

Towards Trustworthy Intrusion Detection: A Calibration-Aware and Explainable Classical-ML Pipeline on NSL-KDD

Zainab Qahtan Mohammed 

Department of Computer Science, College of Basic Education, Diyala University, Diyala, 32001, Iraq.

zainabkahtan@uodiyala.edu.iq

Abstract

Existing assessments of machine-learning-based intrusion detection systems (IDSs) often consider each of accuracy, reliability of probability outputs, and transparency of decisions separately. Few attempt to combine all three dimensions in a single assessment. This paper provides a reproducible IDS pipeline that satisfies all three dimensions simultaneously using four classical algorithms: Logistic Regression, Linear Support Vector Machine, Gaussian Naive Bayes, and Random Forest, appropriate for tabular network-flow data. The pipeline encompasses feature conditioning, model training, probability recalibration, and interpretability. Categorical data are one-hot encoded, and continuous data are Z-score normalized, with the same transformations applied to the NSL-KDD data. For model training, we use the KDDTrain+ set, and we evaluate the final model on the KDDTest+ set using various classification metrics and calibration. Among all proposed models, Random Forest had the best overall metrics, achieving 0.989 accuracy and 0.990 F1 for the attack class, and the best calibration with a Brier score of 0.009. The research shows that classical models, when properly calibrated and made interpretable, can serve as competitive benchmarks for IDS systems. Future research should test this pipeline on other datasets and consider the effect of interpretability on real-world calibrations, such as latency.

Keywords: NSL-KDD benchmark, calibration-aware classifiers, classical machine learning, explainable intrusion detection, network security and Random Forest

Article history: Received: 19 Feb 2026, Accepted: 9 Jun 2026, Published: 15 Sep 2026.

This article is open-access under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In modern networked environments, operational and network-level threats persist. These threats include, but are not limited to: network scanning, credential attacks, malware, denial-of-service attacks, and unauthorized access to systems [1]. An effective intrusion detection system (IDS) should achieve two objectives: identify and classify malicious network traffic and build confidence in the traffic's malicious nature to support informed, efficient responses. An accurate but poorly calibrated system generates confidence estimates that diverge from the truth and either floods analysts with false positives or remains unreported, causing genuine high-risk events to be ignored [2].

Live training enterprise network traffic is almost always unavailable for research. Because of this, the research community constructs simulated environments and uses standard public datasets as subjects for controlled network-based experiments. NSL-KDD was the first standardized training dataset intended to replace the original KDD Cup 99 dataset [3], [4]. The main change in NSL-KDD was removing duplicate KDD records. Because NSL-KDD remains relevant to data and network traffic research, it has a structured, controlled composition. Therefore, it allows calibration and validation of an IDS system and provides a basis for IDS evaluation and system design.

Analyses of the DARPA and KDD evaluation

frameworks have shown that the choice of dataset and evaluation technique influences findings in intrusion detection system (IDS) research [5]. Reproducibility and rigor require researchers to specify their sample-splitting approach, modeling assumptions, and metric definitions. KDD'99 shows significant performance gains when duplicate records are removed, and NSL-KDD, by construction, increases evaluation difficulty.

Benchmark-focused IDS research has gaps in KDD-styled IDS derivatives. These gaps focus on measurement integration skills metrics like accuracy and precision and largely ignore whether the model was confidently error-free or interpretable. To address these issues, this paper embeds efficient classical modeling into the IDS evaluation pipeline, adjusts probabilities, and provides feature-based explanations. The specific contributions of this research are as follows:

1. A machine learning pipeline for binary IDS classification, designed to be reproducible using the NSL-KDD dataset as a benchmark.
2. A comprehensive assessment of four classical learners, Logistic Regression, a probability-adjusted Random Forest, Linear SVM, and Gaussian Naive Bayes, evaluated under the canonical KDDTrain+/KDDTest+ protocol.
3. We also incorporate an operationally oriented evaluation design that contrasts the value of calibration-aware metrics with conventional classification-only scores.
4. The use of the Random Forest model's variable (feature) selection as an Interpretability instrument to promote the analyst's understanding of the model and the classification results.

