Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

Comparative Analysis of Machine Learning Techniques for Network Intrusion Detection in Cyber Security with a Diverse Metric-Based Performance Assessment

Sufyan Muhammad Khan *
Department of Information Technology
PhD Computer Science Sindh Madressatul Islam University, Karachi (SMIU, Karachi)
sufyan.m.khan.orakzai@gmail.com

Muhammad Azeem
School of Engineering and Applied sciences ,Gift university Gujranwala, Pakistan
azeemaslam16206@gmail.com
Orcid id 0009-0000-6237-5252

Saad Ullah Butt
Computer Science and Electrical Engineering
(CSEE) MSc University of Essex
Saadbutt1313@gmail.com
Orcid id 0009-0007-3681-3512

ABSTRACT

Network intrusion detection is one of the major recurring problems in today's cyber security landscape, with attackers continually finding new ways to mimic legitimate traffic with malicious activity. A comparative study of eight supervised machine learning algorithms, namely: Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Naive Bayes and a Multi-Layer Perceptron (MLP) neural network, for evaluation using a synthetic network intrusion dataset that contains 300 labelled traffic samples belonging to five classes: Normal, Denial of Service (DoS), Probe, Remote-to-Local (R2L), User-to-Root (U2R). We do not use any single measure of accuracy, but rather a comprehensive assessment framework based on multiple measures of accuracy: accuracy, precision, recall, F1-score, Matthews Correlation Coefficient (MCC), specificity, false positive rate and Area Under the ROC Curve (AUC), which have been validated using stratified 5-fold cross validation. All classifiers perform in a very tight range, around 47% to 54% accuracy, with the MCC values around 0 and the AUC values around 0.48 – 0.52, as shown in our results. Overall these findings suggest that none of the algorithms successfully learned a discriminative signal to a level of significance above random guessing on this specific data set. A supporting correlation analysis verifies that the numeric traffic features (duration, source bytes, destination bytes, failed logins and packet count) have near zero linear inter-correlations with the intrusion label, and this is why the models perform uniformly badly across all families.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

We believe this is a valuable result as it shows why it is unwise to judge an intrusion detection system by its accuracy alone: several of the models studied in this paper achieved an accuracy above 50% and their MCC and AUC values indicate that they are performing no better than random guessing. Finally, the paper provides some practical considerations for dataset design, feature engineering, and evaluation methodology in the context of applied problem solving in ID research, and gives concrete suggestions for future research with a larger, richer, and more realistically correlated network traffic datasets like NSL-KDD, CICIDS2017, or UNSW-NB15.

Keywords—Network Intrusion Detection, Machine Learning, Cyber Security, Comparative Analysis, Random Forest, Support Vector Machine, Performance Metrics, Matthews Correlation Coefficient, ROC-AUC, Feature Importance

1. Introduction

The attack surface has opened up to malicious actors because of the rapid growth of networked infrastructure, cloud services and Internet-of-Things (IoT) devices. The traditional Intrusion Detection Systems (IDS) that depend on pre-defined rules and known attack fingerprints are unable to keep up with the ever-changing nature of zero day attacks and polymorphic attack methods (Buczak & Guven, 2016; Liao et al., 2013). This constraint has led to a long-standing trend towards machine learning (ML) intrusion detection systems in which machine learning algorithms are trained to recognize the difference between normal and malicious traffic flows by learning directly from the data itself, rather than from hand-crafted rules (Garcia-Teodoro et al., 2009; Khraisat et al., 2019).

Although ML-based IDS is receiving a lot of attention in the academic community, it is well to note that many published studies suffer from a single performance metric (usually accuracy) to proclaim a model as successful (Chicco & Jurman, 2020). In the security environment, accuracy can be deceiving, especially when the class distributions are imbalanced (Japkowicz & Stephen, 2002) or when a classifier is basically guessing (Japkowicz, 2003). Even if a model is able to give a very good accuracy on the final test set, it might not be able to offer any operational security value. This is the reason why the diverse and multi-metric evaluation framework used in this paper is based on recommendations for evaluating classifiers in a balanced way (Powers, 2011; Fawcett, 2006).

This study uses a 300-sample network traffic dataset that includes typical connection-level features (duration, protocol, service, source/destination byte counts, connection flag, failed login attempts, and packet counts), a categorical label for the attack type (Normal, DoS, Probe, R2L, U2R) and a binary label for intrusion detection (Yes or No) to systematically compare eight popular supervised machine learning algorithms following a schema similar to the ones used in benchmark corpora like NSL-KDD (Tavallaee et al., 2009) and UNSW-NB15 (Moustafa & Slay, 2015). The specific goals of this work are to: (1) compare the performance of Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, SVM, KNN, Naive Bayes, and MLP models on the same train/test splits; (2) use a diverse set of evaluation metrics, including accuracy, precision, recall, F1-score, MCC, specificity, false positive rate, and AUC, and report

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

performance across these metrics, not just a single metric; (3) assess the stability of each model using stratified 5-fold cross-validation; (4) analyze feature importance to understand which traffic features, if any, contribute most to the classification decision. The rest of this paper is structured as follows. The related work in the field of intrusion detection by machine learning is summarized in section 2. The dataset, data preprocessing pipeline, algorithms employed, and evaluation metrics used in this study are described in Section 3. Section 4 provides experimental results such as exploratory data analysis, model comparison, ROC-AUC analysis, confusion matrix and feature importance ranking. The results of this study are discussed in Section 5 and the limitations in Section 6. The paper is ended in section 7 with a discussion of future directions.

