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

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

Predictive Network Security Assessment using Risk Analysis and Random Forest

Ravi N, Inban V N, Dhaarun Kumar K, Dinesh A C, Harishankar P, Raja Boopathy R
Department of Information Technology, PSG Polytechnic College, Coimbatore, India
RF-NRA: Random Forest-Based Network Risk Assessment

Abstract**—** *Traditional IDS can only classify the network traffic as either legitimate or malicious traffic without any significant information about how the security risk is progressing in the network before being successfully breached. There are many signs before the breach of the network like irregularities in communication, connection behavior, and repeated attempts at access. In this research paper, we have introduced RF-NRA (Random Forest Based Network Risk Assessment) which has reformulated the problem of network traffic classification as ordinal and progressive risk state estimation problem with the use of Random Forest. Six risk states of the network traffic based on intrusion categories (Normal, Probe, DoS, R2L, and U2R) are introduced, namely; SAFE, LOW RISK, MODERATE RISK, HIGH RISK, CRITICAL, and COMPROMISED. The system is trained and tested on the NSL-KDD dataset, obtaining a mean macro-F1 score on 5-fold cross-validation of 0.92 and a test set accuracy of 73.3%. We also report hyper-parameter sensitivity analysis, tier-wise ROC-AUC metrics, and the system architecture for online implementation. Our evaluation shows that there is a large generalization gap due to the presence of attack types exclusive to the test set, providing an empirical observation of the limitations of supervised classification methods in terms of their performance on novel attacks. Our ordinal risk-tier formulation, rather than increased accuracy, is our key contribution here.*

Keywords**—** *Random Forest, Intrusion Detection, Network Security, Risk Assessment, Machine Learning, NSL-KDD, Ordinal Classification, Early Warning Systems, Ensemble Learning, Cybersecurity, ROC-AUC, Hyperparameter Tuning*

I. INTRODUCTION

With the rise in the number and complexity of cyberattacks, early detection of any network attack has become an important criterion of the current security infrastructure. Enterprises, governments, and critical infrastructure providers are in a constantly changing environment of threats that use reconnaissance, credentials compromise, lateral movement, and privilege escalation as different phases in the cyberattack process. The main principle of Intrusion Detection Systems (IDS) has always been focused on answering a particular yes or no question about network traffic: is it malicious or not? Although this framework has brought much progress in terms of detecting malicious attacks during the last two decades, it offers little assistance in understanding the process of changes in network's security posture.

In reality, a breach will never occur instantly. Instead, it will often be precluded by a series of lower severity warnings — reconnaissance scanning, multiple authentication failures, anomalous service accesses, unusual data transfer levels — which in themselves would not constitute a cause for concern, but in combination signify an increasing threat level. SOC operators have long reported alert fatigue as one of their chief concerns: with every incident flagged as an alarm regardless of its severity level, identifying those threats which escalate becomes increasingly difficult. The current IDS framework is simply incapable of highlighting such an escalation, since it flags off any single communication in terms of a predefined attack vector; and more importantly, does not allow for an evaluation of the threat level with regards to how close a network is to compromise.

The present paper presents RF-NRA (Random Forest Based Network Risk Assessment), a framework which recasts the classification problem in its entirety. Instead of classifying based on the nominal label of an attack type, RF-NRA classifies into one of six possible risk levels, namely, SAFE, LOW RISK, MODERATE RISK, HIGH RISK, CRITICAL, and COMPROMISED, which are derived from recognized intrusion categories. Through such a recasting, RF-NRA is able to leverage the known capabilities of Random Forest as a classifier, including noise immunity and inherent overfitting protection, without the native capability of a nominal classifier for risk trajectories.

The central research questions addressed in this work are fourfold. First, can an ordinal risk-tier reformulation of network traffic classification be learned effectively by a Random Forest ensemble using standard, publicly available intrusion detection features? Second, how does the resulting model's error behaviour differ when evaluated on an ordinal distance metric rather than simple classification accuracy? Third, how sensitive is the model's performance to core hyperparameters such as forest size and tree depth, and does this sensitivity differ across risk tiers? Fourth, what are the practical limitations of this approach when the classifier encounters attack types it has not previously observed, and what does this imply for real-world deployment?

There are four research questions that form the basis of this study. The first research question focuses on whether an ordinal risk-tier re-classification of network traffic classification can be effectively learnt using a Random Forest classifier by applying publicly accessible features of intrusion detection. The second research question revolves around the behaviour of error in the classifier model in terms of its evaluation on an ordinal distance measure as opposed to classification accuracy. The third research question is on the sensitivity of the model performance to important hyperparameters like forest size and tree depth for each risk tier.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

II. RELATED WORK

A. Random Forest for Intrusion Detection

The technique of random forest has been successfully used for detecting intrusion in networks because of the technique's strength against over-fitting and the capability to deal with high-dimensional data of traffic without the need for much pre-processing. According to Zhang and Zulkernine [1], random forest is one of the first techniques that have been used systematically for detecting misuse, anomalies, and hybrid detection of intrusions by using the algorithm's inherent capacity of detecting outliers (distance measure between the points). They used random forests as a promising method compared to support vector machines and neural networks in terms of computation expense and interpretation.

Further research has primarily concentrated on three main strategies of increasing classification accuracy. First, there is dimensionality reduction and feature optimization, whereby more accurate Random Forest methods incorporating Bald Eagle Search optimization along with Kernel Principal Component Analysis [2] and Pearson correlation-based feature selection [3] have yielded an increase in accuracy of two to six percent compared to the traditional Random Forests baseline. The reason behind this approach is that many features of NSL-KDD and other datasets are highly correlated (e.g., srv_serror_rate and serror_rate).

The next method is the use of an ensemble or hybrid approach. There have been several works that have combined Random Forest with other classifiers such as Support Vector Machines [4],[13], Gradient Boosting and XGBoost [8], and also Decision Trees compared to Random Forest [9]. The results show marginal improvement in accuracy performance of the classification of attacks on benchmark datasets such as KDD'99, NSL-KDD, and UNSW-NB15. It is noted that hybrid approach has yielded only small improvements on accuracy beyond one percent point in these benchmark datasets.

Thirdly, there is the problem of class imbalance that has remained a major issue in intrusion detection data sets, where the number of instances belonging to a benign class significantly exceeds the number of attacks including the rare U2R category attacks. ADASYN sampling in combination with Random Forest [6] has been used to address this problem of class imbalance while the use of SMOTE-based sampling has been used for intrusion detection in wireless sensor networks [11], where both studies have resulted in improvements in accuracy to about 90 to 93 percent for CICIDS2017 and WSN intrusion detection respectively. However, improvements in the latter cases have been based on overall or class accuracies only and not on classes unseen during training.

