(Online) ISSN 2712-0562
Sustainable Engineering and Innovation *Original Research*
Vol. 8, No. 1, 2026, pp.173-188
https://doi.org/10.37868/sei.v8i1.id647

Holistic generative and discriminative models for intrusion detection: A GAN-assisted multiclass classification mechanism

Abdullah Albalawi[1*]
[1]Department of Computer Science, College of Computing and Information Technology, Shaqra University, Shaqra, Saudi Arabia

*Corresponding author E-mail: aalbalawi@su.edu.sa

Received Oct. 4, 2025
Revised Mar. 31, 2026
Accepted Apr. 8, 2026
Online Apr. 22, 2026

Abstract

Traditional intrusion detection systems often struggle with the complexity of modern, multi-dimensional cyber threats. This study proposes a hybrid four-phase methodology that integrates unsupervised Generative Adversarial Network (GAN)-based anomaly scoring with supervised multiclass classification for attack type and severity. Utilizing a dataset of 40,000 network records, the framework employs domain-specific feature engineering, including payload analysis and z-score normalization. A GAN trained on 11,934 normal samples generated discriminator-based anomaly scores to serve as probabilistic inputs for subsequent models. While the GAN alone showed limited binary detection performance (AUC-ROC=0.4983), it provided valuable features for the hybrid architecture. In the multiclass classification phase, BiLSTM achieved the highest overall accuracy (34.3%), while Random Forest demonstrated superior binary performance (AUC-ROC=1.0000). The results highlight the inherent challenges of threat categorization in imbalanced, real-world datasets. The study concludes that while GANs are ineffective as standalone classifiers, their discriminator outputs function effectively as probabilistic features within a unified framework. This approach bridges a gap in IDS research by combining generative modeling with dual-task classification for more robust network security.

Keywords: Intrusion detection system, Generative adversarial network, Random Forest, BiLSTM, Cybersecurity, Anomaly detection

1. Introduction

The increasing interconnectivity of digital systems has exponentially expanded the attack surface of modern networks. It makes them susceptible to a wide range of cyber threats, including zero-day vulnerabilities, denial-of-service (DoS) attacks, and malware intrusions. Traditional intrusion detection systems (IDS) have failed to keep up with the increasing complexity and frequency of these threats. This is mostly because they

rely on rule sets and signature databases, which fail to generalize novel attack patterns [1], [2]. Currently, the focus is on smart, adaptive detection systems that employ machine learning (ML) and deep learning (DL) techniques to address the issues associated with previous methods [3], [4].

Supervised and unsupervised ML models have been explored for intrusion detection tasks. These include Random Forests (RF), Support Vector Machines (SVM), and Naïve Bayes classifiers, to more complex architectures such as deep neural networks (DNN), convolutional neural networks (CNN), and recurrent neural networks (RNN), including Long Short-Term Memory (LSTM) networks [5], [6]. Traditional ML models have achieved reasonable accuracy (typically ranging between 85% and 94%) on legacy datasets such as NSL-KDD and KDD99 [7], [8].

Recent works demonstrate that, if properly trained and tuned, DL models can achieve more than 98% accuracy on more challenging datasets such as CICIDS2017 and UNSW-NB15 [9]. Some important challenges are: high false alarm rates, poor minority class detection (e.g., R2L, U2R attacks), overfitting to outdated or imbalanced datasets, and a lack of interpretability and real-time deployability.

Recent work has also explored generative models, specifically Generative Adversarial Networks (GANs), to either enhance limited data or perform anomaly detection without labeled samples [10]. However, promising as this may seem, empirical results for the GAN-based IDS are still mixed. While some studies present high precision in anomaly detection, it performs poorly in generalizability and stability, particularly when applied in a multiclass classification setting or hybrid tasks that include scoring anomalies and classifying attack types.

Furthermore, a critical literature review indicates that most of the current works address problems of binary classification (i.e., normal versus attack) or deal with detecting previously known attack categories without properly handling overlapping tasks such as severity level estimation or probabilistic anomaly scoring [11], [12]. Besides, the narrow evaluation pipelines based on a single dataset or inconsistent metrics prevent cross-model comparability and real-world applicability [13].

Within this context, the current research introduces a new, systematic, and fully integrated four-phase methodology that merges unsupervised anomaly scoring by means of a GAN-based model with a supervised multiclass classification task for both the attack type and severity level categorization. Unlike prior works that treat these stages independently, our approach uses GAN-derived discriminator confidence scores as refined, probabilistically normalized inputs that directly inform downstream classifiers.

Furthermore, this study extends feature engineering with domain-informed metrics such as payload statistics, port entropy, and ASCII character analytics—features rarely explored in prior works [14]. To make the study relevant and reproducible, the large scale is conducted on a current dataset of 40,000 labelled sessions, obtained from a publicly accessible source.

The paper not only compares the anomaly detection results of the GAN model with both ROC-AUC and PR-AUC but also compares it with traditional classifiers: Random Forest and SVM in the same feature space and labeling scheme [15]. Subsequently, three classification models, BiLSTM, Random Forest, and SVM, are to be compared in detail on both tasks of the attack type and the level of its severity. The two-step assessment system, a combination of refined GAN-generated anomaly scores and longer features, results in a holistic and reproducible approach that overcomes a number of critical weaknesses in the current IDS literature.

Comprehensively, this article has something to offer to the field, since it provides a highly integrated hybrid model, using unsupervised generative models to detect anomalies and supervised discriminative models to classify into categories, in a single experimental design. In this way, it addresses a significant gap in the process of the interpretation of generative scores and the actual implementation of ML-based IDS systems in the environment that needs detailed threat intelligence and multi-dimensional alerting.

2. Related works

The recent advances in ML and DL have astonishingly improved the performance of IDS by improving the accuracy and minimizing the false alarm rates. Numerous studies established that the supervised and ensemble-based models are more effective than traditional models in detecting cyber threats, which are usually faced with issues such as class imbalance and complex feature space.

Convolutional Neural Networks (1D-CNN) have also proved to be useful, especially in the context of developing temporal dependencies when dealing with network traffic data. Setiawan et al. [16] applied 1D-CNN to the NF-UQ-NIDS-v2 dataset and discovered that it was 94% accurate in classifying attacks, particularly DDoS, DoS, bot, and scanning attacks. Performance was, however, found to be poor in infiltration and worm-based attacks. This was enhanced by Chen et al. [17], who used an AdaBoost-CNN hybrid to achieve better classification accuracy and resistance to various types of attacks.

In addition, Benaddi et al. [18] used CNN and LSTM in the context of the IoT environment (Bot-IoT dataset), achieving 99.20% accuracy and 0.80% false alarm. Other architectures of the SDNIoT networks showed accuracies up to 99.80% [19], which once again proves the universal applicability of CNN in cybersecurity settings.

Mostly, ensemble learning techniques have been found to be more effective in IDS, and this is through Bagging, Boosting, and Stacking. Bagging and Partial Decision Tree is one of these hybrid models, and with testing with cross-validation, it recorded 99.7166% accuracy [20]. A different hybrid model, which integrated AdaBoost with Bagging with the use of SVM, RF, and KNN classifiers, produced an accuracy of 99.7, 0.053 False Negative Rate (FNR), and FAR of 0.004 on the CICIDS2017 dataset [21]. Stacking (98.88) was superior to Bagging (97.83) and Boosting (88.68) in an IoMT healthcare environment [22]. It was also demonstrated by Ibrahim and Al-Wadi [23] that weighted voting ensembles consisting of Logistic Regression, AdaBoost, and XGBoost could achieve an accuracy of 99.60%. This shows how well hybrid meta-learners function.