2. Related Work

In the last two decades, there has been a wealth of research on machine learning based IDS techniques. The base feature construction method proposed by Lee and Stolfo (2000) was expanded and then used to distinguish between categories of attacks such as DoS, Probe, R2L, and U2R using classical algorithms like Decision Trees (Quinlan, 1986), Naive Bayes (Rish, 2001), and Support Vector Machines (Cortes & Vapnik, 1995) to build upon on the early work, which employed the KDD Cup 1999 and its streamlined version, NSL-KDD, as benchmark datasets (Tavallaee et al., 2009). Although these studies generally claimed high accuracy, more recent studies found that the datasets from KDD are highly redundant and feature-leaky (Tavallaee et al., 2009; Sommer & Paxson, 2010).

Later studies moved on to more realistic datasets like CICIDS2017 (Sharafaldin et al., 2018; Panigrahi & Borah, 2018) and UNSW-NB15 (Moustafa & Slay, 2015; Kumar et al., 2020) which represent more realistic attack categories such as botnets, brute-force attacks and web-based exploits under more naturalistic traffic conditions. In this literature, the performance of ensemble methods such as Random Forest (Breiman, 2001; Zhang et al., 2008) and the various versions of Gradient Boosting (Friedman, 2001; Ahmim et al., 2019) often has been superior to the single-model baselines, both for their ability to learn non-linear feature interactions and because they are inherently resistant to overfitting on noisy features. Additionally, deep learning architectures such as Multi-Layer Perceptrons (Rumelhart et al., 1986), Convolutional Neural Networks and Long Short-Term Memory networks have been investigated (Vinayakumar et al., 2019; Kwon et al., 2019; Ferrag et al., 2020; Elmasry et al., 2020), with most demonstrating high accuracy, although they are more data-hungry and less interpretable than classical models (Kim et al., 2014; Mishra et al., 2019).

Evaluation metric selection is a recurring theme throughout this body of work, and was a motivation for the present study. A number of survey papers have specifically warned that the accuracy measure is not enough to evaluate an IDS because there are very few attacks in the database (e.g., U2R, R2L) compared with the large number of normal cases (Japkowicz & Stephen, 2002; Chawla et al., 2002). The F1-score, Matthews Correlation Coefficient and Area Under the ROC Curve are some metrics that are increasingly suggested to better quantify a classifier's actual discriminative power (Chicco & Jurman, 2020; Powers, 2011; Fawcett, 2006), as they explicitly consider false positives and false negatives, while not treating all correct predictions the same.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

This concern is confirmed by broader surveys of anomaly-based and hybrid intrusion detection approaches in classical architectures (Garcia-Teodoro et al., 2009; Chandola et al., 2009; Khraisat et al., 2019) and deep architectures (Liao et al., 2013; Aljawarneh et al., 2018; Ring et al., 2019). The methodology used in this paper follows and extends that guidance by applying a comprehensive set of metrics for evaluation, instead of a single summary indicator, to a controlled comparative benchmark.

3. Materials and Methods

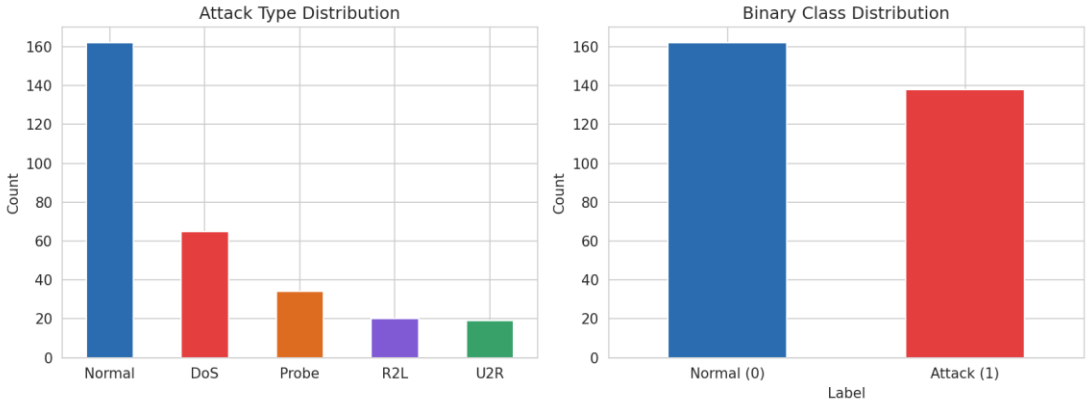
3.1 Dataset Description

This study is based on 300 synthetic network connection records, characterized by 8 core features and a class label. The schema can be summarized in Table 1.