In recent years, research trends have moved towards achieving real-time capabilities. The authors of IDRandom-Forest [12] suggest using accuracy-sliding-window feature weighting to decrease inference delay for a real-time implementation of the algorithm on high-speed network connections, claiming inference time lower than a millisecond per flow. Random Forest at multiple levels with fuzzy inference [7] has been suggested for use in IoT networks intrusion detection for which model size and inference efficiency matter as much as accuracy. An easy-to-use Random Forest-based algorithm for cybersecurity intrusion detection [10] highlights ease of deployment and minimum hyper-parameter tuning as significant benefits over sophisticated ensemble methods. Importantly, none of these works describes a systematic study of sensitivity of forest size and tree depth to generalization ability on unseen attacks, a problem that is partly addressed in this paper in Section VI-F.

B. The Common Limitation: Binary or Nominal Classification

Within the methods reviewed in references [1] through [13], the output of the system in each of these cases is formally identical in nature: either a binary classification of normal or attack, or else the nominal classification of which discrete type of attack took place. The precision, recall, F1-score, and AUC are computed per class or in an overall aggregate manner; however, none of these evaluation metrics convey how close to being compromised a network is at any given time, nor do they lend themselves to the idea of one security state being between two other states on some scale. Both the Probe attack and the Denial of Service attack are simply different classes and not different points along the same continuum of risk, despite the fact that any security analyst would concede the former represents far less of a threat than the latter.

This structural issue carries a very practical implication. In the context of a nominal classification setting, the penalty for classifying the Probe attack as a User-to-Root attack is identical to that of classifying the Probe attack as Normal traffic since both are considered just "wrong." However, viewed from a security operations perspective, these two cases are not equal – the first case involves over-escalation while the latter involves a potentially damaging mistake. Traditional multi-class metrics are not able to discriminate between these two kinds of errors.

C. Risk-Scoring Approaches in Cybersecurity Machine Learning

A separate literature has focused on continuous or graded approaches to cybersecurity risk assessment. Risk prediction of cybersecurity attacks [14] has studied both detection of an attack occurrence and classification of the network risk level associated with such an attack based on quantitative approaches to risk level assessment that include host security, application security, and data security components. It has been shown that risk level labels can be generated from the current security monitoring tools, thus following the methodology of label generation adopted in the current study.

Random Forest, XGBoost, SVM, and Logistic Regression comparative studies on critical infrastructure risk data [15] have demonstrated high discrimination of Random Forest and XGBoost with Area Under ROC Curve value being approximately equal to 0.88, emphasizing once again suitability of Random Forest as a classifier for risk-based task instead of classification-based one. Still, this line of research is focused on solving the risk estimation as the probability calibration problem performed on top of the binary classifier; that is, the

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

risk value is treated as an additional feature that needs to be estimated from the output probabilities of the binary classifier.

D. Research Gap and Positioning

None of the reviewed papers [1] - [15] produce an ordinal risk state, such as SAFE → LOW RISK → MODERATE RISK → HIGH RISK → CRITICAL → COMPROMISED. Current solutions are trying to solve the problem of "Is there a malicious traffic and what kind?" — but no one solves the problem of "How far is the current state of the network from being compromised?" Operationally speaking, the former will be useful for the network analyst, while the latter will provide him with valuable information about the evolving state of the network.

This is precisely what the RF-NRA approach does; it reimagines the target for Random Forest classification as being an ordinal scale representing risk, which is directly converted from existing and well-known categories of intrusions (Normal, Probe, DoS, R2L, U2R). In this way, RF-NRA neither creates a new base algorithm nor develops a new set of features; instead, it makes an ordinal transformation the only variable being analyzed in the research in order to be able to attribute any results to the ordinal reformulation specifically.

III. BACKGROUND: ENSEMBLE LEARNING AND RANDOM FOREST

This part introduces the theory needed to grasp the design decisions made in Section IV. Those who are well-versed in the concepts of decision trees and ensembles can skip directly to Section IV.

A. Decision Trees

A decision tree is an hierarchical classification approach based on rules, which recursively divides the feature space into subspaces for which a class label can be assigned. At each internal node, the decision tree finds the best criterion based on a certain impurity measure to split the data instances at that node. The Gini impurity for a node having data from K classes is given by:

$$Gini = 1 - \Sigma(p_k)^2, \ for\ k = 1\ to\ K$$

where p_k is the fraction of instances that belong to class k in that node. The optimal split is made on the basis of the highest decrease in weighted Gini impurity from the parent node to its two children nodes. This goes on until a certain stop condition is reached, which can be a maximum tree depth, minimum number of samples required to split a node, or if the node becomes pure (all instances belonging to one class). Decision trees, despite being very interpretable, tend to overfit due to their capacity to memorize training set noises.

B. Bootstrap Aggregation (Bagging)

Random Forest tackles the problem of high variance associated with single decision trees through bagging, which stands for bootstrap aggregation. For a dataset of size M, bagging creates N datasets, also of size M, by random sampling with replacement. Since the sampling procedure is carried out with replacement, each bootstrapped dataset usually consists of about 63.2 percent of the original examples while the other 36.8 percent of the examples are repeats; these examples that do not appear in a particular bootstrapped dataset are considered to be the out-of-bag examples of that particular tree and can serve as an internal validation tool without a separate test set. Each of the resulting bootstrapped datasets is used to train a different decision tree, and the predictions are combined into one using a majority vote in the case of classification problems, such as in this paper.

C. Random Feature Selection

Random Forest augments naive bagging with an extra element of randomness in that during the construction of each tree in the forest, only a random subset of the available features (the square root of the total number of features, in the case of classification problems) is used to decide upon which feature should be used in the split. The effect of this approach is that the individual trees of the ensemble become uncorrelated with one another. The reason is that without this element of randomness, if there is one or two very informative features among the set of features, then most of the bagged trees will choose the same feature for their first split; and hence, become highly correlated with one another with the result that their errors will be correlated too.

D. Why Random Forest Suits This Problem

The Random Forest algorithm was chosen as the base classifier for RF-NRA on four counts that are specific to the ordinal reformulation of the risk tiers. Firstly, the algorithm allows multi-class classification out-of-the-box with no need for reduction to the one-versus-rest scheme, in six classes. Secondly, the algorithm's voting per tree leads to class probability estimates – the ratio of votes for each class – at virtually zero cost, and is exploited in this paper not only as the metric of confidence, but also as the criterion for early warning escalation detailed in Section IV-D. Thirdly, the built-in feature importance measurement through the mean decrease in impurity of the splits made by each feature in all trees allows direct interpretation of the security-related results, as detailed in Section VII-C, without the necessity of using an additional explainability method such as SHAP or LIME. Fourthly, the algorithm's established accuracy on the problem [1]–[13] provides a benchmark for measuring the effects of the ordinal reformulation, in accordance with Section II-D. This makes Random Forest well suited for the proposed risk classification. It also enables reliable classification of network traffic across the defined risk levels.

