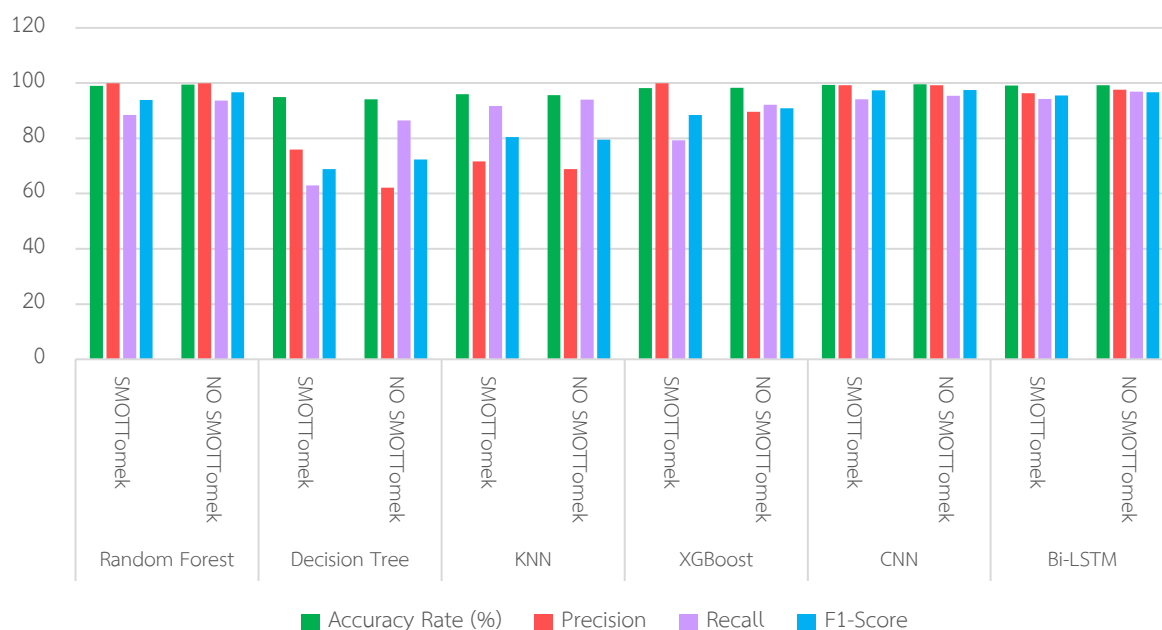


Figure 8 Performance comparison of classifiers before and after applying SMOTETomek.

4.4 Trade-Off Between Performance and Computational Cost

To analyze the trade-off between performance and computational cost, Tables 5 and 6 were compared. SMOTETomek consistently improves Recall and F1-score across most classifiers, particularly for models sensitive to class imbalance. Random Forest shows an increase in Recall from 88.50% to 93.62% and an improvement in F1-score from 93.90% to 96.68%. Decision Tree benefits substantially, with Recall rising from 62.94% to 86.54%. XGBoost also demonstrates notable improvement, with Recall increasing from 79.31% to 92.18%. CNN and Bi-LSTM exhibit moderate but consistent gains. These findings confirm that SMOTETomek effectively enhances minority-class detection performance.

However, the computational impact varies significantly across models. The most pronounced change is observed in Random Forest, where training time increases from 1,801,467 ms to 17,370,896 ms, representing nearly a tenfold rise. This substantial overhead is likely due to the enlarged dataset produced by SMOTETomek, which increases the complexity of constructing multiple decision trees. Although performance improves, such a dramatic increase in training cost may limit its practicality in environments requiring frequent retraining. KNN exhibits a different trade-off pattern. While its training time remains low, testing time increases from 5,962,701 ms to 11,861,653

ms. Since KNN performs distance calculations against all training samples during inference, the expanded dataset significantly amplifies its prediction cost, making real-time deployment more challenging. In contrast, XGBoost and CNN demonstrate a more balanced trade-off. XGBoost's training time increases moderately from 17,261 ms to 31,964 ms while achieving substantial performance gains. CNN shows only a slight increase in training time (428 ms to 601 ms) and even a small reduction in testing time, indicating strong scalability under resampling conditions. Bi-LSTM also maintains high detection performance with a moderate increase in training time (2,625 ms to 4,346 ms), suggesting acceptable computational growth.