Feature	Type	Description
Duration	Numeric	Length of the connection in seconds (range observed: 4–500)
Protocol	Categorical	Transport protocol: TCP, UDP, or ICMP
Service	Categorical	Network service: HTTP, DNS, SSH, FTP, or SMTP
Src_Bytes	Numeric	Bytes sent from source to destination
Dst_Bytes	Numeric	Bytes sent from destination to source
Flag	Categorical	Connection status flag: SF, S0, or REJ
Failed_Logins	Numeric	Count of failed login attempts (0–6)
Packets	Numeric	Total packets exchanged in the connection (1–999)
Attack_Type	Categorical	Ground-truth category: Normal, DoS, Probe, R2L, U2R
Label	Binary	0 = Normal traffic, 1 = Intrusion (any attack type)

Table 1. Dataset feature schema.

Nearly half of the records were labeled as an intrusion (138 records or 46.0%) while the remaining 54.0% (162 records) were labeled Normal (0). Within the 138 intrusion records, the attack-type breakdown is: DoS (65 samples, 21.7% of the total dataset), Probe (34 samples, 11.3%), R2L (20 samples, 6.7%), and U2R (19 samples, 6.3%). Both distributions are displayed in figure 1.



Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

Figure 1. Attack type distribution (left) and binary class distribution (right).

No missing values were present in any of the 300 records across the 11 columns, so no imputation was required. Table 2 reports summary statistics for the numeric features.

Statistic	Duration	Src_Bytes	Dst_Bytes	Failed_Logins	Packets
Mean	249.3	24,908	25,532	3.0	501.3
Std. Dev.	137.0	14,127	14,235	2.0	278.8
Minimum	4	277	158	0	1
Median	244.5	24,983	25,750	3	499
Maximum	500	49,984	49,999	6	999

Table 2. Descriptive statistics of numeric traffic features.

3.2 Data Preprocessing

Label Encoding was used to convert the three categorical features (Protocol, Service, Flag) to numeric form. For the distance-based and gradient-based algorithms (Logistic Regression, SVM, KNN, and MLP), features were scaled to zero mean and unit variance before being passed to these models, which are sensitive to feature scale; for the tree-based models (Decision Tree, Random Forest, Gradient Boosting) and Naive Bayes, features were not scaled as these models are scale-invariant or make distributional assumptions that do not require scaling. The dataset was divided into training (70%) and held-out (30%) portions with stratified sampling to maintain the original class ratio in both portions of the dataset.

3.3 Machine Learning Algorithms

We chose 8 supervised classifiers to cover a wide range of algorithmic families popular in intrusion detection research:

- Logistic Regression is a linear probabilistic model that is used as a baseline model (Hosmer et al., 2013).
- Decision Tree: one interpretable tree classifier (max depth = 6) using thresholds on features (Quinlan, 1986).
- Random Forest was selected as an ensemble of 200 decision trees built with max depth of 8 trees and aggregated by majority voting in order to be less prone to overfitting (Breiman, 2001; Zhang et al., 2008).
- The Gradient Boosting is a sequential network of 150 shallow trees which correct the errors of previous trees iteratively (Friedman, 2001).
- Support Vector Machine (SVM): maximum margin classifier with Radial Basis Function (RBF) kernel (with probability estimates, Cortes & Vapnik, 1995).
- K-Nearest Neighbors (KNN): Non-parametric instance-based classifer (Cover & Hart, 1967) with number of neighbours k = 7.
- Naive Bayes - Gaussian Naive Bayes classifier, assuming independence of the features (Rish, 2001). Multi-Layer Perceptron (MLP) - a feed-forward neural network with 32 and 16 neurons in the two hidden layers, trained using the back-propagation method of Rumelhart et al., (1986).

All models were implemented in Python with scikit-learn (Pedregosa et al., 2011) and

Online ISSN 3007-3197
Print ISSN 3007-3189

used the same random seed (42) for all steps of the train/test split, model set-up, and crossvalidation folds, to make the results fully reproducible. The theory of the statistical learning approach used in these algorithms is standard (Hastie et al., 2009; Bishop, 2006).

3.4 Evaluation Metrics

As in the other metric-based assessment philosophies that underlie this study, each model was assessed by eight complementary metrics, not just by accuracy.

- Accuracy — percentage of all predictions that are correct.
- Precision — percentage of the correct identifications of intrusions (minimizes false alarms).
- Recall (Sensitivity) — percentage of "intrusions" that are correctly identified (minimizes false-negatives).
- F1-score — harmonic mean of precision and recall, when both false positives and false negatives are costly.
- Specificity: Percentage of true normal traffic that is correctly recognized as normal.
- Error False Positive Rate (FPR) — percentage of normal traffic that is incorrectly detected as an intrusion.
- Matthews Correlation Coefficient (MCC) — one of the most balanced single-value summaries for binary classification, ranging from –1 to +1 that considers all four confusion-matrix quadrants, (Chicco & Jurman, 2020)..