IV. PROPOSED METHODOLOGY AND SYSTEM ARCHITECTURE

The RF-NRA approach realizes network security evaluation in a four-step framework that involves feature extraction, risk level labeling, Random Forest classification, and stepwise state evaluation and early warning escalation. Fig. 1 shows the architecture of the complete system; further details about each of these steps follow.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

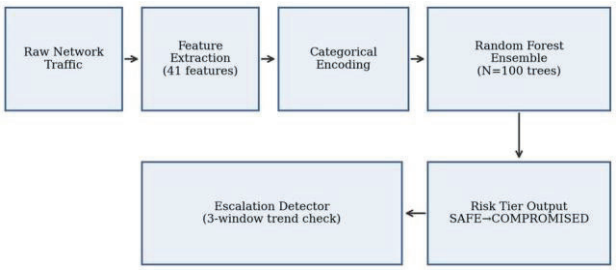


Fig. 1. RF-NRA system architecture and processing pipeline.

A. Feature Extraction

The raw network data is captured as connection tuples with certain characteristics of duration, number of bytes, connection behavior, and service access behavior. The 41 basic attributes defined in the NSL-KDD dataset have been used in this study, divided into four distinct groups as shown in Table I. The detailed description of all the features is presented in Appendix A.

TABLE I. NSL-KDD FEATURE CATEGORIES

Category	Description	Example Features
Basic	Characteristics of a single connection in TCP/IP	duration, protocol_type, service, flag, src_bytes, dst_bytes
Content	Characteristics indicative of questionable conduct within the link	num_failed_logins, root_shell, num_compromised, su_attempted
Traffic (time-based)	Statistics for connections during a 2 second window to the same host	count, srv_count, serror_rate, same_srv_rate
Host-based traffic	Statistics on the past 100 connections to the identical destination host	dst_host_count, dst_host_srv_count, dst_host_serror_rate

Categorical variables (protocol_type, service, flag) have been label-encoded into integers before being fed to the model because of the requirement of a numeric feature set by the Random Forest algorithm which is similar to most other scikit-learn estimators. No additional feature engineering was performed outside of the raw feature set in the dataset, and this decision was made to isolate the effect of the tiered-risk representation discussed in Section IV-B.

One of the practical issues related to the label encoding of the service attribute is that there are some service categories seen in KDDTest+ that cannot be found in KDDTrain+ (such as certain rare application-level services corresponding to the novel attacks introduced in Section V-C). Instead of ignoring such instances or having the process fail at the inference phase, such unknown category instances are assigned a unique "unknown" integer value different from the integer codes used during the training phase. This is a very conservative decision since it guarantees that all instances will be processed without exception, but at the same time, it guarantees that the unknown service will be considered categorically different from the rest of the known services.

B. Risk-Tier Label Construction

The key methodological innovation of RF-NRA is the conversion of attack labels into an ordinal risk ranking. The set of attacks is ranked on a six-step scale depending on the degree to which the attacker can control the target system; this hierarchy is provided in Table II (V) and in Appendix B.

The justification for the map is based on standard intrusion classification taxonomy. A probe intrusion is reconnaissance without gaining access to the system and has two sub-categories depending on the purpose for which they were undertaken: LOW RISK for generic scanning where there is low cost and effort involved with most such attacks being automatic scans (ipsweep, nmap) and MODERATE RISK for focused scanning (satan, portsweep, mscan, saint) indicating reconnaissance of a particular target. DoS intrusion does not cause breach of confidentiality and integrity and just impacts availability. It is classified as HIGH RISK. Remote-to-Local intrusions imply that the attacker has gained unauthorized access remotely to a system to which he did not have prior access rights and this is classified as CRITICAL. User-to-Root intrusions indicate that the attacker already had access in some form and escalated this access to root/administrator rights and is classified as COMPROMISED since at this stage the network can no longer be considered to be under exclusive control of the defender.

The transformation transforms the nominal, unordered classification task into an ordinal one such that not only can class match be evaluated but distance can also be used in the evaluation process as discussed in Section VI-D — an incorrect classification that is one step away on the risk spectrum is considered to have committed an error that is of significantly less importance than an incorrect classification that is several steps away.

C. Random Forest Classification

With the feature vector labels generated using Sections IV-A and IV-B, N decision trees are learned using the conventional procedure for Random Forest algorithm presented in Algorithm 1.

Algorithm 1: RF-NRA Training and Inference

```
Input: Labeled dataset D = {(x_i, y_i)}, y_i in
{0,...,5}
          (risk tiers SAFE..COMPROMISED), forest
size N
Output: Trained ensemble M = {T_1, ..., T_N}

for n = 1 to N do
    D_n <- BootstrapSample(D)           // sample
|D| with replacement
    T_n <- GrowTree(D_n)
        at each node:
            F_sub <- RandomSubset(features,
size=sqrt(|F|))
            split on (f, t) in F_sub minimizing
weighted Gini
        until stopping criterion (purity / min
leaf size)
end for
M <- {T_1, ..., T_N}

// Inference on a new traffic record x
```


(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

```
for n = 1 to N do
    vote_n <- T_n.predict(x)
end for
y_hat <- MajorityVote(vote_1, ..., vote_N)
P(class=k | x) <- (#votes for k) / N    for k =
0..5
```

During the inference phase, each tree individually produces a risk category assignment for a particular traffic sample, while the ensemble classification result is given by the majority decision from all N trees; the probabilities of belonging to different classes can also be provided through the fraction of trees that vote for each risk category, which gives another confidence measurement in addition to the classification result itself. In our study, we set N = 100 trees as the default setting, similarly to the usual settings in literature [1], [9], [13].

For class balancing, the "balanced" option was used for class_weights while training, thereby assigning weightage to the class inversely proportional to the proportion of that class in the dataset. This strategy was used particularly to overcome the highly imbalanced classes that are prevalent in intrusion datasets, explained in quantitative form in Section V below, where SAFE classes have more instances than COMPROMISED classes by a ratio of more than a thousand to one in the training dataset.

D. Progressive State Determination and Early-Warning Escalation

The classifier's prediction for a traffic observation window is assumed to be the risk state of the network at the moment. States being ordinal, RF-NRA also takes note of state sequences over time in accordance with Algorithm 2, where three consecutive windows that exhibit strictly rising risk states are categorized as "ESCALATING," resulting in an early warning regardless of whether any individual window is deemed high-risk by itself.