Subsequent sections will be structured as follows: Section 2 examines existing studies in IDS, calibration, and interpretability, and Section 3 outlines the benchmark, the models, the architecture/Design, preprocessing, the flow of information/tree models, and the evaluation. Section 4 details the experimental Design, followed by Section 5, which describes and benchmarks the results. Section 6 delves into the implications and constraints of these methodologies, followed by the conclusion in Section 7.

2. Related Work

Traditional machine learning techniques applied to large data traffic volumes have evolved dramatically. Building on these early developments, the literature has shifted toward applying sophisticated deep learning and ensemble techniques. In this literature, NSL-KDD remains a central reference benchmark. This is primarily because NSL-KDD's Design seeks to eliminate the data redundancy issues in KDD'99 and create a more equitable validation-testing split. Classical algorithms, including Random Forest, Logistic Regression, Naive Bayes, support vector machines, and decision trees, remain very popular in NSL-KDD studies. This is largely due to their ease of implementation and reduced time and resource costs for the tabular, flow-level data format. The literature overwhelmingly shows that preprocessing quality, along with the evaluation pipeline, increases the competitiveness of these classical methods and machines [6].

In many contributions that cite NSL-KDD, authors have alluded to the dataset's few shortcomings. The NSL-KDD dataset presents traffic profiles that have become outdated and does not include profiles of encrypted traffic, cloud-based traffic, or IoT-based traffic. To address these issues, the authors of NSL-KDD proposed the UNSW-NB15 data set, and Sharafaldin et al. created the CIC-IDS2017 data set with the same motivation [7–9]. In many comparative benchmark studies, authors show that the dataset used to benchmark IDS performance is a major contributor to IDS performance and its generalization capacity. Therefore, results on NSL-KDD should be used to demonstrate controlled research rather than the practicality of real-world applications.

A related area of research focuses on evaluating probability outputs in classification systems. For security professionals, predicted probabilities signify the likelihood that something will actually occur. Research by Niculescu-Mizil and Caruana [2] found that most classifiers, particularly those using margin optimization, produce grossly uncalibrated outputs. Leathart et al. [10] specifically researched the calibration of tree-based models. The scikit-learn library applies sigmoid and isotonic post-hoc

calibration to its CalibratedClassifierCV [11]. Within the context of IDS, poor calibration, beyond a loss of statistical accuracy, can disrupt alert prioritization and the effective allocation of analyst resources, in addition to eroding analyst confidence in high-value findings.

Model interpretability is the third dimension of related research. Lundberg and Lee [12] proposed Structured Shapley Values, a theoretically justified technique for explaining model predictions consistently at the individual-instance level. In IDS frameworks, interpretability offers substantive benefits because it lets analysts determine whether an alert reflects related traffic rather than noise and confirm whether the model aligns with domain concerns. Despite these opportunities, most IDS research focuses on classification performance, with little to no attention to calibration quality or the interpretability of model outputs.

This work sits at the intersection of three research streams: classification, calibration, and interpretability. The aim is not to produce a new dataset or a neural architecture. Methodologically, the contribution is a coherent, reproducible evaluation workflow for the NSL KDD protocol that assesses all three dimensions of quality together. The main purpose of this work stems from the fact that a practical IDS model must be correct, auditable, and trustworthy with respect to the degree of confidence it expresses.

3. Materials and Methods

3.1 Architecture of the Proposed Model

The architecture described in this research is an operationally focused IDS pipeline with two main objectives: first, statistical reliability in estimating

the probability for each classification decision; second, creating human-understandable evidence for each alert associated with the NSL-KDD connection record. The architecture consists of the following five consecutive steps: (1) the ingestion of the raw data, (2) the preparation of the features, (3) the training of the classifiers, (4) the calculation of the confidence score, which is modified, and (5) the explanation of the decision. In the first step, the proposed architecture processes each NSL-KDD record, which combines quantitative and categorical flow data. In the second step, the model resolves feature-space heterogeneity by applying one-hot encoding to symbolic attributes and Z-score normalization to continuous attributes, ensuring that no feature type introduces artificial bias to the linear and ordinal scales or disrupts the categorical indicators.