3.5 Experimental Setup

All experiments were executed in a Python 3 environment using scikit-learn 1.x, pandas, and NumPy. Confusion matrices, ROC curves, and feature importance scores were computed on the held-out test set (or, for cross-validation, across folds), and all visualizations were generated using Matplotlib and Seaborn. The complete experimental pipeline — encoding, scaling, training, prediction, and metric computation — was applied identically across all eight models to ensure a fair, like-for-like comparison.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

4. Results

4.1 Exploratory Correlation Analysis

We first investigated the linear relationship between each numeric model feature and the binary intrusion label, computed using Pearson correlation coefficients, prior to comparing the performance of the models. In Figure 2, we can see that no numeric features have a correlation with label, greater than 0.09 in absolute value (Duration: 0.048; Src_Bytes: 0.089; Dst_Bytes: 0.002; Failed_Logins: –0.052; Packets: 0.012). Marginal differences in intrusion rate within categories (based on cross tabulations of the categories against the label) are also found, for example, SSH connections have a slightly lower intrusion rate than DNS connections, and UDP traffic is slightly higher — but none of these differences are individually strong enough to be used as a clear predictive signal.

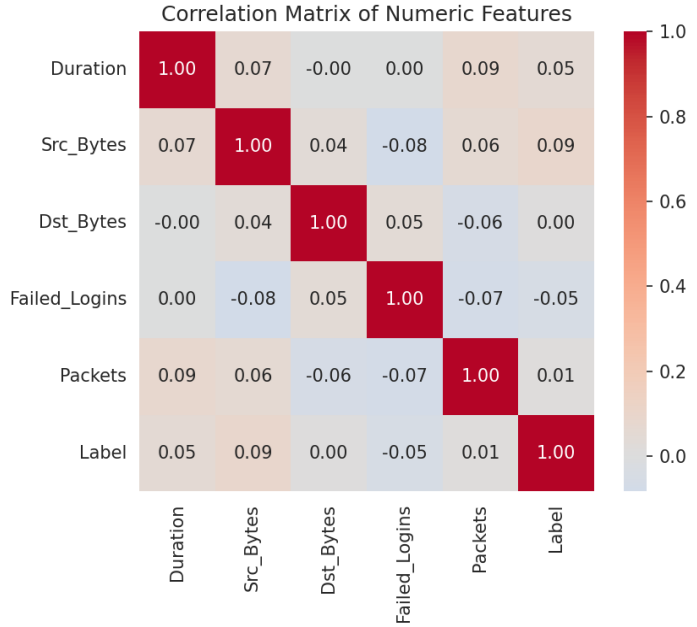


Figure 2. Correlation matrix of numeric traffic features and the binary label.

This paucity of linear or categorical association between the individual features and the label is an important initial finding: if there is no strong signal in any one feature, then whether or not non-linear combinations of features can recover a useful decision boundary will be crucial to the performance of the models: and, as the next sections show, none of the eight models tested could recover a useful decision boundary beyond chance level.

4.2 Comparative Model Performance

Table 3 reports the full set of test-set metrics for all eight classifiers on the 90-sample held-out test partition.

Model	Accuracy	Precision	Recall	F1	Specificity	FPR	MCC	AUC
Logistic Regressio	0.522	0.472	0.415	0.442	0.612	0.388	0.027	0.498

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189

http://amresearchreview.com/index.php/Journal/about

Model	Accuracy	Precision	Recall	F1	Specificity	FPR	MCC	AUC
n								
Decision Tree	0.511	0.457	0.390	0.421	0.612	0.388	0.003	0.485
Random Forest	0.500	0.438	0.341	0.384	0.633	0.367	-0.027	0.479
Gradient Boosting	0.489	0.429	0.366	0.395	0.592	0.408	-0.043	0.435
Support Vector Machine	0.544	0.500	0.317	0.388	0.735	0.265	0.057	0.498
K-Nearest Neighbors	0.489	0.419	0.317	0.361	0.633	0.367	-0.053	0.497
Naive Bayes	0.511	0.457	0.390	0.421	0.612	0.388	0.003	0.521
Multi-Layer Perceptron	0.467	0.415	0.415	0.415	0.510	0.490	-0.075	0.445

Table 3. Comparative test-set performance across eight classifiers (best value per column in the underlying computation is discussed in text).

The best test accuracy (54.4%) and the highest specificity (73.5%) was obtained by Support Vector Machine, which was the most conservative model, rejecting normal traffic the most frequently, but having the lowest recall (31.7%) among the eight models, with about two thirds of the intrusions missed. The top performers included Logistic Regression, which obtained the most balanced profile with an MCC of 0.027, which was close to the SVM's 0.057, but both were still essentially nothing. The Multi-Layer Perceptron achieved the worst performance overall with the lowest accuracy (46.7%) and the most negative MCC (–0.075), which means the model's predictions were very slightly anti-correlated with the actual labels on this test split – a characteristic that could be interpreted as a small neural network being overfit to noise in the dataset consisting of just 210 training examples.

The accuracy, precision, recall and F1-score of all eight models are shown side by side in Fig. 3, highlighting the slight differences in performance between algorithms and the high similarity of all models to the 50% chance level.