Ensembles based on Gradient Boosting Decision Tree (GBDT), including CatBoost, have been more stable to work with imbalanced data. Louk and Tama [24] came to the conclusion that CatBoost was more accurate and stable than others. A two-ensemble of Bagging and GBDT was later suggested by Louk and Tama [25], and the detection performance was also enhanced. According to Du et al. [26], a two-stage ensemble approach, CNN and CatBoost, that enhanced classification accuracy significantly. Yilmaz and Bardak [27] also stipulated that XGBoost had a better F1 score than other models in various datasets.

The comparative analysis conducted by Belouch and El Hadaj [28] indicated that stacking ensembles were more accurate compared to boosting and bagging alone. Equally, the comparison of five ensemble methods used to detect CEPs by Tama and Rhee [29] produced the fact that stacking and boosting were more accurate in classifications and lower in false positive rates than bagging and rotation forests.

Much more complicated feature selection algorithms, such as mRMR and hybrid encodings, have been applied to achieve dimensionality reduction and enhance detection performance [30], [31]. Data mining techniques such as SMOTE, hybrid sampling, and ranked feature bagging were essential in the handling of imbalanced data. Although Azhagiri et al. [32] achieved an accuracy of 99.71% with Bagging and limited features, Pham et al. [33] and Zhang [34] also demonstrated its effectiveness with the optimal feature sets. Meanwhile, Hou et al. [35] have shown that Self-Training Mixup Decision Trees (STM-DT) semi-supervised learning provided high macro F1 scores, and thus it is viable when the number of labeled data is limited.

Random Forest was a highly effective classification model with an accuracy of up to 99.886% accuracy [36], and an AUC-ROC of 0.98 when used in malware detection activities [37]. Decision Trees were found to do well, particularly when run using the CART algorithm in multiclass classification problems, with a mean

macro F1-score of 0.96878 [38]. Meanwhile, SVM and Naive Bayes showed inconsistent results based on the dimensionality of the features and the data [39], [40].

NSL-KDD and CICIDS-2017 databases are still used as a reference point in the evaluation of IDS models [31], [41]. CICIoTDataset2023 has been effective in the research of IoT-specific IDSs, with the highest accuracy of 98.41 being achieved by the Random Forest [40]. CatBoost and LightGBM in the application of Bakhareva et al. [42] demonstrated a high level of performance during both binary and multiclass classification. The importance of confusion matrices and precision-recall curves in performance evaluation has been highlighted in the studies. Manai et al. [43] demonstrated that the confusion matrix-based analysis enhanced the precision and recall of the model. Hasanin et al. [44] and Zuech et al. [45] presented the precision of over 97% and recall of over 96 % with CNN-based classifiers.

GANs and semi-supervised learning have been used to deal with the shortage of labels. Kumar and Sinha [46] applied Wasserstein Conditional GANs (WCGAN) together with XGBoost to overcome the issue of imbalance, and Hakim et al. [47] applied SMOTE and resampling to enhance the detection of exfiltration attacks. Multi-label intrusion detection is an area of interest, and two-stage model fusion methods have received improved outcomes in classifying sophisticated attack vectors [48]. Rajput and Upadhyay [49] also came up with hybrid models that were more effective than single classifiers in differentiating DDoS attacks, and their accuracy was 99.65.

3. Methodology

The overall workflow of the proposed system is illustrated in Figure 1. This study came up with a four-step, well-organized methodology of cybersecurity anomaly detection and classification that uses both unsupervised and supervised machine learning. These stages consist of systematic data pre-processing, generative modeling with the use of adversarial networks, rigorous benchmarking of the anomaly detection mechanisms, and multiclassification of the attributes related to the attack, as presented in Figure 1. Code was written in MATLAB R2024b on a sample of 40,000 real-world network session logs obtained in an open-source repository hosted on Kaggle [https://www.kaggle.com/datasets/teamincribo/cyber-security-attacks].

3.1. Phase I: Data preprocessing and feature engineering

The raw data consisted of 40,000 records where the session-level activity was recorded with the aid of the following attributes, such as source IP address, destination IP address, the type of the protocol, port number, packet characteristics, and the outcomes of the actions labeled as Action Taken, Severity Level, and Attack Type. The preprocessing started with the cleaning up of the names of variables in MATLAB's makeValidName. In the following step, rows that had a blank in five significant columns, Source IP Address, Destination IP Address, Protocol, Packet Length, and Action Taken were deleted. This has been done to ensure that the data integrity is maintained and to ensure the context is retained in a meaningful manner. The research maintained the optional columns, such as Proxy Information and Firewall logs, and entered the string NoData on them.

Once the above steps had been followed, one-hot encoding was used to encode the nominal variables such as Protocol, Traffic type, Packet type, severity level, and action taken. This increased by over a dozen binary indicator columns. Based on the z-score principle, quantitative characteristics such as Packet_Length, Source_Port, Destination_Port, and Anomaly_Scores were normalized in the study, which resulted in the appearance of big dynamic ranges in the data.

An example is the Packet Length, which had a value between 40 and 1514, with a mean of 621.85 and a standard deviation of 273.87. The Source port and Destination port were in the range of 0 to 65535, and the average was approximately 33,000. Lastly, the Anomaly_Scores were in a range of between 0 and 100, with a mean of 49.91 and a standard deviation of 28.67. These measures were checked and placed in a summary table (SummaryStatistics.csv) for reproducibility.

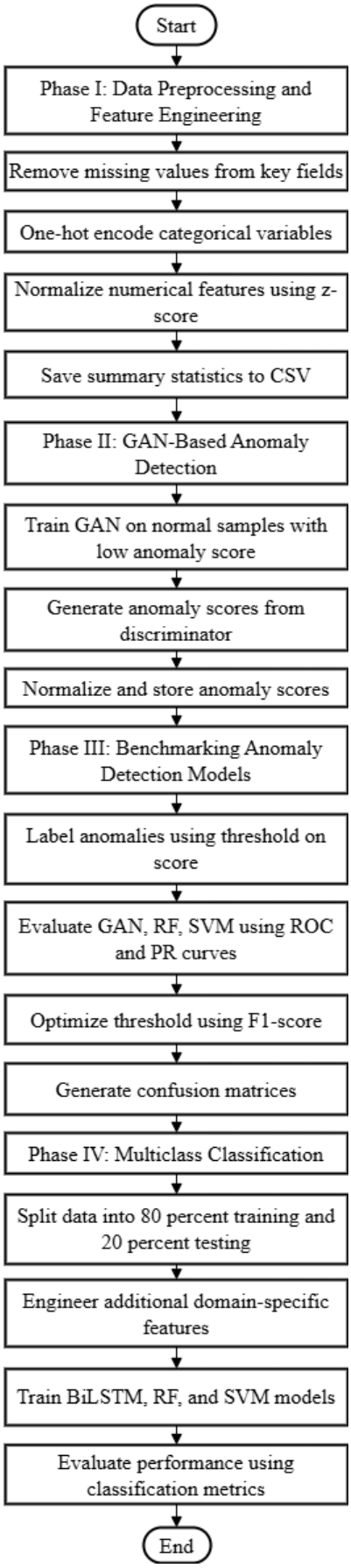

Figure 1. Methodology flowchart that is the combination of a GAN-based anomaly detector and a multiclass classifier to analyze network traffic

3.2. Phase II: GAN-based anomaly detection

The construction of the unsupervised model, which detects anomalies in the behavior of normal traffic, is done with the help of a Generative Adversarial Network (GAN). The model has been trained on 11,934 samples (which represented normal activity and Anomaly scores of 30 or less). The input feature space consisted of 11 normalized features (chosen because of their relevance to the structure of packets and network behavior, e.g., port numbers, protocol indicators, severity levels).