In this stage, the feature matrix will be exposed to four classical classifiers used in parallel. These are: the baseline model with interpretable output, Logistic Regression; the margin-based probabilistic Linear SVM; the non-linear Random Forest used for structured tabular data; and the compact probabilistic Gaussian Naive Bayes model providing a lower bound. The fourth stage will address SVMs and other classifiers' tendency to issue overconfident and poorly scaled probability scores. This will be tackled with a cross-validated sigmoid fitting. In the fifth and final stage, we use ranked feature contributions with a variable decision threshold, then integrate them with the new calibrated scores. This means every traffic prediction will be interpretable, as the most influential factors will justify the prediction, along with a risk-level evaluation.

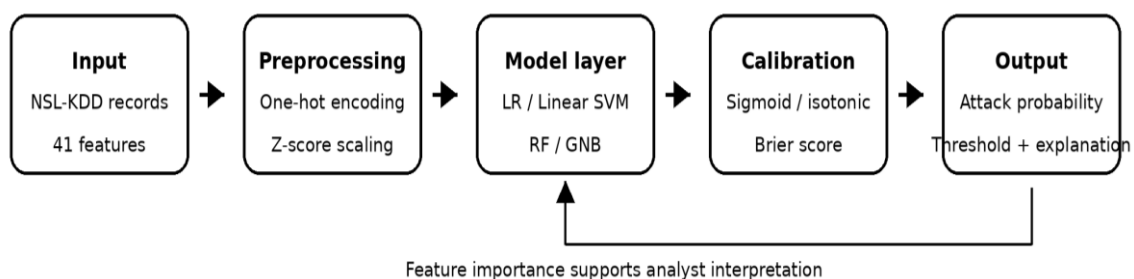


Fig. 1 A refined schematic of the proposed IDS framework, integrated with calibration-aware and interpretable components.

Table 1: Dataset summary (KDDTest+).

Task Type	Evaluation Set	Total Records	Benign Instances	Attack Instances	No. of Features
Binary Classif.	KDDTest+ (held-out)	22,544	9,711 (43.1%)	12,833 (56.9%)	41

3.2 Dataset

This study uses experiments based on the NSL-KDD benchmark for its standard partitioning method [13]. We trained classifiers on the KDDTrain+ section and evaluated them on the KDDTest+ section, which was never accessed. This completely prevents any data leakage from the training set to the evaluation set. In the NSL-KDD benchmark, each sample has 41 different features. These include network flow statistics (e.g., total bytes exchanged, total time taken, etc.) and 3 categorical labels: protocol, service type and the connection status flag.

The classification task is framed as binary: each record is assigned one of two labels, Normal or Attack. To guard against unstable model selection, we guide hyperparameter decisions with stratified k-fold cross-validation within the training set, while reserving the KDDTest+ partition exclusively for the final performance report. Although NSL-KDD has well-documented constraints as an ageing benchmark, its fixed structure and community adoption make it a reliable instrument for producing reproducible and historically comparable IDS evaluations.

3.3 Preprocessing

The preprocessing stage is uniformly applied to both training and testing partitions. Below is a list of the basic steps taken to preprocess the data. Considering the context, the steps are:

1. One-hot encoding for all three protocol type services, and flag symbolic values.
2. Imposition of Z-score normalization (standardization) for all numerical values.
3. Taking a subset of NSL-KDD, split into KDDTrain+ for training and KDDTest+ for final testing.
4. Class imbalance is weighted as stated.
5. Crucially, every fitting transformation (encoders and scalers) is estimated on the training partition only and then applied as-is to the test partition, preventing information leakage from the held-out data into the training process.

3.4 Machine Learning Models

We chose four models that are fast and easy to apply, given their ubiquity in tabular IDS tasks and their transparency for benchmark evaluations. The models are:

- Logistic Regression (LR): an efficient linear learner with L2 regularisation ($n = 1$) and directly interpretable coefficients.
- Linear Support Vector Machine (Linear SVM): a margin-based classifier that can provide probabilistic estimates.
- Random Forest (RF): An ensemble classifier composed of decision trees that can openly provide a feature importance score.
- Gaussian Naive Bayes (GNB): A simple classifier used as a base for pyramids that is used to provide the lower boundary behavior for the processed pipeline.

3.5 Probability Calibration

Probability calibration allows for the true frequencies of outcomes to be compared against the confidence scores produced by predictive models. An example of probability score misalignment is using an SVM in an Intrusion Detection System (IDS), where a given score reflects a high degree of certainty when the information in the score suggests otherwise. This situation would most certainly hinder discernment in an IDS.