5. Discussion

The experimental results reveal several key insights into the performance of traditional machine learning and deep learning classifiers for intrusion detection using the NF-UNSW-NB15 dataset. Across both experiments (with and without SMOTETomek Link), it is evident that data imbalance and overlapping class boundaries critically affect classifier performance, particularly in correctly identifying the minority attack class.

5.1 Impact of Imbalanced and Overlapping Data

In the baseline experiment without SMOTETomek Link, classifiers such as Random Forest and XGBoost achieve high overall accuracy (98.97% and 98.15%, respectively). However, the Recall for the attack class is significantly lower (Random Forest Recall: 88.50%, XGBoost Recall: 79.31%), highlighting an inability to correctly detect malicious traffic. This is primarily due to severe class imbalance where benign samples dominate the dataset, causing classifiers to bias toward the majority class. Additionally, overlapping feature distributions between benign and attack traffic further reduced the models' discriminatory capability.

The introduction of SMOTETomek Link led to substantial improvements in detecting attack samples. By synthetically augmenting the minority class and simultaneously removing borderline overlapping samples, SMOTETomek significantly enhanced Recall across most models. Random Forest's Recall increased from 88.50% to 93.62%, while XGBoost's Recall improved from 79.31% to 92.18%. This confirms that simultaneously addressing class imbalance and feature overlap is essential for improving intrusion detection performance.

5.2 Traditional vs. Deep Learning Models

Traditional models such as Decision Tree and KNN show mixed performance. Decision Tree, despite its interpretability and fast computation, struggles with overlapping feature distributions, leading to unstable minority class detection (Recall improves from 62.94% to 86.54% after SMOTETomek, but overall performance remains lower compared to other models). KNN, on the other hand, benefits significantly from SMOTETomek due to improved neighborhood structure, but suffers from extremely high inference time, making it unsuitable for real-time intrusion detection systems.

Deep learning models-CNN and Bi-LSTM-consistently outperforms traditional models, especially after the SMOTETomek Link integration. CNN achieves the highest overall accuracy (99.53%) and strongly improves Recall for the attack class (from 94% to 95%). Its ability to extract spatial feature patterns from network flows made it highly effective for intrusion detection. Bi-LSTM also demonstrates strong performance (99.21% accuracy), particularly in learning temporal dependencies in packet sequences, although its training time is considerably higher due to its recurrent architecture.

The experimental results demonstrate that class imbalance and overlapping feature distributions significantly influence intrusion detection performance. While most classifiers achieve high overall accuracy, minority-class detection is initially limited due to bias toward the dominant benign class. The integration of SMOTETomek effectively mitigates these issues by both oversampling the minority class and cleaning overlapping boundary samples, resulting in consistent improvements in minority detection capability. However, beyond performance metrics, the computational implications of SMOTETomek vary considerably across models. The most notable impact is observed in Random Forest, where training time increases nearly tenfold after resampling. This substantial rise is attributable to the ensemble nature of Random Forest, which constructs numerous decision trees using bootstrapped subsets of the expanded dataset. As SMOTETomek enlarges the training data, tree construction complexity scales significantly, leading to disproportionate training overhead. Although detection performance improves, such a drastic increase in training cost may not be justified in real-world deployment scenarios that require frequent retraining or operate under limited computational resources.

In contrast, XGBoost achieves a substantial improvement in minority-class detection with only a moderate increase in training time. Unlike Random Forest, XGBoost builds trees sequentially using gradient boosting, which appears to scale more efficiently with the resampled dataset. This indicates a more favorable trade-off between performance gain and computational cost. From a practical perspective, XGBoost offers a better balance for operational intrusion detection systems where both detection capability and efficiency are critical. Among deep learning models, CNN demonstrates strong detection performance with minimal computational growth, indicating good scalability under resampling conditions. Bi-LSTM also maintains high classification effectiveness, though its recurrent structure leads to higher training cost compared to CNN. Meanwhile, KNN remains computationally expensive during inference, and the increased dataset size further amplifies its testing time, limiting its suitability for real-time environments.

