

A Lightweight and Explainable Machine Learning Framework Network Intrusion Detection and Monitoring

Vishal Dange

Department of computer science, Sambhajirao Kendre Mahavidyalaya, Jalkot.
Vishaldange2099@gmail.com

Abstract—Network Intrusion Detection Systems (IDS) built on machine learning routinely report high accuracy on benchmark datasets, yet practical adoption is limited by excessive false-positive rates, opaque black-box predictions, and evaluation protocols that rarely test generalization beyond a single dataset. This paper proposes a lightweight and explainable web-based intrusion detection framework that combines filter- and embedded-based feature selection to shrink the input feature space, the Synthetic Minority Over-sampling Technique (SMOTE) to improve detection of rare attack classes, a comparative evaluation of lightweight classifiers — Decision Tree, Random Forest, and XGBoost — chosen for their favorable accuracy-to-inference-time trade-off, shapley Additive explanations (SHAP) to expose the features driving each classification decision, and a Flask-based dashboard for real-time visualization of traffic, attack distribution, and per-alert risk scores. The design further incorporates a cross-dataset evaluation protocol, training on one benchmark dataset and testing on another, to assess generalization rather than relying on same-dataset performance alone, since recent literature shows this can substantially overstate real-world effectiveness. This paper documents the system architecture, methodology, experimental design, and evaluation protocol; quantitative results will be reported following execution of the described experiments on CICIDS2017 and UNSW-NB15. The overall contribution is a practically deployable IDS design that treats false-positive reduction, computational lightness, minority-class sensitivity, interpretability, and cross-dataset generalization as jointly evaluated objectives.

Keywords—Intrusion Detection; Explainable AI (SHAP); Lightweight Machine Learning; Cross-Dataset Generalization; SMOTE; Real-Time Monitoring.

I. INTRODUCTION

Modern networks are subject to a continuously growing volume and diversity of cyber-attacks, including denial-of-service, port scanning, brute-force, botnet, and web-based intrusions. Intrusion Detection Systems (IDS) are a foundational defence, tasked with distinguishing malicious network traffic from benign activity so that administrators can respond before damage occurs. As attack volume and network scale have grown, manual, signature-based inspection has become insufficient, motivating a large body of research into machine-learning-based IDS that can automatically classify traffic in near real time.

Despite strong reported accuracy, existing ML-based IDS approaches face several recurring, practically limiting problems. First, class imbalance is endemic to IDS datasets — benign traffic vastly outnumbers rare attack types such as infiltration or web attacks — causing models to overfit to the majority class and miss minority attacks precisely when detecting them matters most. Second, many approaches use the full, high-dimensional feature set exposed by flow-extraction tools such as CICFlowMeter, which increases computational cost, slows inference, and can itself inflate the false-positive rate by including redundant or noisy features. Third, the classifiers most often used to obtain top reported accuracy — deep neural networks and large ensembles — are computationally heavier than is often necessary or practical for resource-constrained deployments, and their internal decision logic is largely opaque to a human analyst. Fourth, this opacity is a genuine operational liability: when an IDS flags traffic as malicious, an analyst investigating the alert has no structured way to see which features drove that decision, which slows triage and reduces trust in automated recommendations. Fifth, IDS research is predominantly evaluated within a single dataset using a train/test split drawn from the same distribution; recent cross-dataset studies show that models achieving near-perfect accuracy within one dataset can degrade to close to chance-level

performance when evaluated on a different network environment, indicating that same-dataset accuracy alone is a poor proxy for real-world generalisation.

Taken together, the literature reviewed in Section II indicates that existing ML-based IDS solutions tend to optimise detection accuracy while giving comparatively little combined attention to computational efficiency, minority-class detection, decision interpretability, and cross-environment generalisation — with most works addressing at most one or two of these concerns in isolation. This leaves a research gap for an IDS design that treats all of these properties as joint, explicitly evaluated objectives within a single, deployable system.

Research question: Can a lightweight machine-learning IDS combining feature selection, SMOTE, and SHAP provide effective and interpretable network intrusion detection while supporting real-time, web-based monitoring and generalising across evaluation datasets?

This research question is addressed through the following specific objectives:

1. Reduce the false-positive rate of the baseline classifier by applying Mutual Information and Random Forest-based feature selection prior to model training.
2. Identify a lightweight classifier with the best accuracy-to-inference-time trade-off by comparatively evaluating Decision Tree, Random Forest, and XGBoost on identical data splits.
3. Improve minority-class detection by applying SMOTE and comparing Precision, Recall, and F1-score for rare attack classes before and after balancing.
4. Implement SHAP-based explainability so that every classification the system produces is accompanied by a ranked list of contributing features.
5. Design and justify a per-flow risk-scoring function that maps classifier confidence and attack severity onto an interpretable 0–100 scale.
6. Build a Flask-based web dashboard that displays live traffic volume, attack-type distribution, per-alert confidence, and risk score.
7. Evaluate cross-dataset generalisation by training the selected model on one benchmark

dataset (e.g., CICIDS2017) and testing on a second, structurally different dataset (e.g., UNSW-NB15).

8. Report inference time and trained-model size for each classifier, in addition to standard detection metrics, to substantiate the “lightweight” claim quantitatively.

The main contributions of this work are: (i) a filter/embedded feature-selection mechanism (Mutual Information and Random Forest importance) to reduce unnecessary network features and the false-positive rate; (ii) a comparative evaluation of lightweight classifiers — Decision Tree, Random Forest, and XGBoost — selected for favourable accuracy-versus-inference-time trade-offs; (iii) SMOTE-based class balancing to improve recall and F1-score on minority attack classes; (iv) a SHAP-based explainability layer reporting the features contributing to each classification; (v) a Flask-based web dashboard for real-time visualisation of traffic, attack distribution, and per-alert risk score; and (vi) a cross-dataset evaluation protocol training on one benchmark IDS dataset and testing on another, to explicitly measure generalisation.