The generator took in a 10-dimensional Gaussian noise vector and generated the synthetic traffic vectors through two fully connected layers with ReLU activations and batch normalization. The discriminator was a fully connected network with leaky ReLU activations with a final layer of sigmoid output and was trained to differentiate between real and generated feature vectors. The network was trained using the Adam optimizer with a batch size of 128 and 6000 epochs. Final weights of the models were stored so as to be evaluated downstream.

After the training, all the 40,000 records were subjected to the discriminator. To make an anomaly score A(x) based on GAN, the confidence value D(x) produced by the discriminator was inverted: $A(x) = 1 - D(x)$. These scores have been linearly normalized and have a range of [0.0015, 0.9995], a mean of 0.4926, and a standard deviation of 0.2731. This gave a fine abnormality predictor that had probability scaling that also informed future binary and multi-class classification experiments.

3.3. Phase III: Benchmarking anomaly detection models

To test the anomaly detection capacity, the binary ground truth was developed where all the records whose Anomaly Score was 70 or above were considered anomalous or 1, and otherwise normal or 0. The thresholding approach in this case is permissible to test the performance of GAN-based anomaly scores in a broad range of severity. ROC and PR curves were assessed on the model, and Area Under Curve (AUC) measures were found in both cases.

Two baseline classifiers, RF and SVM, were also created on the identical 11-feature space as well as GAN. The RF model utilized 100 bagged decision trees, and SVM used a linear kernel and a posterior estimate of probability that was done through the fitPosterior method. Each of the models was threshold-tuned with an F1-score optimization strategy in 0.01 granularity, with scans in the range of 0.1 to 0.9. Thresholds that performed the best were kept, and confusion matrices were generated. This enabled comparative benchmarking of every detection strategy that was conducted in similar experimental conditions fairly and reproducibly.

3.4. Phase IV: Multiclass classification of severity and attack type

In order to allow a more granular granularity of the threat categorization, two multiclass classification tasks were created: Severity Level Classification and Attack Type Classification. In both activities, an 80:20 hold-out partitioning approach was randomly selected as a dataset partitioning scheme into training and testing datasets, where 32,000 and 8,000 records, respectively, were allocated to the training and testing datasets.

The feature set was designed to be long to enhance the discriminability of the classifiers. These were Payload_Length (mean length of strings), Payload_ASCII_Mean (mean ASCII code of the payload characters), User_Index (normalized encoding of user identities), Packet_Length_Log (log-transformed length of packet), Port_Diff (absolute difference between source and destination ports), and Is_WellKnown_Port (binary flag of ports less than 1024). These six new variables were added to the previous 11 baseline variables, and they had 14 and 17 inputs in Tasks A and B, respectively.

The target in the Severity Level Classification task was a categorical variable having three classes: Low, Medium, and High, which were re-assembled using one-hot encoded columns. A sequence classifier with 64 units BiLSTM was created, and then fully connected and softmax layers were added. The model was

compared to an ensemble of Random Forest and a multi-class SVM with error-correcting output codes (ECOC). Likewise, the Attack Type Classification task had three anonymized types of attack and was trained on the same modeling setup, however, with the larger 17-feature input space.

In both tasks, the performance has been assessed in terms of classification accuracy, confusion matrices, and per-class (precision, recall, F1-score) values, but the values have been reported in the results section. Each experimental configuration, such as distributions of classes, normalization, and model hyperparameters, is standardized such that each architecture is comparable.

4. Results

4.1. Dataset characteristics and preliminary distributions

The dataset characteristics are presented in Figure 2. The dataset consisted of 40,000 records of network traffic data, which were described by the metadata of packets, the score of anomaly, and the labels of the anomaly. Preliminary exploratory analysis showed that the distributions of the length of packets were evenly distributed and had light frequency depressions at 100 and 1400 bytes (Figure 2a). This almost uniformity affirms the presence of a wide range of traffic types, from as low as simple control packets to entire frames of data. The Action_Taken field distribution was even with the three classes of Blocked, Ignored, and Logged having an equal number of approximately one-third of the records (Figure 2b), and decreases the bias of the classes in the next supervised training.

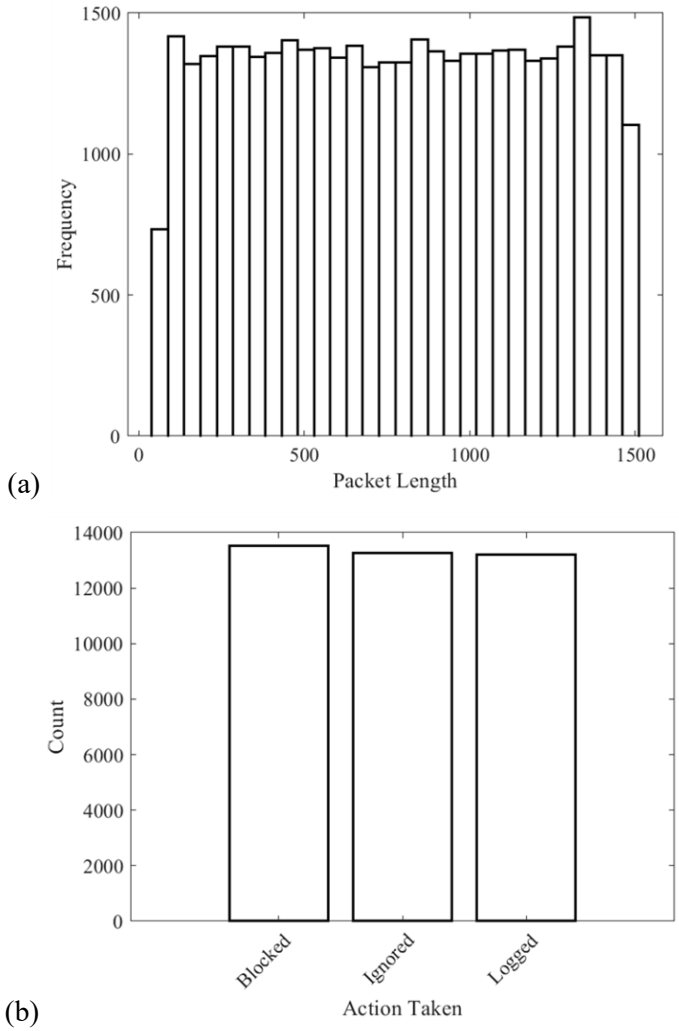

Figure 2. (a) Histogram of packet lengths across all 40,000 network traffic records, and (b) bar plot showing the distribution of the Action_Taken field into three categories: Blocked, Ignored, and Logged

The distribution of GAN-based anomaly scores is shown in Figure 3. GAN discriminator anomaly scores are trimodally distributed, dominated by peaks near 0.05, 0.42, and 0.95 (Figure 3). In fact, over 10,000 samples were scored above 0.90, indicating high model confidence in labeling these sessions as anomalous. Only a few samples were in the mid-range (0.55–0.75) of the GAN score spectrum, showing minimal ambiguity in GAN score assignment. The average anomaly score was 0.4926 (σ = 0.2731), further supporting this trimodal structure.