6. Conclusion

This study investigates the classification performance of six machine learning and deep learning classifiers-Random Forest, Decision Tree, K-Nearest Neighbors (KNN), XGBoost, Convolutional Neural Network (CNN), and Bi-LSTM-on the NF-UNSW-NB15 dataset for detecting malicious network traffic. The baseline experiment reveal that conventional models struggle to effectively classify minority

attack classes due to data imbalance and feature overlap, despite achieving high overall accuracy. These findings align with well-known challenges in intrusion detection systems, where imbalanced class distributions lead to poor detection of critical minority attack behaviors.

To address this, the SMOTETomek Link method is implemented to simultaneously oversample minority instances and eliminate boundary-level overlapping samples. The enhanced experimental results demonstrate notable improvements in Recall, F1-score, and overall classifier robustness across all models. Ensemble-based models, particularly XGBoost and Random Forest, achieve superior performance due to their ability to effectively leverage synthetic balanced data while maintaining generalization. Deep learning models such as CNN and Bi-LSTM also exhibit strong performance, particularly in modeling sequential and nonlinear network traffic patterns, though at a higher computational cost.

Overall, the findings confirm that handling both class imbalance and data overlap is critical for improving the accuracy and reliability of intrusion detection systems. The integration of SMOTETomek Link significantly enhances model sensitivity to minority attack classes and improves detection performance without sacrificing generalization.

7. Future Work

Future research directions can be explored in several dimensions:

7.1 Cost-Sensitive and Ensemble Hybrid Learning

While SMOTETomek Link effectively balances the dataset, incorporating cost-sensitive learning or hybrid ensemble approaches (e.g., XGBoost-LSTM or RF-BiLSTM) may further improve minority class detection, especially for low-frequency attack types like Zero-Day and Advanced Persistent Threats (APTs).

7.2 Real-Time Intrusion Detection Integration

Extending this work to real-time or streaming intrusion detection environments using frameworks such as Apache Kafka or Spark Streaming would better reflect real-world cybersecurity requirements, including detection latency and computational efficiency.

7.3 Adaptive and Self-Learning Models

Network attack behaviors evolve over time. Implementing adaptive learning or incremental training techniques (e.g., Lifelong Learning, Online Learning, and Meta-Learning) could enable the models to continuously update and maintain performance without full retraining.

7.4 Multi-Class Attack Categorization

In this study, the classes are mapped into binary classification (benign vs. attack). Extending the approach to multi-class classification would allow for identifying and categorizing specific types of attacks such as DoS, Reconnaissance, Exploits, and Generic, enhancing the practical utility of the system.

7.5 Feature Selection Mechanism

In addition, future work may investigate feature selection strategies to identify the most informative traffic attributes and reduce potential redundancy among the 41 flow-based features. Applying filter-, wrapper-, or embedded-based feature selection techniques could enhance model interpretability, reduce computational complexity, and improve detection efficiency, particularly for deployment in real-time intrusion detection environments.