For support vector machines (SVMs), post hoc calibration is applied using the CalibratedClassifierCV implementation in scikit-learn, with a sigmoid mapping estimated by cross-validation. This common approach is outlined in the scikit-learn calibration module [11]. Calibration is evaluated using the Brier score; the lower the Brier score, the more aligned the predicted probabilities are with the event rates. In addition to the Brier score, reliability diagrams plot mean predicted probabilities against the actual observed positives in each confidence bin to illustrate the reliability of the predicted scores. Fig. 2 shows the reliability diagram of Logistic Regression, calibrated Linear SVM, and Random Forest on the NSL-KDD test set in terms of calibration performance.

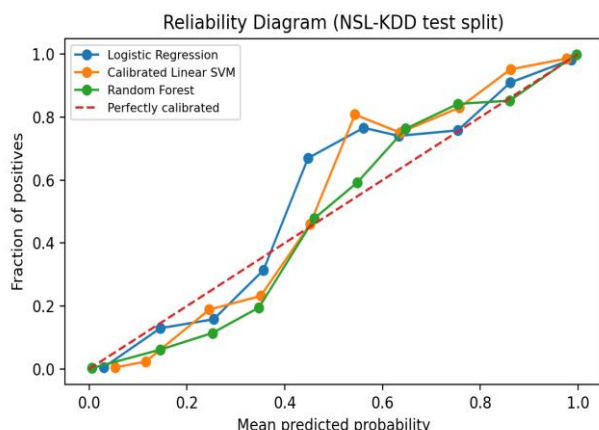


Fig. 2 Proposed IDS architecture and reliability diagrams for LR, calibrated SVM, and RF on the NSL-KDD test set.

3.6 Evaluation Metrics

A composite of classification-quality, ranking-quality, and probability-calibration measures captures multiple complementary facets of IDS behaviour. The following indicators are reported throughout the study:

- Accuracy measures the total number of records classified correctly.
- Attack-class metrics, precision, recall, and the F1-score on the attack class constitute the standard reporting metrics in IDS evaluation studies.
- Macro-F1 represents class balance, reducing single-class domination.
- ROC-AUC indicates the quality of ranks of the estimated values.
- Probability estimates affect calibration quality, which is measured by the Brier score.

4. Experimental Setup

All experiments follow the NSL-KDD protocol. We fit every model on KDDTrain+ and evaluate it externally on KDDTest+ without any feedback from the test split, ensuring strict separation between training and testing. The experiments are kept simple to avoid overfitting. Logistic Regression uses L2 regularization and mitigates class imbalance. Class imbalance is addressed using a Linear Support Vector Machine with cross-validated sigmoid calibration. We set a moderate number of trees and a limited depth for the Random Forest to improve generalization. All models use a fixed random seed (random_state = 42) to keep the experiments in check.

All models are built in Python 3 and depend on scikit-learn. Categorical attributes such as protocol_type, service, and flag are one-hot encoded, creating 122 indicator dimensions. The numerical attributes are transformed to a standard normal distribution using the StandardScaler. The class imbalance is dealt with by Logistic Regression using the class_weight='balanced'. The Linear SVM is wrapped in scikit-learn's CalibratedClassifierCV with five cross-validation folds. The Random Forest is configured with 100 trees and a maximum depth of 20 levels. To prevent leakage from training into testing, we fit every preprocessing component only on the training partition, then apply the resulting pipeline unchanged to the test partition.

5. Results

5.1 Presentation of Classification and Calibration Results

Table 2 consolidates the classification and calibration outcomes for all four models evaluated on the KDDTest+ partition. Random Forest outperforms all models with a predictive accuracy of **0.989**. The model also achieves an attack-class precision of 0.989, with 0.991 recall and a **0.990** F1 score. The model's ROC-AUC of 0.999, the highest in this study, suggests it almost perfectly distinguishes attack from normal instances, confirming that the model score for threshold-based triage is reliable.

Calibrated Linear SVM and Logistic Regression achieved overall accuracies of **0.959** and **0.951**, respectively, making them the second- and third-best models. Of the three models, the calibrated SVM has the highest F1 score and recall. We can attribute this to the margin-based SVM's ability to better manipulate the geometry of the preprocessed feature space of the KDD dataset. Of all the models, the Gaussian Naive Bayes achieves the lowest accuracy of **0.779**. This is expected, as it assumes feature independence, which the KDD dataset violates due to correlations among traffic features. Although this model has an attack-class precision of **0.992**, it has a substantially lower recall of 0.616, meaning it captures only a small portion of actual attack instances and is unsuitable as a primary intrusion detection system.