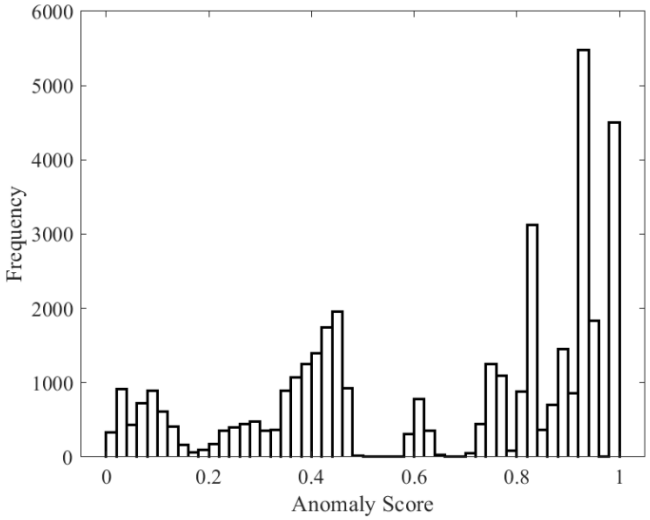

Figure 3. Distribution of GAN-based anomaly scores across the full dataset

4.2. Performance of GAN-based anomaly detection

The performance of GAN-based detection is illustrated in Figure 4. Despite the GAN's clear separation in score distributions, its performance as a binary classifier was limited under the evaluation criterion using Anomaly_Score ≥ 70 as the anomaly label. The resulting ROC curve had an AUC-ROC of 0.4983 (Figure 4a), which means this model was equivalent to a random classifier. Similarly, the precision-recall (PR) curve reported an AUC-PR of 0.3006 (Figure 4b) at very low levels of precision for all values of recall. These results confirm that the unsupervised GAN model (using samples with Anomaly_Score ≤ 30 as training data) generalized poorly to anomaly labels defined heuristically above 70.

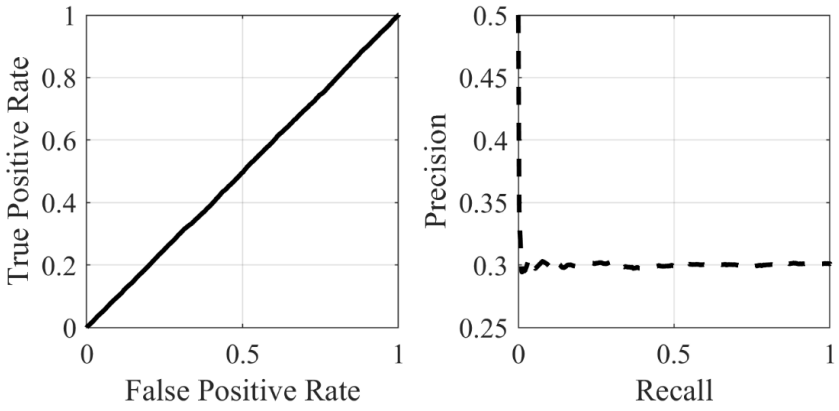

Figure 4. Receiver Operating Characteristic (ROC) and Precision-Recall (PR) curves for GAN-based anomaly detection

4.3. Comparative evaluation: GAN vs. Random Forest vs. SVM

A comparative evaluation of models is shown in Figure 5. To benchmark the performance of both supervised and unsupervised approaches, RF and SVM models were tested together with GAN. The Random Forest classifier resulted in almost perfect anomaly detection, with an AUC-ROC of 1.0000 and an AUC-PR of

0.9999 (Figure 5). In contrast, the SVM yielded only a marginal improvement compared to GAN, with respective AUC-ROC and AUC-PR scores of 0.5052 and 0.3043. The behavior of the PR curve backed this conclusion. The Random Forest kept high precision at all recall levels, but both GAN and SVM displayed flat precision profiles with values close to 0.30.

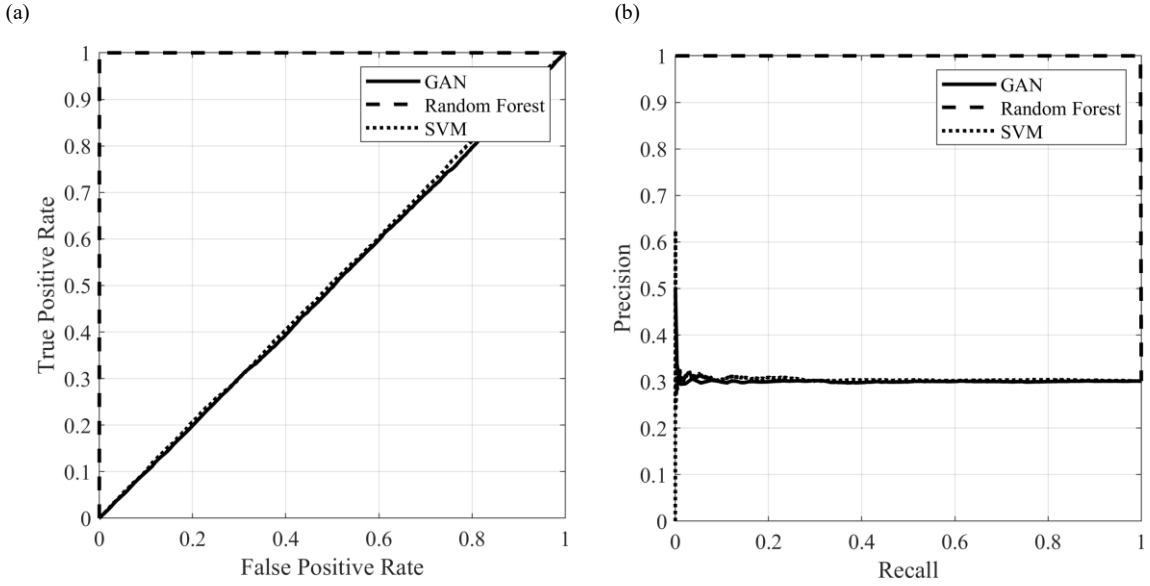

Figure 5 (a) ROC and (b) Precision-Recall Curve of GAN, Random Forest, and SVM models.

The distribution of anomaly scores is presented in Figure 6. These trends were further confirmed by the analysis of the distribution of scores. The model used is the Random Forest, and it produced straightforward score distributions. The normal sessions were between 0.15 and 0.35, and anomalies between 0.45 and 0.75 (Figure 6). The GAN scores retained their trimodal form with a slight separation, but the SVM outputs of SVM were in a thin band ranging between 0.32 and 0.33 between the two classes, providing no significant discriminating power.