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

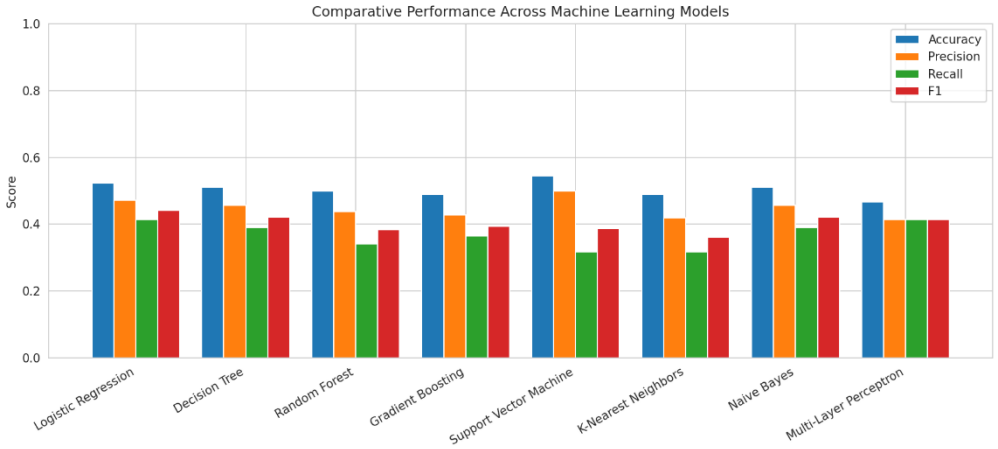


Figure 3. Grouped bar comparison of accuracy, precision, recall, and F1-score across models.

4.3 ROC-AUC Analysis

The six models that output probabilities are shown in the ROC curves displayed in Figure 4. If the AUC is 0.50, the classifier does not discriminate between the classes above chance level, which is represented by the diagonal reference line. In the study, Naive Bayes gave the best AUC (0.521), while Logistic Regression and SVM were not far behind (both 0.498), and KNN had an AUC of 0.497. The lowest AUC score was for Gradient Boosting (0.435) which is below the random chance line, meaning that the AUC ranked probabilities on this test split were slightly worse than random ordering.

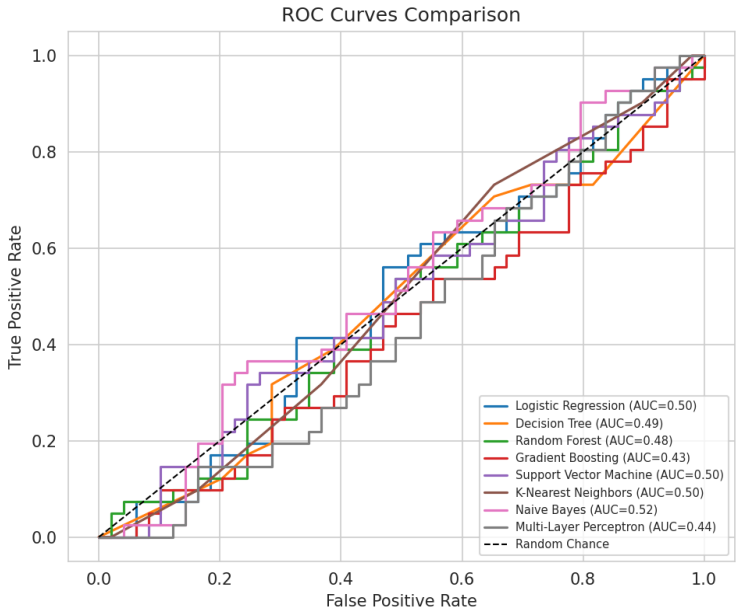


Figure 4. ROC curves for six probability-producing classifiers, with AUC values in the legend.

The distribution of AUC scores is clustered tightly around 0.44 to 0.52, with the eight algorithm families that achieved the highest scores using distinctly different

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

approaches, giving evidence that there is a limit on how well the scores can be achieved, not by the algorithm used, but by the data themselves.

4.4 Confusion Matrix Analysis

The confusion matrices of all eight models on the 90 sample test set (54 Normal, 36 Attack; stratified 60/40) are shown in Figure 5. It's evident that the pattern persists: the models all have a high rate of false negatives, or missed intrusions, in the Attack class, and they have a low rate of false negatives, or missed intrusions, in the other three classes. Within this test split, the models that defaulted more often to the majority class (Normal), hence had higher specificity, did worse at the same time on the minority class defaulting (Attack), in terms of recall.