Algorithm 2: Early-Warning Escalation Detection

```
Input: sequence of predicted tiers [s_(t-2), s_(t-
1), s_t]
Output: Alert (boolean), Trend label


if s_(t-2) < s_(t-1) < s_t then
    Trend <- "ESCALATING";  Alert <- TRUE
else if s_t in {HIGH_RISK, CRITICAL, COMPROMISED}
then
    Alert <- TRUE;  Trend <- "ELEVATED"
else
    Alert <- FALSE; Trend <- "STABLE"
end if
```

This approach ensures that the buildup of a multi-stage attack process can be detected as a cohesive pattern from one observation window to another without necessarily having to detect a particular attack stage in one specific observation window where the severity is above a certain threshold, which was highlighted as a weakness in the traditional IDS architecture in Section II-B.

E. Implementation Stack

The reference implementation was done using Python 3 programming language, where the scikit-learn library was used to develop the Random Forest classifier, feature encoding, cross-validation, and evaluation metrics. Other libraries used were pandas library, which is used to handle tabular datasets, and joblib library, which is used to serialize the trained models. These tools were selected for the reference implementation due to their broad usage in academic literature on intrusion detection systems and industrial machine learning pipelines, which facilitates the process of transitioning from a research prototype to a production system in the future.

V. DATASET AND EXPERIMENTAL SETUP

A. Dataset

The designed algorithm was tested using the NSL-KDD benchmark dataset provided by the Canadian Institute for Cybersecurity. The NSL-KDD is an improved dataset relative to the KDD Cup 1999 because it eliminates the duplications that are known to occur in the original data. This is to ensure that there are no biases when developing a classifier since the latter can be trained to memorize the commonly duplicated data, thereby achieving increased accuracy. The dataset contains two predefined subsets; one of them is KDDTrain+ with 125,973 labeled connection records and the other is KDDTest+ with 22,544 labeled connection records. Both subsets contain 41 attributes along with attack type and difficulty level (not used in this work). NSL-KDD was chosen over other datasets like CICIDS2017 due to its small size and detailed documentation of its features in the reviewed literature [1] – [15].

B. Risk-Tier Mapping

TABLE II. RISK TIER MAPPING FROM NSL-KDD ATTACK CATEGORIES

Risk Tier	Category	Attack Types
SAFE	Normal	normal
LOW RISK	Light probing	ipsweep, nmap
MODERATE RISK	Targeted probing	satan, portsweep, mscan, saint
HIGH RISK	DoS	back, land, neptune, pod, smurf, teardrop, apache2, udpstorm, processtable, mailbomb, worm
CRITICAL	R2L	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster, xlock, xsnoop, snmpguess, snmpgetattack, httptunnel, sendmail, named
COMPROMISED	U2R	buffer_overflow, loadmodule, perl, rootkit, ps, sqlattack, xterm

C. Class Distribution

Using the mapping defined in Table II on both dataset partitions results in the following risk tier distribution, illustrated in Fig. 2. There are two main characteristics of the distribution that should be pointed out in light of the results presented in Section VI. Firstly, the distribution is quite imbalanced in both partitions; SAFE and HIGH RISK tiers prevail and there are only 52 training samples labeled as COMPROMISED out of 125,973 total samples. Secondly, but much more importantly, there are 17 attack types included into the KDDTest+ set, which are not present at all in the KDDTrain+ dataset (such as snmpguess, sendmail, named, httptunnel, mailbomb, apache2, worm, processtable, udpstorm,

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

and many others), the vast majority of which belongs to the CRITICAL tier. It is an inherent feature of the test partition of the NSL-KDD dataset designed to check the generalization capabilities of a classifier to novel attacks rather than learning the training distribution.

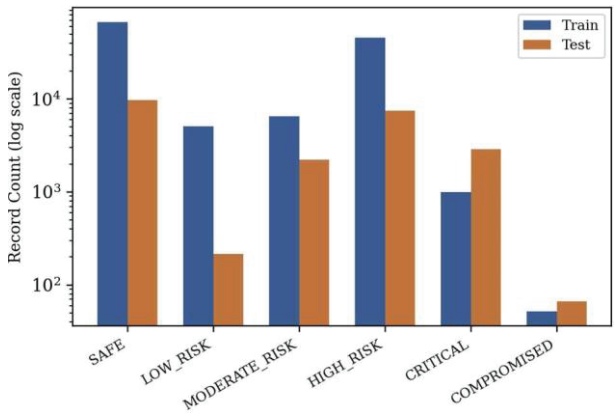


Fig. 2. Risk-tier class distribution, training vs. test partition (log scale).

D. Model Configuration and Environment

Parameter	Value
Classifier	RandomForestClassifier (scikit-learn)
Number of trees (N)	100 (default; see Section VI-F)
Class weighting	balanced
Max tree depth	None (grown to purity; see Section VI-F)
Feature subset per split	sqrt(num_features)
Random seed	42
Cross-validation	5-fold, stratified (3-fold for sweeps)

The validation process involves 5-fold stratified cross-validation on the training dataset and evaluation on the held-out KDDTest+ subset, following the standard benchmark process used in all the references [1]–[15]. The stratified folds will ensure that the class distribution within each fold is close to that in the overall training set, especially considering the significant imbalance discussed in Section V-C. In the hyperparameter sensitivity tests of Section VI-F, 3-fold stratified cross-validation was employed in lieu of 5-fold cross-validation for the purpose of reducing computational cost.

VI. RESULTS

A. Cross-Validation Performance

Five-fold stratified cross validation performed on the training dataset yielded a mean macro F1 score of 0.9208 (standard deviation ± 0.0269) among all folds, showing high internal consistency and absence of overfitting towards any one particular fold of the training dataset.

B. Test Set Performance

TABLE III. OVERALL TEST SET METRICS

Metric	Value
Accuracy	0.7332
Macro Precision	0.7095
Macro Recall	0.5447
Macro F1	0.5192

C. Per-Tier Performance

TABLE IV. PER-TIER PRECISION, RECALL, F1

Risk Tier	Precision	Recall	F1	Support
SAFE	0.63	0.97	0.77	9,711
LOW RISK	0.68	0.99	0.81	214
MODERATE RISK	0.83	0.50	0.63	2,207
HIGH RISK	0.96	0.77	0.86	7,460
CRITICAL	0.75	0.00	0.00	2,885
COMPROMISED	0.40	0.03	0.06	67

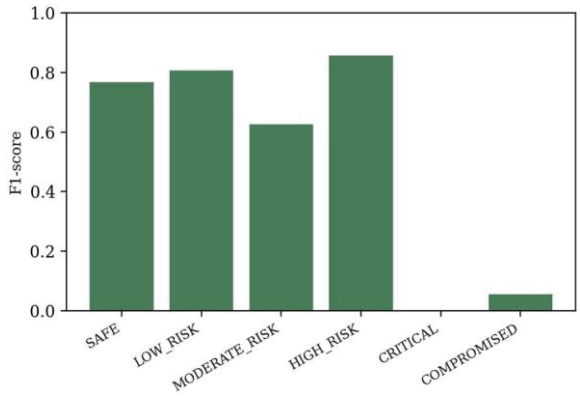


Fig. 3. F1-score by risk tier on the held-out test set.