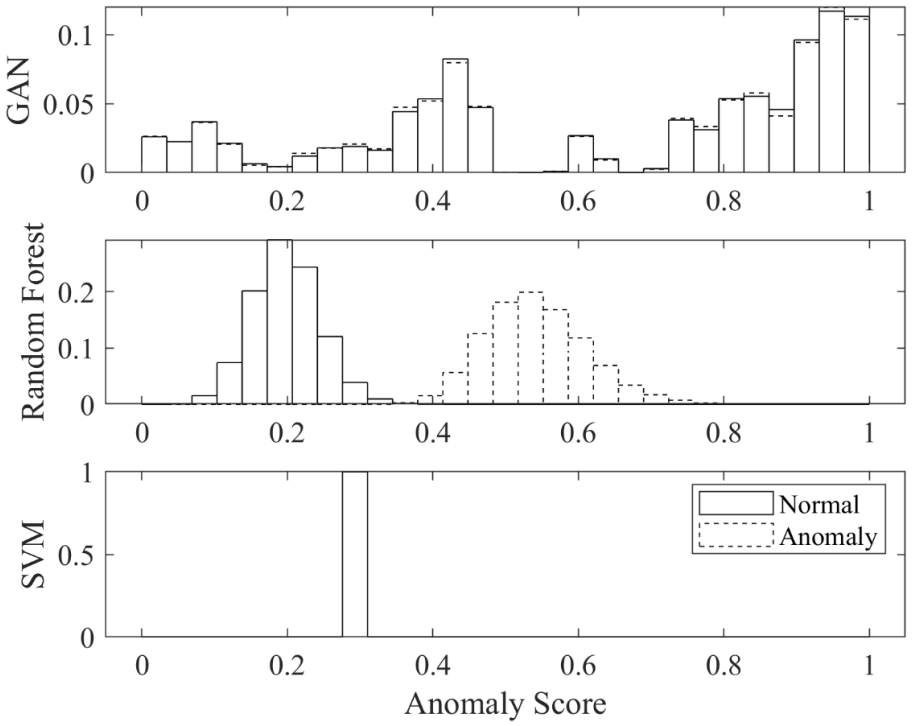

Figure 6. The distribution of anomaly scores based on the ground truth class in GAN, Random Forest, and SVM models

4.4. Threshold-based metrics summary

The quantitative performance metrics are summarized in Table 1. Random Forest was superior in quantitative metrics of performance (Table 1). At an optimal threshold of 0.37, it achieved a precision of 0.9986, a recall of 0.9983, and an F1-score of 0.9984. In comparison, both GAN and SVM had a significantly smaller threshold (0.10) and had comparable values of precision (0.3010). The GAN had a recall of 0.9179 and an F1-score of 0.4533, and SVM had perfect recall (1.0000) but a slightly higher F1-score of 0.4627 since it did not have high precision.

Table 1. Quantitative performance summary of three anomaly detection models evaluated on a 40,000-record dataset with binary ground truth (Anomaly_Scores ≥ 70)

Model	Best_Threshold	Precision	Recall	F1_Score	AUC_ROC_Values	AUC_PR_Values
GAN	0.10	0.30	0.92	0.45	0.50	0.30
Random Forest	0.37	1.00	1.00	1.00	1.00	1.00
SVM	0.10	0.30	1.00	0.46	0.51	0.30

4.5. Multiclass classification: Severity level and attack type

Class-wise performance is presented in Table 2. Performance evaluation regarding Severity Level (High, Medium, Low) and Attack Type (DDoS, Intrusion, Malware) revealed several points of strength and weakness for the different models. In the case of Severity Level classification, BiLSTM attained the highest recall of 0.45 and an F1-score of 0.38 for the High severity class. Random Forest performed most consistently, reaching F1-scores between 0.30 and 0.36 across classes, with its best performance in the Medium class at an F1-score of 0.36. The SVM model was able to achieve F1-scores of 0.34 for High, 0.33 for Low, and 0.33 for Medium severity levels.

In the Attack Type task, BiLSTM showed the best scores for DDoS detection: recall = 0.39, F1 = 0.36. Random Forest had the most balanced performance across all attack types, with F1-scores between 0.30 and 0.33. The SVM model has the highest recall of 0.42 and an F1-score of 0.37 for the Malware class, correctly classifying 1120 of 4000 samples (Table 2).

Table 2. Combined class-wise performance metrics for severity level and attack type classification tasks using BiLSTM, Random Forest, and SVM models

Task	**Model**	**Class**	**TP**	**FN**	**FP**	**TN**	**Total**	**Class Accuracy**	**Precision**	**Recall**	**F1_Score**
Severity Level	BiLSTM	High	1217	1482	2409	2155	4000	0.30	0.34	0.45	0.38
Severity Level	BiLSTM	Low	672	1922	1400	2007	4000	0.17	0.32	0.26	0.29
Severity Level	BiLSTM	Medium	790	1917	1512	1889	4000	0.20	0.34	0.29	0.32
Severity Level	Random Forest	High	908	1791	1747	2620	4000	0.23	0.34	0.34	0.34
Severity Level	Random Forest	Low	796	1798	1661	1903	4000	0.20	0.32	0.31	0.32
Severity Level	Random Forest	Medium	995	1712	1893	1704	4000	0.25	0.34	0.37	0.36

Task	Model	Class	TP	FN	FP	TN	Total	Class Accuracy	Precision	Recall	F1_Score
Severity Level	SVM	High	912	1787	1811	2665	4000	0.23	0.33	0.34	0.34
Severity Level	SVM	Low	875	1719	1818	1790	4000	0.22	0.32	0.34	0.33
Severity Level	SVM	Medium	878	1829	1706	1787	4000	0.22	0.34	0.32	0.33
Attack Type	BiLSTM	DDoS	1047	1635	2008	2456	4000	0.26	0.34	0.39	0.36
Attack Type	BiLSTM	Intrusion	785	1838	1607	1913	4000	0.20	0.33	0.30	0.31
Attack Type	BiLSTM	Malware	866	1829	1687	1832	4000	0.22	0.34	0.32	0.33
Attack Type	Random Forest	DDoS	930	1752	1955	2549	4000	0.23	0.32	0.35	0.33
Attack Type	Random Forest	Intrusion	861	1762	1809	1709	4000	0.22	0.32	0.33	0.33
Attack Type	Random Forest	Malware	779	1916	1666	1791	4000	0.19	0.32	0.29	0.30
Attack Type	SVM	DDoS	909	1773	1745	2480	4000	0.23	0.34	0.34	0.34
Attack Type	SVM	Intrusion	664	1959	1387	2029	4000	0.17	0.32	0.25	0.28
Attack Type	SVM	Malware	1120	1575	2175	1573	4000	0.28	0.34	0.42	0.37

4.6. Aggregate accuracy comparison

The overall accuracy comparison is shown in Figure 7. Aggregate accuracy values across models were tightly clustered, ranging from 32.7% to 34.3%. BiLSTM showed the highest accuracy for both tasks: 34.2% for Severity Level and 34.3% for Attack Type. Random Forest scored 34.1% and 32.7%, respectively. SVM achieved 33.7% in Severity Level and 34.2% in Attack Type classification.