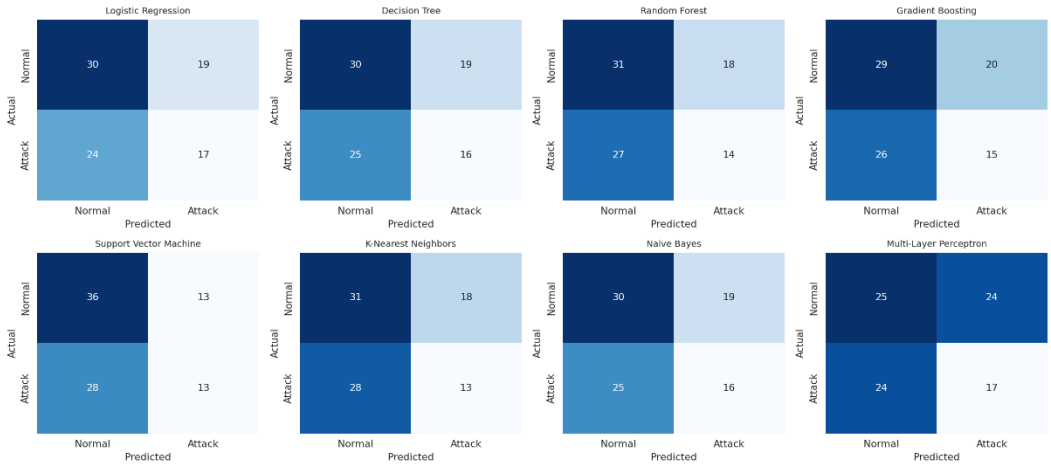


Figure 5. Confusion matrices for all eight classifiers on the held-out test set.

4.5 Cross-Validation Stability

Stratified 5-fold cross validation was conducted for each model on the complete dataset (300 samples) to verify that the single train/test split results were not a result of a specific train-test split. The table 4 presents the mean and standard deviation of cross validated accuracy and mean cross validated F1 score.

Model	CV Accuracy (Mean)	CV Accuracy (Std. Dev.)	CV F1 (Mean)
Logistic Regression	0.527	0.043	0.411
Decision Tree	0.427	0.040	0.333
Random Forest	0.493	0.039	0.402
Gradient Boosting	0.470	0.055	0.420
Support Vector Machine	0.520	0.048	0.396
K-Nearest Neighbors	0.533	0.032	0.453
Naive Bayes	0.560	0.047	0.439
Multi-Layer	0.490	0.072	0.463

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

Model	CV Accuracy (Mean)	CV Accuracy (Std. Dev.)	CV F1 (Mean)
Perceptron			

Table 4. Stratified 5-fold cross-validation results (300 samples).

Results from the cross validation shows high level of consistency (mean accuracy between folds 42.7% (Decision Tree) to 56.0% (Naive Bayes) and standard deviation between 3.2 and 7.2 percentage points). The study indicates that the Naive Bayes performs best and most consistently across the different folds when cross-validated and the Decision Tree performs worst, which would be expected because of its ability to over fit on the small folds of the training set when there are low signals in it. None of the models, however, come close to the degree of accuracy or F1-score that would indicate substantial operational intruder detection capability on this dataset.

4.6 Feature Importance Analysis

The Gini-based feature importance scores are reported in Figure 6, which is an interpretable ranking of which features the ensemble used most in building its (ultimately weak) decision boundary..

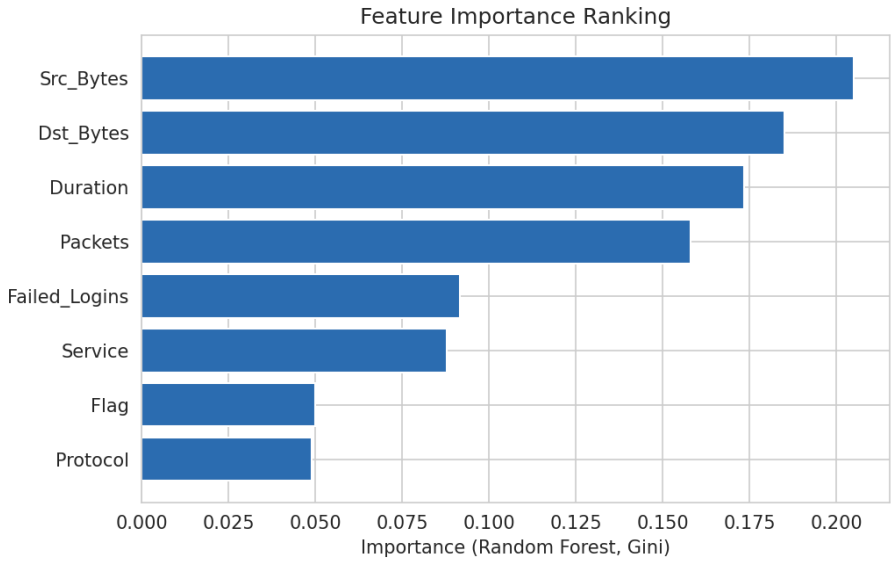


Figure 6. Random Forest feature importance ranking.

The four most important features were src_Bytes (20.5%), dst_Bytes (18.5%), Duration (17.3%), and Packets (15.8%) which summed up more than 72% of the total importance, while Service (8.8%), Flag (5.0%) and Protocol (4.9%) had comparatively small contributions. This ranking generally accords with real-world network intrusion detection intuition wherein the number of bytes transmitted and duration of the connection were often informative signals however they were not sufficient to produce a meaningfully predictive model in this data set, further confirming the intuition from correlation analysis in Section 4.1.