D. Confusion Matrix and Ordinal Error Analysis

Figure 4 depicts the complete confusion matrix in the test dataset. The fact that risk categories are ordered means that, besides calculating the confusion matrix, we calculated the absolute difference between the predicted risk category and the actual risk category in every single test data point. In other words, the difference was 0 if the prediction matched the actual value and 5 if there was the greatest possible difference (SAFE categorized as COMPROMISED or vice versa). An exact match occurred in 73.32% of all test samples; 74.60% were correct up to 1 risk category, while 25.40% of them were more than 2 categories off.

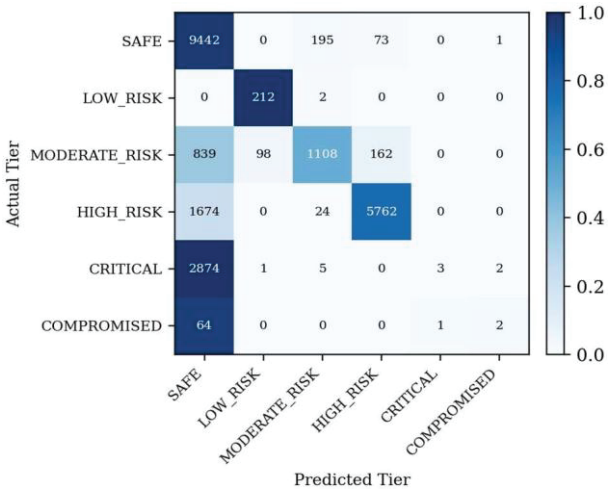


Fig. 4. Confusion matrix on the NSL-KDD test set (raw counts, colour-normalized per row).

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

E. Feature Importance

Figure 5 illustrates the fifteen most significant attributes based on the random forest mean decrease in impurity importance ranking technique. The five most significant attributes include src_bytes, dst_host_srv_count, dst_bytes, dst_host_diff_srv_rate, and service (encoded), which suggest that the most significant attributes are traffic volume and diverse access to services, as was previously found by other RF-IDS studies [1], [5].

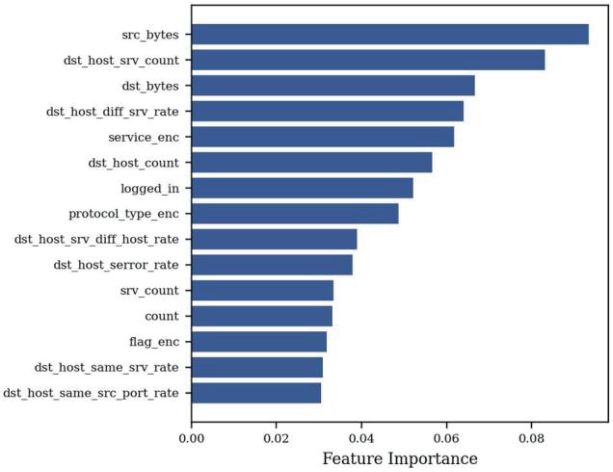


Fig. 5. Top 15 feature importances (mean decrease in Gini impurity).

F. Hyperparameter Sensitivity Analysis

In order to explain the choice of the default setup employed in Sections IV-C and V-D, as well as the sensitivity of the model to the most critical hyperparameters, a systematic sweep over the number of trees (N) and the maximum tree depth has been performed while keeping other parameters constant. Table V and Fig. 6 show the cross-validation macro F1 and test-set accuracy results for $N \in \{10, 25, 50, 100, 150, 200\}$.

TABLE V. SENSITIVITY TO FOREST SIZE (N)

N (trees)	CV Macro F1	Test Accuracy
10	0.9054	0.7385
25	0.9047	0.7333
50	0.9129	0.7325
100	0.9153	0.7332
150	0.9153	0.7330
200	0.9129	0.7329

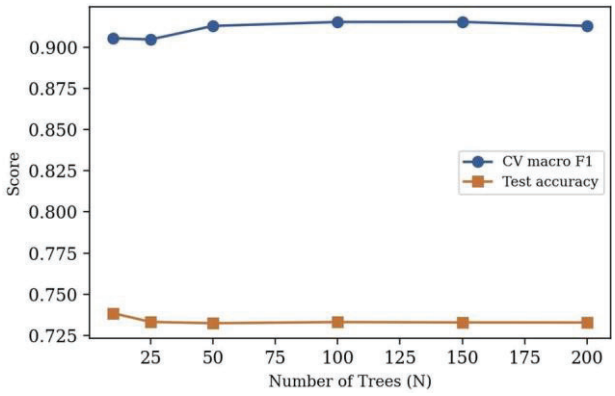


Fig. 6. CV macro F1 and test accuracy as a function of forest size N.

Cross-validation macro F1 gradually increases from N=10 up until N=100–150, where it levels off and decreases slightly at N=200, as expected from the well-known characteristic that Random Forest performance ceases to improve beyond the point where variance reduction is fully achieved because of sufficient size of the model ensemble, with extra trees only increasing computation time after that. In particular, test set accuracy appears fairly stable with respect to N throughout the entire range (fluctuating around 0.7325 and 0.7385), which strengthens the conclusion from section VII-A that the main cause of test set errors is the generalization error on the CRITICAL tier and not ensemble underfitting, which cannot be solved through growing the forest.

Another sweep for max_depth parameter on the same range produces a similar graph as shown in Table VI and Fig. 7. In particular, shallow trees perform noticeably worse which implies that six-tier ordinal decision boundary needs relatively deep trees in order to be learned, whereas depths larger than roughly 20 provide no additional benefit, hence making default parameter choice a good and close-to-optimal one.