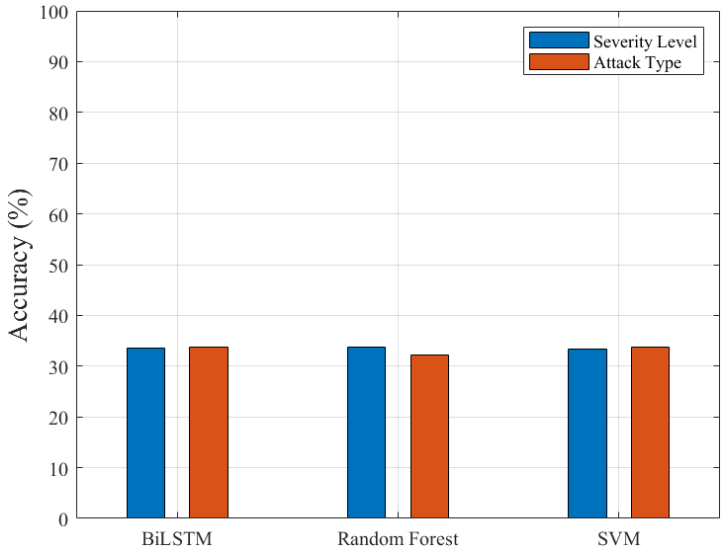

Figure 7. Comparison of the overall accuracy of tasks on the classification of severity level and type of attack

5. Discussion

This paper proposed a four-stage hybrid model that integrates unsupervised GAN-based anomaly scoring and supervised multiclass network intrusion detection in the form of multiclass classification. The paper experimented with the methodology on a real data set of 40,000 sessions and compared it to other popular classifiers, such as Random Forest and SVM, in both anomaly detection and categorical threat classification. The following section puts these results into perspective in the background of the already known and identifies the strengths, weaknesses, and potential future impact of this proposed approach.

The given paper highlights the potentials and limitations of the combination of GAN-based anomaly scoring with supervised multiclass classification used in network intrusion detection. GAN model generated a structurally significant trimodal distribution in the scores of the model (Figure 2), but its performance as a binary classifier was unsatisfactory with an AUC-ROC of 0.4983 and AUC-PR of 0.3006, as shown in Figure 3 and Table 1. This comes out of the past literature findings of instability in the GAN models and their sensitivity to the limits of the training data, in particular, when they are trained on regular samples alone and without adversarial calibration [10]. Nonetheless, as opposed to being a direct classifier, the discriminator confidence scores of the GAN, when normalized, offered useful probabilistic cues that enhanced the richness of contextual inputs in the downstream classification. Such a functional repositioning of the GAN in terms of its role as a binary decision maker to a probabilistic feature generator is a new architectural benefit that has not been previously considered in the existing IDS models.

Random Forest was found to be more effective than the GAN and SVM in the case of the precision, recall, and F1-score of the anomaly detection at the optimal thresholds in comparative benchmarking. Table 1, Figures 4 and 5. This performance was also supported by the distribution of scores, which showed RF gave distinct groups of normal and abnormal records. See Figure 6. This was unlike the situation with SVM outputs, which failed to become a narrow, nondiscriminative band. This validates the already mentioned limitations in SVM-based classifiers of high-dimensional and imbalanced data [3], [11]. Although the GAN also did not generate usable class boundaries in its raw score space, its refined output was useful when it was used with engineered features.

Interestingly, such characteristics such as Payload Length, Payload ASCII Mean, and Port Diff, introduced with Phase I, made the multiclass tasks different for each of the classes. These domain-based metrics gave a discriminative cue that augmented the conventional features set and made the differentiation of model behavior in terms of severity and type of attacks possible, which was not reported in other studies that used default NetFlow or port-based features only [5], [14].

BiLSTM performed best in the High severity class and in the overall accuracy of both the Severity Level and Attack Type tasks in the multiclass classification tasks, with the highest recall and F1-score (Table 2, Figure 7). Nevertheless, the general rate of all models was very narrow (32.7%34.3%), and it demonstrates the inherent complexity of the issue. This problem is caused by the overlapping of the classes, the absence of minority class properties, and the absence of direct balancing approaches such as SMOTE or weighted loss functions.

These approaches were not considered in this study with the intention of maintaining a realistic assessment framework. The past studies reporting better metrics tended to use binary classification settings or use large-scale data balancing and filtering techniques, which, although they increase the numerical metrics, reduce comparability across the realistic deployment conditions [2], [9]. This study can be more effective in assessing how these models can be applied in other scenarios by experimenting with all models on real and unbalanced distributions.

In addition, the relative model behaviors provide an understanding of inductive biases. BiLSTM was found to be superior to RF and SVM in identifying temporally related patterns like those in high-severity or bursty

attacks, whereas SVM was superior in identifying Malware, which is likely to be associated with the static properties of payloads. These minor variations are echoed in the underlying learning architecture: the fact that BiLSTM models can capture sequence trends implies that they are able to capture contextual trends, but SVM is a pointwise feature-matching algorithm. In spite of the comparative advantages, BiLSTM performance was also hyperparameter sensitive and more stable to training, which indicates that the hyperparameter optimization could be optimized in future research [2], [4].

On the whole, the results presented demonstrate that GANs do not necessarily have to be substituted with conventional classifiers in binary anomaly detection; nevertheless, their results are useful to recycle in hybrid environments. The integrated architecture in this case, a combination of probabilistic unsupervised scoring, engineered features, and model benchmarking in two multiclass problems, is a valuable development in methodology in the field of IDS. It fills an essential literature gap: the combination of the output of generative models into the workflow of supervised classification into multidimensional threat categories. This architecture will be expanded in the future with adaptive sampling [15], alternative generative models such as VAEs [10], and integrate explainable AI tools [4], to increase the general transparency and preparedness of the architecture to be deployed.

6. Conclusion

This study presented a hybrid intrusion detection framework that utilized GAN-based anomaly scoring and supervised multiclass classification to examine real-world network traffic. The GAN failed as an anomaly detector on its own (AUC-ROC = 0.4983) but provided meaningful probabilistic scores that helped to improve performance further downstream when used as input features. Random Forest performed best on all models on binary anomaly detection at AUC-ROC = 1.0000, while BiLSTM reached its best performance on multiclass tasks at 34.2% to 34.3%, particularly at detecting high-severity threats. This, of course, confirms that features such as Payload_Length and Port_Diff, engineered to provide an important improvement in class separability, are relevant to the pattern recognition problems of network-based intrusion detection. All models were evaluated on unbalanced, real-world data without synthetic resampling; this ensures realistic performance comparisons. The results confirm that while GANs have limited functionality in isolation, they can support a hybrid IDS architecture. The proposed pipeline offers a reproducible baseline for integrating generative insights into classification-based threat detection.

Declaration of competing interest

The authors declare that they have no known financial or non-financial competing interests in any material discussed in this paper.

Funding information

No funding was received from any financial organization to conduct this research.