Table 2: Performance comparison on *the KDDTest+* partition.

Metric	Logistic Regression	Calibrated SVM	Random Forest	Gaussian NB
Accuracy (%)	95.1	95.9	98.9	77.9
Error Rate (%)	4.9	4.1	1.1	22.1
AUC-ROC	0.986	0.986	0.999	0.960
Attack F1 (%)	95.7	96.4	99.0	76.0
Attack Recall (%)	96.5	97.7	99.1	61.6
Attack Precision (%)	94.9	95.1	98.9	99.2
Macro-F1 (%)	95.0	95.8	98.8	77.7
Brier Score	0.039	0.039	0.009	0.221

On the calibration dimension, Random Forest again leads with a Brier score of 0.009, confirming that its probability outputs closely track the empirical attack rate in the test data. Logistic Regression and the calibrated SVM share a Brier score of 0.039, reflecting acceptable but less precise probability alignment. Gaussian Naive Bayes records a substantially higher Brier score of 0.221, marking it as the least reliable model for confidence-driven decision-making. Overall, these calibration results reinforce the central thesis of this study: evaluating an IDS solely on classification accuracy is insufficient; the trustworthiness of the model's probability estimates is an equally important criterion for operational deployment.

5.2 Confusion Matrix of the Best Model

Table 3 illustrates the confusion matrix for the Random Forest model, the best-performing model in this research. This confusion matrix is evaluated on the complete KDDTest+ partition. Of the 22,544 records, Random Forest correctly classified 12,723 as attacks and 9,573 as normal. It generated 138 false positives and 110 false negatives. In an operational IDS, false positives increase analyst workload. In contrast, false negatives allow intrusions to go undetected, potentially with a large impact on system defense. The low counts of false positives and false negatives show that Random Forest provides a solid, tactically significant detection baseline within NSL-KDD.

Table 3: Confusion matrix for Random Forest on the full KDDTest+ partition (binary classification).

Benign	TN = 9,573	FP = 138	98.6%
Malicious	FN = 110	TP = 12,723	99.1%

5.3 Deployment Cost and Model Comparison

Table 4 summarizes each model's practical deployment characteristics. When resources are limited, we recommend Logistic Regression because of its small storage footprint and faster sample-based inference. The calibrated Linear SVM is also storage efficient, but the post-hoc calibration wrapper increases inference latency. Random Forest offers the best calibration and detection performance, but results in a larger serialized model and slower inference. In practice, model selection requires a trade-off among detection performance, reliability, and inference speed.

Table 4: Deployment cost summary measured under a standard CPU-based experimental environment.

Metric	Logistic Regression	Calibrated Linear SVM	Random Forest
Disk Footprint (MB)	0.006	0.009	18.511
Latency (ms/record)	0.018	0.120	0.244

5.4 Interpretability Results

We performed a global feature-importance analysis on the Random Forest model to evaluate how the model makes classification decisions. The ten highest-ranked features, shown in Fig. 3, were all preprocessing steps that produced network-flow descriptors. These descriptors all contain discriminative signals that correlate to the binary classification task. To support operations, an IDS alert must include context. Without context, analysts cannot assess an alert's plausibility or the IDS's root

cause. By surfacing the features that most influence each prediction, the model allows analysts to confirm that actual network anomalies are being

acted upon, not just random relationships in the training data.