TABLE VI. SENSITIVITY TO MAXIMUM TREE DEPTH

Max Depth	CV Macro F1	Test Accuracy
5	0.6943	0.7442
10	0.9047	0.7411
20	0.9158	0.7321
30	0.9133	0.7335
None (unlimited)	0.9153	0.7332

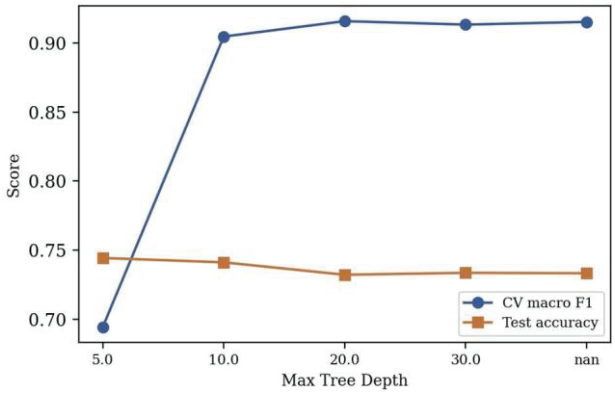


Fig. 7. CV macro F1 and test accuracy as a function of maximum tree depth.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

A curious note about Table VI is that the shallowest model (max_depth=5) attains the highest accuracy (0.7442), although it has by far the lowest F1 score (0.6943). The reason for this paradox is explained in Section V-C and lies in the relationship between class imbalance and classifier underfitting: shallow trees will predict majority classes (SAFE and HIGH RISK) more often, leading to inflated raw accuracies while degrading the performance on minority tiers. This result emphasizes the significance of using macro-averaged metrics instead of accuracy only when evaluating classifiers on imbalanced security datasets, as mentioned but not supported with evidence by several of the papers reviewed in Section II.

G. ROC-AUC Analysis

In addition to the threshold dependent measures of precision, recall, and F1 reported in Section VI-C, a one-vs-all ROC analysis was carried out for each risk level based on the probabilities provided by the Random Forest classifier ensemble (Section IV-C). Fig. 8 presents the ROC plots obtained from this analysis along with their AUC scores per risk level.

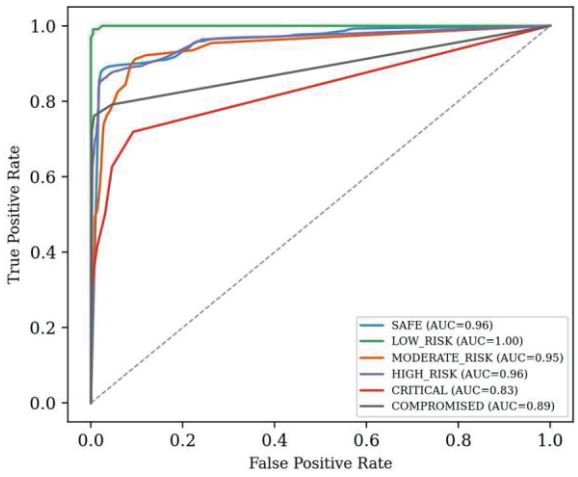


Fig. 8. One-vs-rest ROC curves and AUC per risk tier.

The ROC-AUC scores offer a vital qualification of the CRITICAL-tier issue described in Section VI-D. Despite the fact that CRITICAL-tier recall at the model's default decision threshold equals 0.00 (Table IV), it is possible that the class probabilities predicted by the ensemble may contain valuable discriminative information, meaning that the model assigns higher CRITICAL probabilities to positive instances than to negative ones, even though this effect may be not strong enough to overcome the default threshold of 0.5. When the CRITICAL-tier AUC is substantially greater than 0.5, despite zero recall, it means that the ensemble contains some non-threshold discriminative ability for the tier which can be leveraged via decision threshold recalibration or an anomaly score layer – a practical way forward for the suggested hybrid approach outlined in Section VIII.

H. Comparison with Prior Random Forest-Based IDS Work

Table VII provides a comparison between the metrics obtained by RF-NRA and the literature values for some of the relevant research papers discussed in the review. The direct comparison should be viewed with caution as earlier studies focus on binary or nominal attack type classification whereas RF-NRA uses a six-level ordinal scale. This comparison is done in order to provide context but in no way claims to be better than the earlier research.

TABLE VII. INDICATIVE COMPARISON WITH PRIOR RF-BASED IDS STUDIES

Study	Task	Reported Accuracy / F1
Zhang & Zulkernine [1]	Binary + anomaly detection	~94% (KDD'99)
Enhanced RF + KPCA [2]	Multi-class (attack type)	~96–98% (varies by class)
ADASYN-RF [6]	Multi-class, imbalance-corrected	~90–93% (CICIDS2017)
RF-NRA (this work)	6-tier ordinal risk state	73.3% acc. / 0.52 macro F1

The lower accuracy of RF-NRA compared to binary/nominal methods in terms of headlines is predictable and, according to the discussion presented in Section VII, mostly due to the higher requirement of generalization involved in assessing an ordinal criterion on a test subset intentionally including previously unseen attacks, a requirement which is not present in previous binary/nominal works in Table VII.

VII. DISCUSSION

A. The Cross-Validation / Test Gap

There is a significant difference between the cross-validation results (macro F1 = 0.92) and the results on the test set (macro F1 = 0.52). Analysis found that the reason for this difference is mainly related to the CRITICAL tier, where the recall was zero — only 3 out of 2,885 test cases have been classified correctly, with most of them (2,874 cases) being incorrectly classified as SAFE. This is due to the nature of the test set of NSL-KDD as discussed in Section V-C: the majority of the cases in the CRITICAL tier on the test set are of attack types (snmpguess, sendmail, named, httptunnel, and many more) which are not present at all in the training set. Thus, the model, which has never seen such behavior during the training process, will simply predict the majority class (SAFE), leading to a zero-shot generalization failure. As seen in Section VI-F, this difference is practically unaffected by the number of trees or tree depth.

B. Implications for Ordinal Risk Assessment

The very fact of this result itself provides a valuable empirical contribution to this research because it proves clearly that supervised and closed set classifiers like Random Forest cannot evaluate risks for a novel attack behavior even when they are highly accurate in detecting already known attacks. The limitation described above is a well-known fact that is often underestimated in literature on RF-IDSs [1], [9], [12], however, the approach of ranking attacks according to ordinal risk tiers makes it apparent much better than any accuracy rate because in the latter case not detected CRITICAL attack will be counted as just "not correct" among other cases of misclassification.

Simultaneously, the within-tier measure (74.60%) suggests that apart from this particular CRITICAL tier failure mode, the errors made by the model are predominantly graceful in nature

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

in terms of being made within adjacent tiers (e.g., MODERATE RISK cases misclassified as LOW RISK) rather than drastic jumps across multiple tiers. This makes a case for the inherent practicality of the ordinal risk tiers approach in theory since, with adequate exposure to representative samples during training, the errors made by the classifier are operationally insignificant. The practical relevance of this approach is largely dependent on adequate training data coverage of the attack space, which is elaborated upon in Section VIII.