8. References

- [1] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surv. Tut.*, vol. 18, no. 2, pp. 1153–1176, 2016, doi: 10.1109/COMST.2015.2494502.
- [2] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: A survey," *Appl. Sci.*, vol. 9, no. 20, Oct. 2019, Art. no. 4396, doi: 10.3390/app9204396.
- [3] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009, doi: 10.1109/TKDE.2008.239.
- [4] B. Krawczyk, "Learning from imbalanced data: Open challenges and future directions," *Prog. Artif. Intell.*, vol. 5, no. 4, pp. 221–232, Apr. 2016, doi: 10.1007/s13748-016-0094-0.
- [5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.
- [6] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning from Imbalanced Data Sets*. Cham, Switzerland: Springer International Publishing, 2018, doi: 10.1007/978-3-319-98074-4.
- [7] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explor. Newslett.*, vol. 6, no. 1, pp. 20–29, Jun. 2004, doi: 10.1145/1007730.1007735.
- [8] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining – KDD, 16*, vol. 1, no. 1, pp. 785–794, Aug. 2016, doi: 10.1145/2939672.2939785.
- [9] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.
- [10] Y. Kim, "Convolutional neural networks for sentence classification," 2014, *arXiv*: 1408.5882, doi: 10.48550/arxiv.1408.5882.
- [11] S. Wang, X. Wang, S. Wang, and D. Wang, "Bi-directional long short-term memory method based on attention mechanism and rolling update for short-term load forecasting," *Int. J. Elect. Power Energy Syst.*, vol. 109, pp. 470–479, Jul. 2019, doi: 10.1016/j.ijepes.2019.02.022.

- [12] A. Alshammari and A. Aldribi, "Apply machine learning techniques to detect malicious network traffic in cloud computing," *J. Big Data*, vol. 8, Jun. 2021, Art. no. 90, doi: 10.1186/s40537-021-00475-1.
- [13] M. Kumar, M. Hanumanthappa, and T. V. S. Kumar, "Intrusion detection system using decision tree algorithm," in *IEEE 14th International Conference on Communication Technology*, Chengdu, China, 2012, pp. 629–634, doi: 10.1109/ICCT.2012.6511281.
- [14] M. Dixit, R. Sharma, S. Shaikh, and K. Muley, "Internet traffic detection using Naïve Bayes and K-Nearest Neighbors (KNN) algorithm," in *International Conference on Intelligent Computing and Control Systems (ICCS)*, Madurai, India, 2019, pp. 1153–1157, doi: 10.1109/ICCS45141.2019.9065655.
- [15] N. Farnaaz and M. A. Jabbar, "Random forest modeling for network intrusion detection system," *Procedia Comput. Sci.*, vol. 89, pp. 213–217, Jan. 2016, doi: 10.1016/j.procs.2016.06.047.
- [16] I. L. Cherif and A. Kortebi, "On using eXtreme Gradient Boosting (XGBoost) machine learning algorithm for home network traffic classification," in *Wireless Days (WD)*, Manchester, UK, 2019, pp. 1–6, doi: 10.1109/WD.2019.8734193.
- [17] A. Deshmukh and K. Ravulakollu, "An efficient CNN-based intrusion detection system for IoT: Use case towards cybersecurity," *Technologies*, vol. 12, no. 10, Oct. 2024, Art. no. 203, doi: 10.3390/technologies12100203.
- [18] I. Eweoya et al., "A CNN-based network profiling system for intrusion detection: Addressing evolving cyber threats," *Asian J. Comput. Sci. Technol.*, vol. 14, no. 1, pp. 20–27, Mar. 2025, doi: 10.70112/ajcst-2025.14.1.4332.
- [19] R. M. Al-Khatib, L. Heilat, W. Qudah, S. Alhatamleh, and A. Al-Khateeb, "A novel improved deep learning model based on Bi-LSTM algorithm for intrusion detection in WSN," *Networks and Heterogeneous Media*, vol. 20, no. 2, pp. 532–565, 2025, doi: 10.3934/nhm. 2025024.
- [20] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, Canberra, ACT, Australia, 2015, pp. 1–6, doi: 10.1109/MilCIS.2015. 7348942.
- [21] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 dataset," *Inf. Secur. J. Glob. Perspect.*, vol. 25, no. 1–3, pp. 18–31, Jan. 2016, doi: 10.1080/19393555.2015.1125974.
- [22] T. Wu, H. Fan, H. Zhu, C. You, H. Zhou, and X. Huang, "Intrusion detection system combined enhanced random forest with SMOTE algorithm," *EURASIP J. Adv. Signal Process.*, vol. 2022, May 2022, Art. No. 39, doi: 10.1186/s13634-022-00871-6.