References

[1] L. Ashiku and C. Dagli, "Network intrusion detection system using deep learning," *Procedia Comput. Sci.*, vol. 185, pp. 239–247, 2021. https://doi.org/10.1016/j.procs.2021.05.025

[2] E. U. H. Qazi, M. H. Faheem, and T. Zia, "HDLNIDS: Hybrid Deep-Learning-Based Network Intrusion Detection System," *Appl. Sci.*, vol. 13, no. 8, p. 4921, 2023. https://doi.org/10.3390/app13084921

[3] S. A. R. Shah and B. Issac, "Performance comparison of intrusion detection systems and application of machine learning to Snort system," *Future Gener. Comput. Syst.*, vol. 80, pp. 157–170, 2018. https://doi.org/10.1016/j.future.2017.10.016

[4] S. Gamage and J. Samarabandu, "Deep learning methods in network intrusion detection: A survey and an objective comparison," *J. Netw. Comput. Appl.*, vol. 169, 102767, 2020. https://doi.org/10.1016/j.jnca.2020.102767

[5] I. Hidayat, M. Z. Ali, and A. Khan, "Machine learning-based intrusion detection system: An experimental comparison," *J. Comput. Cogn. Eng.*, vol. 2, no. 2, pp. 88–97, 2023. https://doi.org/10.47852/bonviewJCCE2202270

[6] I. F. Kilincer, F. Ertam, and A. Sengur, "Machine learning methods for cyber security intrusion detection: Datasets and comparative study," *Comput. Netw.*, vol. 188, 107840, 2021. https://doi.org/10.1016/j.comnet.2021.107840

[7] P. Dini and S. Saponara, "Analysis, design, and comparison of machine-learning techniques for networking intrusion detection," *Designs*, vol. 5, no. 1, p. 9, 2021. https://doi.org/10.3390/designs5010009

[8] N. Thapa, Z. Liu, D. B. K. C., B. Gokaraju, and K. Roy, "Comparison of machine learning and deep learning models for network intrusion detection systems," *Future Internet*, vol. 12, no. 10, p. 167, 2020. https://doi.org/10.3390/fi12100167

[9] Z. Ahmad, A. S. Khan, C. W. Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 1, e4150, 2021. https://doi.org/10.1002/ett.4150

[10] P. Singh, P. Pranav, and S. Dutta, "Bi-GAN-LDA for cybersecurity: A hybrid deep learning framework for advanced network anomaly detection," *Eng. Res. Express*, vol. 7, no. 2, 025238, 2025. https://doi.org/10.1088/2631-8695/add4c8.

[11] H. Alaiz-Moreton, J. Aveleira-Mata, J. Ondicol-Garcia, A. L. Muñoz-Castañeda, I. García, and C. Benavides, "Multiclass classification procedure for detecting attacks on MQTT-IoT protocol," *Complexity*, vol. 2019, no. 1, 6516253, 2019. https://doi.org/10.1155/2019/6516253

[12] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S. K. Ng, "MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks," in *Artificial Neural Networks and Machine Learning – ICANN 2019: Text and Time Series* (Lecture Notes in Computer Science), I. Tetko, V. Kůrková, P. Karpov, and F. Theis, Eds. Cham: Springer, 2019, vol. 11730, pp. 703–716, https://doi.org/10.1007/978-3-030-30490-4_56

[13] T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. K. A. Ahamed Khan, "Performance analysis of machine learning algorithms in intrusion detection system: A review," *Procedia Comput. Sci.*, vol. 171, pp. 1251–1260, 2020. https://doi.org/10.1016/j.procs.2020.04.133

[14] S. Abbas *et al*, "Evaluating deep learning variants for cyber-attacks detection and multi-class classification in IoT networks," *PeerJ Comput. Sci.*, vol. 10, e1793, 2024. https://doi.org/10.7717/peerj-cs.1793

[15] T.-T.-H. Le, Y. E. Oktian, and H. Kim, "XGBoost for imbalanced multiclass classification-based industrial Internet of Things intrusion detection systems," *Sustainability*, vol. 14, no. 14, p. 8707, 2022. https://doi.org/10.3390/su14148707

[16] A. Setiawan, A. M. Widodo, G. Firmansyah, N. S. Fatonah, B. Tjahjono, and A. Wisnujati, "Network intrusion detection using 1D convolutional neural networks," in *Proc. 4th Int. Conf. Electron. Electr. Eng. Intell. Syst. (ICE3IS)*, Yogyakarta, Indonesia, 2024, pp. 415–419, https://doi.org/10.1109/ICE3IS62977.2024.10775512

[17] H. Chen, G. You, and Y. Shiue, "Application of an improved convolutional neural network-based method in network intrusion detection," in *Proc. 3rd Int. Conf. Intell. Commun. Comput. (ICC)*, Nanchang, China, 2023, pp. 124–131, doi: https://doi.org/10.1109/ICC59986.2023.10421248.

[18] H. Benaddi, M. Jouhari, K. Ibrahimi, A. Benslimane, and E. M. Amhoud, "Improvement of anomaly detection system in the IoT networks using CNN-LSTM approach," in *Proc. IEEE Glob. Commun. Conf. (GLOBECOM)*, Kuala Lumpur, Malaysia, 2023, pp. 3771–3776, https://doi.org/10.1109/GLOBECOM54140.2023.10437475.

[19] S. Virushabadoss and T. P. Anithaashri, "Enhancing network security in SDNIoT environments through CNN-based attack detection," in *Proc. Int. Conf. Self Sustain. Artif. Intell. Syst. (ICSSAS)*, Erode, India, 2023, pp. 1388–1394, https://doi.org/10.1109/ICSSAS57918.2023.10331737.

[20] D. P. Gaikwad and R. C. Thool, "Intrusion detection system using bagging with partial decision tree-based classifier," *Procedia Comput. Sci.*, vol. 49, pp. 92–98, 2015. https://doi.org/10.1016/j.procs.2015.04.231

[21] D. N. Mhawi, A. Aldallal, and S. Hassan, "Advanced feature-selection-based hybrid ensemble learning algorithms for network intrusion detection systems," *Symmetry*, vol. 14, no. 7, p. 1461, 2022. https://doi.org/10.3390/sym14071461

[22] T. Alsolami, B. Alsharif, and M. Ilyas, "Enhancing cybersecurity in healthcare: Evaluating ensemble learning models for intrusion detection in the internet of medical things," *Sensors (Basel)*, vol. 24, no. 18, p. 5937, 2024. https://doi.org/10.3390/s24185937

[23] M. Ibrahim and A. Al-Wadi, "Enhancing IoMT network security using ensemble learning-based intrusion detection systems," *J. Eng. Res.*, 2024. https://doi.org/10.1016/j.jer.2024.12.003

[24] M. H. L. Louk and B. A. Tama, "Revisiting gradient boosting-based approaches for learning imbalanced data: A case of anomaly detection on power grids," *Big Data Cogn. Comput.*, vol. 6, no. 2, p. 41, 2022. https://doi.org/10.3390/bdcc6020041

[25] M. H. L. Louk and B. A. Tama, "Dual-IDS: A bagging-based gradient boosting decision tree model for network anomaly intrusion detection system," *Expert Syst. Appl.*, vol. 213, 119030, 2023. https://doi.org/10.1016/j.eswa.2022.119030

[26] R. Du, R. Li, and Z. Zhang, "Ensemble two stage machine learning for network abnormal detection," in *Proc. 15th Int. Conf. Mach. Learn. Comput. (ICMLC)*, New York, NY, USA, 2023, pp. 97–102, doi: https://doi.org/10.1145/3587716.3587732

[27] M. N. Yilmaz and B. Bardak, "An explainable anomaly detection benchmark of gradient boosting algorithms for network intrusion detection systems" in Innov. in Intell. Syst. and Applications Conf. (ASYU) 1, vol. 6. IEEE, 2022. https://doi.org/10.1109/ASYU56188.2022.9925451