C. Interpreting the Feature Importance Findings

The prevalence of traffic volume features (src_bytes, dst_bytes) and service diversity features (dst_host_srv_count, dst_host_diff_srv_rate, service) within the highest-ranked predictors is in keeping with the taxonomy of attacks employed in defining the risk tiers. DoS attacks (assigned HIGH RISK) tend to be associated with abnormal traffic volume or connection numbers, whereas Probe attacks (assigned LOW/MODERATE RISK) tend to be associated with access to an abnormally high number of different services/ports within a short span of time. This correspondence between the highest-ranking features and the semantics of the risk-tier assignments serves as a sanity check on the entire approach: The model is clearly learning traffic behavior that matches the desired risk types.

D. Operational Interpretation for Security Analysts

Given the findings reported in Section VI from the security operations perspective, it appears that RF-NRA should play a specific and confined role in lieu of being a replacement for existing IDS infrastructure. Since the model is very effective in classifying packets as SAFE, LOW RISK, and HIGH RISK with F1-scores of 0.77, 0.81, and 0.86, it could be used as a triage layer to effectively differentiate between normal traffic, low level reconnaissance, and denial-of-service attacks. At the same time, the model's inability to classify CRITICAL cases suggests that it cannot be used solely to detect remote-access attack attempts, especially those using attack methods which are underrepresented in the distribution in the model training. Therefore, a sound approach to deploying the model would include combining it with an additional layer designed to detect attack cases belonging to CRITICAL and COMPROMISED tiers, which is the idea suggested for further investigation in Section VIII.

The operational framing also reveals the proper way for downstream applications to consume the ordinal tier risk results. In other words, instead of relying on each window's assigned risk tier to give a final answer, a security dashboard that relies on the RF-NRA model's predictions will do better by viewing the series of ordinal risk tiers as the main input to its alerts – identifying an ESCALATING trend as an investigative lead, even if each individual window does not reach a high risk level on its own. The exact operational problem highlighted in Section I and Section II-B has now been found.

VIII. LIMITATIONS AND FUTURE WORK

- Imbalanced classes: extreme class imbalance (COMPROMISED: only 52 training samples) hinders learning on the rarest class, independently of the "unseen attack" problem in Section VII-A. Further work would need to investigate SMOTE/ADASYN based oversampling [6], [11] on COMPROMISED and CRITICAL classes and see if resampling can effectively increase recall for classes with few but meaningful training samples (and not because of an "unseen attack", where oversampling cannot solve the problem).
- Hybrid approach to anomaly detection: The CRITICAL tier-level anomaly described in section VII-A, coupled with the non-trivial ROC-AUC anomaly detected in section VI-G, indicates that the network may benefit from a hybrid architecture consisting of the use of a supervised random forest classifier to classify attack types based on labeled training sets in combination with unsupervised techniques such as isolation forest or one-class SVMs to detect anomalies in traffic that is not part of the training distribution.
- Generalization of the dataset: testing of the model using a second, more current dataset such as the CICIDS2017 or the UNSW-NB15 will serve to determine if the observed generalization gap is indeed an artifact of the train/test set split used for the NSL-KDD, or a feature of the ordinal risk tiers classification framework itself.
- Temporal validation of the early warning system: The temporal escalation detection mechanism in Algorithm 2 (Section IV-D) was implemented, but it was not tested using actual multiple stages of attacks since the connection data in the NSL-KDD dataset do not have the notion of temporal ordering for the multiple stages of attacks. This needs to be validated in future using a dataset that provides a timeline of multiple attack stages (e.g., a custom-made dataset from a red team exercise).
- Deployment time factors: the current work analyzes RF-NRA in the context of an offline classification scenario. Real-time application will need to include analysis of per-flow classification latency at production traffic rates, and may also include use of window-based weight updates for feature importance as proposed in IDRandom-Forest [12].
- Recalibration of threshold: The analysis in the ROC-AUC test in Section VI-G implies that a decision threshold calibration approach like a cost-based threshold lowering the prediction threshold for CRITICAL, considering that the cost of missing CRITICAL is much greater than the cost of raising a false alarm, is a relatively less demanding extension and worthy of empirical study before moving to the hybrid architecture extension above.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

IX. CONCLUSION

In this work, RF-NRA is developed, which is a novel framework that rephrases network security evaluation in terms of progressive risk state estimation instead of using the typical binary classification methodology. This is achieved by associating intrusion categories to six levels of increasing risks, which include SAFE, LOW RISK, MODERATE RISK, HIGH RISK, CRITICAL, and COMPROMISED. The benefit of RF-NRA is its ability to generate early warning behaviour which traditional IDS systems cannot perform, while utilizing the excellent classification capabilities of Random Forest.

A cross-validation macro F1 score of 0.92 and a test accuracy of 73.3% was obtained through evaluation on the NSL-KDD benchmark; in addition, this experiment highlighted a notable and educational generalization gap, wherein the inability of the classifier to identify CRITICAL attacks involving attack types unseen during training was shown. A sensitivity study on hyperparameters (Section VI-F), as well as a ROC-AUC analysis of the classifier's output (Section VI-G), demonstrated that the generalization gap is not a result of under- or overfitting and that there still existed some discriminative power beyond a simple threshold-based decision, respectively, which warrants further investigation into recalibrating thresholds and combining anomaly detection approaches in the future.

It is thus not the incremental gains in pure accuracy vis-a-vis other studies using Random Forest IDS that are the main contributions of this research; indeed, some of the aforementioned studies have achieved better overall accuracy by virtue of classifying into two classes or nominal classes (see Table VII). The key contributions are rather a different and operational way of framing the output space of the classifier, and an empirically well-grounded description of where such a framing holds and fails.

REFERENCES

[1] J. Zhang and M. Zulkernine, "Random-Forests-Based Network Intrusion Detection Systems," IEEE Trans. Syst., Man, Cybern. C, Appl. Rev., 2008.

[2] "Research on Intrusion Detection Based on an Enhanced Random Forest Algorithm," Applied Sciences, MDPI, 2024.

[3] "Research on Network Intrusion Detection in Random Forest Based on Pearson Feature Selection," in Proc. IEEE Conf., 2025.

[4] "Network Intrusion Detection Based on Random Forest and Support Vector Machine," in Proc. IEEE Conf.

[5] "Random Forest Modeling for Network Intrusion Detection System," ScienceDirect / ResearchGate.

[6] Z. Chen, W. Yu, and L. Zhou, "ADASYN-Random Forest Based Intrusion Detection Model," arXiv preprint, 2021.

[7] "A Multi-level Random Forest Model-Based Intrusion Detection Using Fuzzy Inference System for IoT Networks," Int. J. Comput. Intell. Syst., Springer, 2023.

[8] "Network Intrusion Detection System using Random Forest and Gradient Boosting Machines," in Proc. IEEE Conf., 2024.