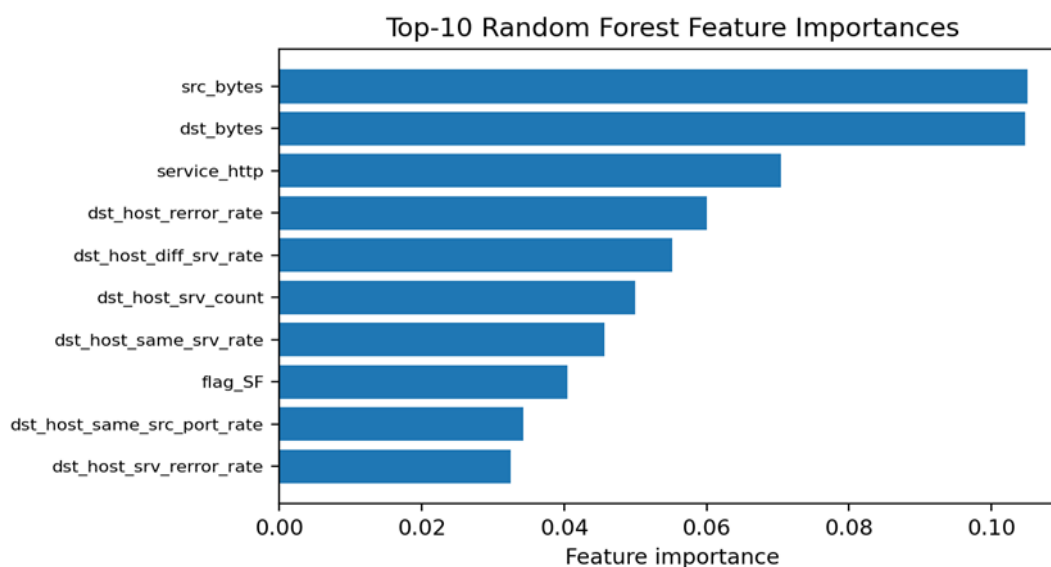


Fig. 3 Top-10 feature importance scores obtained from the Random Forest classifier after feature preprocessing.

6. Discussion

The results show that Random Forest performs best. This includes all tested metrics: accuracy, F1 score, ROC-AUC, and calibration quality. The superiority of tree-based ensembles on NSL-KDD is consistent with prior literature and likely reflects the dataset's tabular, flow-level structure, which suits the recursive partitioning logic of decision forests. However, this finding should not be generalized as evidence that Random Forest will dominate in other settings; NSL-KDD is a tightly controlled benchmark, and the IDS literature documents numerous studies with strong results on this corpus. The principal contribution of this work, therefore, lies not in the raw performance figures but in the integrated, reproducible framework that combines detection accuracy, calibrated confidence, and feature-level interpretability within a single evaluation pipeline constructed according to CRISP-ML methodology principles, which we elaborate in subsequent sections.

The model comparison reveals several practically important trade-offs. When storage size and inference speed are prioritized, the calibrated Linear SVM presents a viable alternative to Random Forest, offering reasonable detection performance at

a fraction of the memory cost. Logistic Regression provides an even leaner option with the fastest inference, making it attractive for latency-critical deployments. The Gaussian Naive Bayes results illustrate a broader lesson: high precision alone, achieved here partly by predicting the majority class more conservatively, does not constitute an effective intrusion detector when recall is low and calibration is poor. These observations collectively indicate that IDS system designers should weigh not only classification scores but also the operational cost of false alarms, missed detections, inference overhead, and the reliability of the confidence values used to drive automated or analyst-assisted response workflows.

Well-calibrated probability outputs have direct practical value in deployment contexts. When a model's confidence scores accurately reflect event likelihoods, security operations teams can implement tiered response strategies: log low-confidence alerts for periodic review, escalate moderate-confidence alerts to an analyst, and route high-confidence detections to automated mitigation. In contrast, an uncalibrated model may assign uniformly high confidence to its predictions regardless of true risk, misleading analysts into either overreacting or

under-prioritizing genuine threats. The Brier scores and reliability diagrams presented in this study provide empirical validation that the calibrated models, particularly Random Forest, produce probability estimates that can be trusted to support such structured response workflows.

The principal limitation of the present study is its exclusive reliance on NSL-KDD as the evaluation corpus [14]. While the dataset is appropriate for controlled, reproducible experimentation, it does not capture the diversity of contemporary network environments and lacks representative traffic from encrypted connections, cloud-hosted services, IoT devices, and recently documented attack families. Validation on more current benchmarks—such as UNSW-NB15 and CIC-IDS2017 — will be necessary before the conclusions of this study can be extended to real-world deployment contexts. A secondary limitation is that the interpretability analysis is limited to global feature-importance rankings derived from the Random Forest model; instance-level explanation methods such as SHAP, which would allow analysts to interrogate individual predictions, were outside the scope of the current work.