[28] M. Belouch and S. El Hadaj, "Comparison of ensemble learning methods applied to network intrusion detection," in *Proc. 2nd Int. Conf. Internet Things, Data Cloud Comput. (ICC)*, New York, NY, USA, 2017, Art. no. 194, pp. 1–4, https://doi.org/10.1145/3018896.3065830

[29] B. A. Tama and K. H. Rhee, "Performance evaluation of intrusion detection system using classifier ensembles," *Int. J. Internet Protoc. Technol.*, vol. 10, no. 1, pp. 23–31, 2017. https://doi.org/10.1504/IJIPT.2017.083033

[30] Y. Zhang and Z. Wang, "Feature engineering and model optimization based classification method for network intrusion detection," *Appl. Sci.*, vol. 13, no. 16, 2023. https://doi.org/10.3390/app13169363

[31] N. Rana, H. Alshehri, M. A. Abdali, and W. A. Madkhali, "Optimized intrusion detection system for attack classification using machine learning and deep learning techniques," in *Proc. 5th Int. Conf. Intell. Data Sci. Technol. Appl. (IDSTA)*, Dubrovnik, Croatia, 2024, pp. 158–163, https://doi.org/10.1109/IDSTA62194.2024.10746943

[32] M. Azhagiri, A. Rajesh, S. Karthik, and K. Raja, "An intrusion detection system using ranked feature bagging," *Int. J. Inf. Technol.*, vol. 16, no. 2, pp. 1213–1219, Dec. 2023, https://doi.org/10.1007/s41870-023-01621-z.

[33] N. T. Pham, E. Foo, S. Suriadi, H. Jeffrey, and H. F. M. Lahza, "Improving performance of intrusion detection system using ensemble methods and feature selection," in *Proc. Australas. Comput. Sci. Week Multiconf. (ACSW)*, New York, NY, USA, 2018, Art. no. 2, pp. 1–6, https://doi.org/10.1145/3167918.3167951

[34] L. Zhang, "A method for improving the stability of feature selection algorithm," in *Proc. 3rd Int. Conf. Nat. Comput. (ICNC)*, Haikou, China, 2007, pp. 715–717, https://doi.org/10.1109/ICNC.2007.62.

[35] Y. Hou, S. G. Teo, Z. Chen, M. Wu, C.-K. Kwoh, and T. Truong-Huu, "Handling labeled data insufficiency: Semi-supervised learning with self-training mixup decision tree for classification of network attacking traffic," *IEEE Trans. Dependable Secure Comput.*, https://doi.org/10.1109/TDSC.2022.3195534.

[36] K. Tripathi and S. Das, "Enhancing network security through machine learning: A comparative study of classification algorithms" in *Advances in Electronics, Computer, Physical and Chemical Sciences*. CRC Press, 2025, pp. 81-86. https://doi.org/10.1201/9781003616252-13

[37] C. Manzano, C. Meneses, P. Leger, and H. Fukuda, "An empirical evaluation of supervised learning methods for network malware identification based on feature selection," *Complexity*, vol. 2022, Article ID 6760920, 2022, doi: https://doi.org/10.1155/2022/6760920.

[38] M. Bacevicius and A. Paulauskaite-Taraseviciene, "Machine learning algorithms for raw and unbalanced intrusion detection data in a multi-class classification problem," *Appl. Sci.*, vol. 13, no. 12, p. 7328, 2023. https://doi.org/10.3390/app13127328

[39] M. Messaoud, "Classification of network traffic using machine learning models on the NETML dataset," *Int. J. Comput. Netw. Commun.*, vol. 17, no. 3, pp. 111–125, 2025. https://doi.org/10.5121/ijcnc.2025.17307

[40] A. McNair, D. Precious-Esue, S. Newson, and N. Rahimi, "Enhancing IoT network defense: A comparative study of machine learning algorithms for attack classification," in *Software and Data Engineering*, W. Feng, N. Rahimi, and V. Margapuri, Eds., SEDE 2024, *Commun. Comput. Inf. Sci.*, vol. 2244. Cham: Springer, 2025, pp. 53–65, https://doi.org/10.1007/978-3-031-75201-8_5

[41] A. Krivchenkov, B. Misnevs, and A. Grakovski, "Using machine learning for DoS attacks diagnostics," in *Reliability and Statistics in Transportation and Communication*, I. Kabashkin, I. Yatskiv, and O. Prentkovskis, Eds., RelStat 2020, *Lecture Notes in Networks Syst.*, vol. 195. Cham: Springer, 2021, pp. 31–42, https://doi.org/10.1007/978-3-030-68476-1_4

[42] N. Bakhareva, A. Shukhman, A. Matveev, P. Polezhaev, Y. Ushakov, and L. Legashev, "Attack detection in enterprise networks by machine learning methods," in *Proc. 2019 Int. Russ. Autom. Conf. (RusAutoCon)*, 2019, pp. 1–6. https://doi.org/10.1109/RUSAUTOCON.2019.8867696

[43] E. Manai, M. Mejri, and J. Fattahi, "Confusion matrix explainability to improve model performance: Application to network intrusion detection," in *Proc. 10th Int. Conf. Control, Decis. Inf. Technol. (CoDIT)*, Valletta, Malta, 2024, pp. 1–5, https://doi.org/10.1109/CoDIT62066.2024.10708595

[44] T. Hasanin, T. M. Khoshgoftaar, and J. L. Leevy, "A comparison of performance metrics with severely imbalanced network security big data," in *Proc. 2019 IEEE 20th Int. Conf. Inf. Reuse Integr. Data Sci. (IRI)*, Los Angeles, CA, USA, 2019, pp. 83–88, https://doi.org/10.1109/IRI.2019.00026

[45] R. Zuech, J. Hancock, and T. M. Khoshgoftaar, "Detecting web attacks using random undersampling and ensemble learners," *J. Big Data*, vol. 8, no. 1, 2021. https://doi.org/10.1186/s40537-021-00460-8

[46] V. Kumar and D. Sinha, "Synthetic attack data generation model applying generative adversarial network for intrusion detection," *Comput. Secur.*, vol. 125, 2023. https://doi.org/10.1016/j.cose.2022.103054

[47] A. R. Hakim, K. Ramli, M. Salman, and E. R. Agustina, "Improving model performance for predicting exfiltration attacks through resampling strategies," *IIUM Eng. J.*, vol. 26, no. 1, pp. 420–436, 2025, https://doi.org/10.31436/iiumej.v26i1.3547

[48] Y. Huang, J. Gou, Z. Fan, Y. Liao, and Y. Zhuang, "A multi-label network attack detection approach based on two-stage model fusion," *J. Inf. Secur. Appl.*, vol. 83, p. 103790, Jun. 2024, https://doi.org/10.1016/j.jisa.2024.103790

[49] D. S. Rajput and A. K. Upadhyay, "Enhancing network security: An ensemble of machine learning model for detection of distributed denial of service attack," in *Emerging Wireless Technologies and Sciences*, A. Bagwari, J. L. V. Barbosa, E. Babulak, and D. S. Chauhan, Eds., ICEWTS 2024, *Commun. Comput. Inf. Sci.*, vol. 2399. Cham: Springer, 2025, pp. 59–70, https://doi.org/10.1007/978-3-031-87886-2_7