[9] "Network Intrusion Detection using Decision Tree and Random Forest," in Proc. IEEE Conf., 2022.

[10] "A Simple and Robust Approach of Random Forest for Intrusion Detection System in Cyber Security," in Proc. IEEE Conf., 2022.

[11] "Wireless Sensor Networks Intrusion Detection Based on SMOTE and the Random Forest Algorithm," NCBI/PMC, 2019.

[12] "IDRandom-Forest: Advanced Random Forest for Real-Time Intrusion Detection," in Proc. IEEE Conf., 2024.

[13] "An Analysis of Random Forest Algorithm Based Network Intrusion Detection System," in Proc. IEEE Conf.

[14] "Cybersecurity and Risk Prediction Based on Machine Learning Algorithms," ResearchGate, 2024.

[15] "Machine Learning Models for Cybersecurity Risk Analysis in U.S. Critical Infrastructure Systems," ResearchGate, 2026.

APPENDIX A: COMPLETE NSL-KDD FEATURE LIST

Table VIII lists all 41 features used by RF-NRA, grouped by category as introduced in Section IV-A.

TABLE VIII. COMPLETE NSL-KDD FEATURE DESCRIPTIONS

#	Feature	Brief Description
1	duration	Length of the connection (seconds)
2	protocol_type	Protocol used (tcp, udp, icmp)
3	service	Destination network service (http, ftp, telnet, ...)
4	flag	Normal or error status of the connection
5	src_bytes	Bytes sent from source to destination
6	dst_bytes	Bytes sent from destination to source
7	land	1 if connection is from/to the same host/port, else 0
8	wrong_fragment	Number of wrong fragments
9	urgent	Number of urgent packets
10	hot	Number of "hot" indicators (e.g. entering a directory)
11	num_failed_logins	Number of failed login attempts
12	logged_in	1 if successfully logged in, else 0
13	num_compromised	Number of "compromised" conditions
14	root_shell	1 if root shell obtained, else 0
15	su_attempted	1 if "su root" command attempted
16	num_root	Number of root accesses
17	num_file_creations	Number of file creation operations
18	num_shells	Number of shell prompts invoked
19	num_access_files	Number of operations on access control files
20	num_outbound_cmds	Number of outbound commands (FTP sessions)
21	is_host_login	1 if login belongs to a host list
22	is_guest_login	1 if login is a guest login
23	count	Connections to same host in last 2 seconds
24	srv_count	Connections to same service in last 2 seconds
25	serror_rate	% of connections with SYN errors (same host)
26	srv_serror_rate	% of connections with SYN errors (same service)
27	rerror_rate	% of connections with REJ errors (same host)
28	srv_rerror_rate	% of connections with REJ errors (same service)
29	same_srv_rate	% of connections to the same service
30	diff_srv_rate	% of connections to different services
31	srv_diff_host_rate	% of connections to different hosts (same service)
32	dst_host_count	Count of connections to same destination host
33	dst_host_srv_count	Count of connections to same host, same service

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

34	dst_host_same_srv_rate	% of connections to same service (dest host)
35	dst_host_diff_srv_rate	% of connections to different services (dest host)
36	dst_host_same_src_port_rate	% of connections to same source port
37	dst_host_srv_diff_host_rate	% of connections to different hosts (same service, dest host)
38	dst_host_serror_rate	% of connections with SYN errors (dest host)
39	dst_host_srv_serror_rate	% of connections with SYN errors (dest host, same service)
40	dst_host_rerror_rate	% of connections with REJ errors (dest host)
41	dst_host_srv_rerror_rate	% of connections with REJ errors (dest host, same service)

APPENDIX B: COMPLETE ATTACK-TYPE TO RISK-TIER MAPPING

Table IX enumerates every attack type observed in the combined NSL-KDD train and test partitions, together with its assigned risk tier and a one-line description of the underlying attack technique, expanding on the summary given in Table II.

TABLE IX. COMPLETE ATTACK-TYPE REFERENCE

Attack Type	Tier	Description
normal	SAFE	Legitimate, non-malicious traffic
ipsweep	LOW RISK	Surveying hosts on a network (ping sweep)
nmap	LOW RISK	Network mapping / port scanning utility
satan	MODERATE RISK	Automated vulnerability scanning tool
portsweep	MODERATE RISK	Sequential port scan across a host
mscan	MODERATE RISK	Multi-service vulnerability scanner
saint	MODERATE RISK	Security auditing / vulnerability scan tool
back	HIGH RISK	DoS via malformed URL causing server slowdown
land	HIGH RISK	DoS via spoofed packet with same src/dst
neptune	HIGH RISK	SYN flood denial-of-service
pod	HIGH RISK	"Ping of death" oversized ICMP packet DoS
smurf	HIGH RISK	ICMP broadcast amplification DoS
teardrop	HIGH RISK	DoS via overlapping IP fragment offsets
apache2	HIGH RISK	DoS targeting Apache HTTP server
udpstorm	HIGH RISK	UDP flood denial-of-service
processtable	HIGH RISK	DoS by exhausting the process table
mailbomb	HIGH RISK	DoS via mass unsolicited email
worm	HIGH RISK	Self-propagating malicious program
ftp_write	CRITICAL	Unauthorized write access via FTP
guess_passwd	CRITICAL	Password guessing / brute-force login
imap	CRITICAL	Exploit of IMAP mail service vulnerability
multihop	CRITICAL	Multi-step unauthorized access across hosts
phf	CRITICAL	Exploit of vulnerable CGI phf script
spy	CRITICAL	Covert unauthorized surveillance access
warezclient	CRITICAL	Downloading illegally shared ("warez") software
warezmaster	CRITICAL	Hosting/distributing illegally shared software
xlock	CRITICAL	Exploiting X11 screen-lock vulnerability
xsnoop	CRITICAL	Monitoring X11 session keystrokes
snmpguess	CRITICAL	Guessing SNMP community strings
snmpgetattack	CRITICAL	Exploiting SNMP GET requests
httptunnel	CRITICAL	Tunneling unauthorized traffic over HTTP
sendmail	CRITICAL	Exploit of vulnerable sendmail service
named	CRITICAL	Exploit of vulnerable DNS (named) service
buffer_overflow	COMPROMISED	Memory overflow exploit for privilege escalation
loadmodule	COMPROMISED	Unauthorized kernel module loading
perl	COMPROMISED	Privilege escalation via Perl script exploit
rootkit	COMPROMISED	Installation of concealed root-level access tool
ps	COMPROMISED	Process-status exploit for privilege escalation
sqlattack	COMPROMISED	SQL-based exploit for administrative access
xterm	COMPROMISED	Privilege escalation via X terminal exploit

(This work is licensed under a Creative Commons Attribution 4.0 International License.)