7. Conclusion and Future Work

We describe a structured framework for evaluating intrusion detection systems (IDS), with a unified approach to classification, probability calibration, and model interpretability, developed as a repeatable process on the NSL-KDD benchmark. This process integrates feature engineering, training with classical machine learning learners, post-hoc probability adjustment, and explanation through feature importance into a single pipeline. Among the models trained in this framework, Random Forest performed best, with an accuracy of 0.989, an attack-class F1 score of 0.990, an ROC-AUC of 0.999, and the lowest Brier score of 0.009. These results provide a compelling argument that classical machine learning models, when employed rigorously, can be well-calibrated and interpretable in the context of IDS.

Because these results are based on the NSL-KDD benchmark, which is a controlled setting, generalizing them to modern production systems should be done with care. Future work will extend

the proposed framework to other contemporary datasets, such as UNSW-NB15 and CIC-IDS 2017, and run it in near-real environments. Future work will also incorporate risk-informed threshold optimization for alerts and post-hoc explanatory models, such as SHAP, within the alerts. These alerts will provide human-readable justifications to security analysts.

Acknowledgement

The Department of Computer Science, College of Basic Education, Diyala University supported this research. The author thanks the contributors to the NSL-KDD benchmark dataset and the scikit-learn community for their open-source contributions.

Conflict of Interest

The author has no conflict of interest to declare. This research has no financial partnerships that may have influenced the research conducted.

References

- [1] Hindy, H., Brosset, D., Bayne, E., Seeam, A., Tachtatzis, C., Atkinson, R., & Bellekens, X. (2020). A taxonomy and survey of intrusion detection system design techniques, network threats and datasets. arXiv. <https://doi.org/10.48550/arXiv.1806.03517>
- [2] Niculescu-Mizil, A., & Caruana, R. (2005). Predicting good probabilities with supervised learning. In Proceedings of the 22nd International Conference on Machine Learning (ICML) (pp. 625–632). ACM. <https://doi.org/10.1145/1102351.1102430>
- [3] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA) (pp. 1–6). IEEE. <https://doi.org/10.1109/CISDA.2009.5356528>
- [4] Canadian Institute for Cybersecurity (CIC), University of New Brunswick. (2009). ISCX NSL-KDD dataset. Retrieved January 2026, from <https://www.unb.ca/cic/datasets/nsf.html>
- [5] McHugh, J. (2000). Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. ACM Transactions on Information and System Security, 3(4), 262–294. <https://doi.org/10.1145/382912.382923>
- [6] Abbas, Q., Hina, S., Sajjad, H., Zaidi, K. S., &

- Akbar, R. (2023). Optimization of predictive performance of intrusion detection system using a hybrid ensemble model for secure systems. *PeerJ Computer Science*, 9, e1552. <https://doi.org/10.7717/peerj-cs.1552>
- [7] Sapre, S., Ahmadi, P., & Islam, K. (2019). A robust comparison of the KDDCup99 and NSL-KDD IoT network intrusion detection datasets through various machine learning algorithms. *arXiv*. <https://doi.org/10.48550/arXiv.1912.13204>
- [8] Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems. In *Proceedings of the Military Communications and Information Systems Conference (MilCIS)* (pp. 1–6). IEEE. <https://doi.org/10.1109/MilCIS.2015.7348942>
- [9] 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). <https://doi.org/10.5220/0006639801080116>
- [10] Leathart, T., Frank, E., Holmes, G., & Pfahringer, B. (2017). Probability calibration trees. In *Proceedings of the 9th Asian Conference on Machine Learning (ACML), Proceedings of Machine Learning Research*, 77, 145–160. <https://proceedings.mlr.press/v77/leathart17a.html>
- [11] Scikit-learn developers. (2026). Probability calibration – CalibratedClassifierCV. Scikit-learn Documentation. <https://scikit-learn.org/stable/modules/calibration.html>
- [12] Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 4765–4774. <https://doi.org/10.48550/arXiv.1705.07874>
- [13] NSL-KDD Dataset [Dataset]. Zenodo, 2025. Available online: <https://doi.org/10.5281/zenodo.17424143>
- [14] Sarhan, M., Layeghy, S., & Portmann, M. (2022). Towards a standard feature set for network intrusion detection system datasets. *Mobile Networks and Applications*, 27(1), 357–370. <https://doi.org/10.1007/s11036-021-01843-0>