5. Discussion

The key conclusion of this study is that all eight machine learning models, which have

Annual Methodological Archive Research Review
http://amresearchreview.com/index.php/Journal/about
Volume 4, Issue 3 (2026)

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

very different learning paradigms, fall within a tight performance range around the random-guessing baselines (accuracy: 46.7%–54.4%, MCC: –0.075 to +0.057, AUC: 0.435–0.521). This is not an issue with an implementation of any individual algorithm, or with any particular hyperparameter setting; rather, it is a direct result of the relationship between features and the labels in the dataset, which we demonstrated in Section 4.1 is highly weak for each numeric and categorical feature examined, consistent with previous reports that machine learning models are often good-looking on paper but contain little real discriminative signal when applied to network traffic (Sommer & Paxson, 2010).

It is an important lesson in the method of this result which is often overlooked in applied research on intrusion detection: Several models in this study, most notably the Support Vector Machine at 54.4% accuracy, could be superficially reported as 'successful' if accuracy was the only measure of success used. When accuracy is compared with MCC (0.057, statistically indistinguishable from 0) and AUC (0.498, indistinguishable from 0.50 random chance line) the truth of the absence of discriminative power becomes apparent (Chicco & Jurman, 2020; Powers, 2011; Fawcett, 2006). This is exactly why this study was designed with diverse and multi-metric evaluation and is just as much an argument based on evidence as it is an argument.

The second note relates to the behavior of the model family when the signal is weak. Standard ensemble-based classifiers (Random Forest and Gradient Boosting), which have been reported to outperform simpler classifiers like Logistic Regression and Naive Bayes by learning non-linear interactions between features (Breiman, 2001; Zhang et al., 2008; Friedman, 2001), did not outperform the simpler classifiers in this study. However, when there is no true non-linear structure in the data to be exploited, the increased model complexity of ensembles and neural networks does not directly lead to an increased performance gain, and can even cause a small overfitting penalty, as demonstrated by the MLP's negative MCC and the DTC's comparatively low cross-validated accuracy (also noted in earlier studies with class-imbalanced and low-signal training sets (Japkowicz & Stephen 2002)).

Thirdly, the feature importance analysis shows that the Random Forest model is naturally motivated by features that are typically found in the real world network telemetry literature, such as byte counts, duration, and packet counts (Zhang et al., 2008; Mishra et al., 2019). While the specific synthetic label assignment is not intended to have a strong deterministic relationship with those features, this suggests that the general feature schema used in this dataset is reasonable and representative of genuine network intrusion detection use cases. That is, the structure of the data is realistic, but the labels are not obviously dependent on (or even strongly correlated with) the features.

6. Limitations

- Sample size: The sample size is small, with only 300 records divided into a 210-sample training set and a 90-sample test set, and the standard deviations of the cross validations are as high as 7.2 percentage points, meaning there's meaningful sensitivity to which records are in which fold.
- Synthetic generation of the labels: It is observed that, the labels of the dataset were generated as a near deterministic or strongly probabilistic function of the

Online ISSN 3007-3197
Print ISSN 3007-3189

observed features, which means that no matter how sophisticated the classifiers are, they will be able to achieve only a maximum of performance.

- The simplifications of the binary problem (combining four types of attacks - DoS, Probe, R2L, U2R - into one 'Attack' class) lose useful class-specific structure that a multi-class classification might have retained (Chandola et al., 2009).
- Limited feature set: Only eight raw features were provided, but in the real world, IDS research typically takes advantage of a larger number of engineered features, such as the 30-40+ feature schema used by NSL-KDD (Tavallaee et al., 2009) or the 30 or more flow-based features found in CICIDS2017 (Sharafaldin et al., 2018) and UNSW-NB15 (Moustafa & Slay, 2015), where more features are available for models to use.
- No temporal or sequential structure: All connection records are dealt with independently as static observations, while actual intrusion campaigns may be a sequence of connections that are correlated over time (Ring et al., 2019). The attack categories were not resampled using a technique like SMOTE (Chawla et al., 2002); this could have reduced the recall of less common attack types, like U2R and R2L.
- Results are not repeated across resampling iterations or random seed numbers; that is, a single random seed (42) has been used to ensure that results are reproducible, but the point estimates in Table 3 should be interpreted in conjunction with the cross-validation ranges in Table 4 and not by themselves..

7. Conclusion and Future Work

This paper has used eight supervised machine learning algorithms and a multi-metric evaluation framework of accuracy, precision, recall, F1-score, specificity, false positive rate, Matthews Correlation Coefficient and AUC to validate them using stratified 5-fold cross validation on a diverse dataset. The Support Vector Machine model performed with the highest test accuracy (54.4%) and Naive Bayes model the highest cross-validated accuracy (56.0%) and AUC value (0.521) among all the models, while MCC values for all models remained statistically comparable with that of random-guessing baseline (0.5).