- [23] R. Ahsan, W. Shi, and J. Corriveau, "Network intrusion detection using machine learning approaches: Addressing data imbalance," *IET Cyber-Phys. Syst. Theor. Appl.*, vol. 7, no. 1, Jun. 2021, doi: 10.1049/cps2.12013.
- [24] M. A. Talukder, S. Sharmin, M. A. Uddin, M. M. Islam, and S. Aryal, "MLSTL-WSN: Machine learning-based intrusion detection using SMOTETomek in WSNs," *Int. J. Inf. Secur.*, vol. 23, pp. 2139–2158, Mar. 2024, doi: 10.1007/s10207-024-00833-z.
- [25] L. V and S. Rajkumar, "Hybrid ensemble federated learning using SMOTE-Tomek for efficient DDoS detection on constrained edge devices over 5G networks," *Results Eng.*, vol. 28, Dec. 2025, Art. no. 107601, doi: 10.1016/j.rineng.2025.107601.
- [26] K. Omari and A. Oukhatar, "Advanced phishing website detection with SMOTETomek-XGB: Addressing class imbalance for optimal results," *Procedia Comput. Sci.*, vol. 252, pp. 289–295, 2025, doi: 10.1016/j.procs.2024.12.031.
- [27] D. A. Otchere, T. O. A. Ganat, J. O. Ojero, B. N. Tackie-Otoo, and M. Y. Taki, "Application of gradient boosting regression model for the evaluation of feature selection techniques in improving reservoir characterisation predictions," *J. Pet. Sci. Eng.*, vol. 208, Jan. 2022, Art. no. 109244, doi: 10.1016/j.petrol.2021.109244.
- [28] L. Mohammadpour, T. C. Ling, C. S. Liew, and A. Aryanfar, "A survey of CNN-based network intrusion detection," *Appl. Sci.*, vol. 12, no. 16, Aug. 2022, Art. no. 8162, doi: 10.3390/app12168162.
- [29] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 98–113, Jan. 1997, doi: 10.1109/72.554195.
- [30] L. K. Suresh Kumar, "An efficient network intrusion detection model combining CNN and BiLSTM," *CIBG*, vol. 27, no. 6, pp. 1782–1801, Dec. 2021. [Online]. Available: <https://cibgp.com/index.php/1323-6903/article/view/2273/2233>
- [31] J. Gao, "Network intrusion detection method combining CNN and BiLSTM in cloud computing environment," *Comput. Intell. Neurosci.*, vol. 2022, Apr. 2022, Art. no. 7272479, doi: 10.1155/2022/7272479.
- [32] J. J. A. Armenteros, C. K. Sønderby, S. K. Sønderby, H. Nielsen, and O. Winther, "DeepLoc: Prediction of protein subcellular localization using deep learning," *Bioinformatics*, vol. 33, no. 21, pp. 3387–3395, Nov. 2017, doi: 10.1093/bioinformatics/btx431.
- [33] J. Zhang, X. Zhang, Z. Liu, F. Fu, Y. Jiao, and F. Xu, "A network intrusion detection model based on BiLSTM with multi-head attention mechanism," *Electronics*, vol. 12, no. 19, Oct. 2023, Art. no. 4170, doi: 10.3390/electronics12194170.

- [34] E. F. Swana, W. Doorsamy, and P. Bokoro, "Tomek link and SMOTE approaches for machine fault classification with an imbalanced dataset," *Sensors*, vol. 22, no. 9, Jan. 2022, Art. no. 3246, doi: 10.3390/s22093246.
- [35] V. Jyothsna and K. M. Prasad, "Anomaly-based intrusion detection system," in *Computer and Network Security*, J. Sen, Ed., IntechOpen, Jun. 2019, doi: 10.5772/intechopen.82287..
- [36] G. Velarde, A. Sudhir, S. Deshmane, A. Deshmunkh, K. Sharma, and V. Joshi, "Evaluating XGBoost for balanced and imbalanced data: Application to Fraud detection," 2023, arXiv: 2303.15218, doi: 10.48550/arxiv.2303.15218.