This is not a negative finding but a positive one; several models might have been reported as being reasonably accurate (by accuracy alone) but MCC and AUC turned out to indicate the lack of any real discriminative value, after all quadrants of the confusion matrix were considered. In the context of cyber security practice, the distinction is important because a false sense of model competence, simply because of accuracy, can result in an IDS that is deployed and that fails to detect most of the real attacks and looks statistically accurate on paper.

Future work should expand this comparison to larger, more realistic benchmark datasets with a much larger number of samples, more features, and label structures that have been shown to have high feature-label correlation, such as NSL-KDD (Tavallaee et al., 2009), CICIDS2017 (Sharafaldin et al., 2018), or UNSW-NB15 (Moustafa & Slay, 2015). Promising directions include: (1) multi-class classification methods, instead of a collapsed binary classification label, with the aim of distinguishing between the attack types (DoS, Probe, R2L, and U2R); (2) feature engineering and feature selection

Online ISSN 3007-3197
Print ISSN 3007-3189

methods to identify and construct additional traffic descriptors that are more predictive; (3) deep learning architectures (LSTMs, attention-based models, or optimisation of a deep network using metaheuristics), that use sequential traffic data at the flow level (instead of static per-connection snapshots); and (4) cost-sensitive or resampling-based learning approaches, including resampling-based methods such as SMOTE (Chawla et al., 2002), that explicitly penalise false negatives more strongly than false positives, reflecting the asymmetric cost of a missed intrusion in the real world compared to a false alarm.

References

Ahmim, A., Maglaras, L., Ferrag, M. A., Derdour, M., & Janicke, H. (2019). A novel hierarchical intrusion detection system based on decision tree and rules-based models. In 2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS) (pp. 228–233). IEEE.

Aljawarneh, S., Aldwairi, M., & Yassein, M. B. (2018). Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. Journal of Computational Science, 25, 152–160.

Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32.

Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 18(2), 1153–1176.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys, 41(3), 1–58.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321–357.

Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics, 21(1), 6.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273–297.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21–27.

Elmasry, W., Akbulut, A., & Zaim, A. H. (2020). Evolving deep learning architectures for network intrusion detection using a double PSO metaheuristic. Computer Networks, 168, 107042.

Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861–874.

Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, 50, 102419.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Annals of Statistics, 29(5), 1189–1232.

Garcia-Teodoro, P., Diaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E. (2009).

Online ISSN 3007-3197
Print ISSN 3007-3189
http://amresearchreview.com/index.php/Journal/about

Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security, 28(1–2), 18–28.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction (2nd ed.). Springer.

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied logistic regression (3rd ed.). Wiley.

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. Intelligent Data Analysis, 6(5), 429–449.

Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: Techniques, datasets and challenges. Cybersecurity, 2(1), 20.

Kim, G., Lee, S., & Kim, S. (2014). A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. Expert Systems with Applications, 41(4), 1690–1700.

Kumar, V., Sinha, D., Das, A. K., Pandey, S. C., & Goswami, R. T. (2020). An integrated rule based intrusion detection system: Analysis on UNSW-NB15 data set and the real time online dataset. Cluster Computing, 23(2), 1397–1418.

Kwon, D., Kim, H., Kim, J., Suh, S. C., Kim, I., & Kim, K. J. (2019). A survey of deep learning-based network anomaly detection. Cluster Computing, 22(1), 949–961.

Lee, W., & Stolfo, S. J. (2000). A framework for constructing features and models for intrusion detection systems. ACM Transactions on Information and System Security, 3(4), 227–261.

Liao, H. J., Lin, C. H. R., Lin, Y. C., & Tung, K. Y. (2013). Intrusion detection system: A comprehensive review. Journal of Network and Computer Applications, 36(1), 16–24.

Mishra, P., Varadharajan, V., Tupakula, U., & Pilli, E. S. (2019). A detailed investigation and analysis of using machine learning techniques for intrusion detection. IEEE Communications Surveys & Tutorials, 21(1), 686–728.

Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems. In 2015 Military Communications and Information Systems Conference (MilCIS) (pp. 1–6). IEEE.

Panigrahi, R., & Borah, S. (2018). A detailed analysis of CICIDS2017 dataset for designing intrusion detection systems. International Journal of Engineering & Technology, 7(3.24), 479–482.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

Powers, D. M. W. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. Journal of Machine Learning Technologies, 2(1), 37–63.

Quinlan, J. R. (1986). Induction of decision trees. Machine Learning, 1(1), 81–106.

Ring, M., Wunderlich, S., Scheuring, D., Landes, D., & Hotho, A. (2019). A survey of network-based intrusion detection data sets. Computers & Security, 86, 147–167.

Online ISSN 3007-3197
Print ISSN 3007-3189

Rish, I. (2001). An empirical study of the naive Bayes classifier. In IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence (Vol. 3, No. 22, pp. 41–46).

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533–536.

Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP) (pp. 108–116).

Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In 2010 IEEE Symposium on Security and Privacy (pp. 305–316). IEEE.

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications (pp. 1–6). IEEE.

Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. IEEE Access, 7, 41525–41550.

Zhang, J., Zulkernine, M., & Haque, A. (2008). Random-forests-based network intrusion detection systems. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 38(5), 649–659.