II. LITERATURE REVIEW

The CICIDS2017 dataset, developed by the Canadian Institute for Cybersecurity, remains one of the most widely used benchmarks for IDS research because it captures realistic, labelled bidirectional network flows spanning benign traffic and fourteen attack categories drawn from seven attack families, extracted using CICFlowMeter [1]. The companion UNSW-NB15 dataset similarly provides labelled, feature-extracted flows across a broad set of modern attack categories and is frequently used alongside CICIDS2017 for comparative and cross-dataset evaluation [2].

On the model side, Random Forest [3] and gradient-boosted tree ensembles such as XGBoost [4] are consistently reported as offering a favourable balance between detection accuracy and computational cost relative to deep-learning alternatives, which motivates their selection as the lightweight classifiers evaluated in this work.

Class imbalance is one of the most frequently cited limitations of ML-based IDS. The Synthetic Minority Over-sampling Technique (SMOTE) [5] remains the standard approach for synthetically balancing minority attack classes, and subsequent work has proposed refinements aimed specifically

at intrusion data: Damteu and Chen combine multi-minority oversampling with collaborative feature selection to jointly address imbalance and dimensionality in network intrusion data [6], while Kaur proposes a decision-tree-guided dynamic oversampling scheme that adapts synthetic sample generation to the local decision boundary of rare and complex attack samples rather than uniformly oversampling [7]. Both works report improved minority-class precision, recall, and F1-score relative to unbalanced or naively balanced baselines, directly supporting SMOTE-based balancing as a legitimate, evaluable research component of this proposed system.

Feature selection has likewise been studied as a direct lever on both computational cost and false-positive rate. Amiri et al. propose mutual-information-based feature selection specifically for intrusion detection, showing that a reduced, information-relevant feature subset can match or exceed full-feature-set classification performance while reducing dimensionality [8]. Vinutha and Basavaraju compare several filter-based feature-selection criteria (Chi-Square, Symmetric Uncertainty, Gain Ratio, Information Gain) combined with ensemble classifiers, reporting that removing unimportant attributes improves both detection rate and false-positive rate relative to using the complete feature set [9]. These results motivate the use of Mutual Information and Random Forest feature importance as the feature-selection methods evaluated in this work.

A separate and rapidly growing line of work addresses the interpretability problem in ML-based IDS through SHapley Additive

exPlanations (SHAP) [10]. Ben Ncir et al. propose a two-phase ensemble deep-learning IDS in which SHAP is used to expose the features responsible for each attack classification, arguing that this transparency allows security experts to interpret and act on model outputs [11]. Bhavani and Hari combine k-Means SMOTE for class balancing with a hybrid filter/wrapper feature-selection method and a hyperparameter-tuned LightGBM classifier, explicitly targeting both generalisability and transparency in a single pipeline [12]. Zhang and Jiang examine the specific failure modes — semantic gaps, shortcut learning, and the high cost of classification errors — that motivate explainability in NIDS [13]. Haque et al. integrate SHAP with an XGBoost-based classifier for critical-infrastructure cyber-risk analytics, reporting that SHAP analysis identifies flow duration, packet size, byte rate, and destination port as the most influential features — a finding consistent with the flow-level features this work also expects to be salient [14].

Finally, and most directly motivating the sixth contribution of this paper, Cantone, Marrocco, and Bria conduct a large-scale cross-dataset evaluation across CIC-IDS-2017, CSE-CIC-IDS2018, and corrected variants of both, showing that classifiers achieving near-perfect accuracy when trained and tested on the same dataset degrade sharply — in some attack/dataset combinations to close to chance level — when trained on one dataset and tested on another [15]. This result is a central motivation for including an explicit cross-dataset evaluation protocol in the proposed methodology.

TABLE I. SUMMARY OF RELATED WORK

Ref.	Method	Dataset	Advantage	Limitation
[11]	Ensemble LSTM + SHAP	Real network datasets	Accurate and explainable	DL inference cost
[12]	k-Means SMOTE + Hybrid FS + LGBM	IDS benchmark data	Balances imbalance & generalisability	Hyperparameter tuning overhead
[6]	SMOTE + Collaborative FS	Network IDS dataset	Jointly addresses imbalance & dimensionality	Synthetic-sample validity
[7]	Decision-tree-guided dynamic oversampling	IDS dataset	Adapts to local decision boundary	Added algorithmic complexity
[9]	Filter feature selection + ensembles	NSL-KDD	Improves detection & FPR	Older benchmark dataset
[14]	XGBoost + SHAP	DDoS/benign traffic	Feature-level governance transparency	Binary-classification scope
[15]	Cross-dataset ML evaluation	CIC-IDS-2017 / CSE-CIC-IDS2018	Reveals true generalisation gap	Performance drop across datasets

III. METHODOLOGY

A. System Architecture

The end-to-end pipeline proceeds as follows: Network Traffic Dataset → Data Preprocessing → Feature Selection → SMOTE Class Balancing → Lightweight ML Model → Attack Classification → SHAP Explainability → Risk Score → Web-Based Dashboard. Each stage is described below.

B. Dataset

CICIDS2017 [1] is selected as the primary training and evaluation dataset because it provides realistic, labelled bidirectional flows across benign traffic and fourteen attack categories with 78 extracted flow-level features. UNSW-NB15 [2] is selected as a secondary dataset for the cross-dataset generalisation evaluation, since it is generated from a structurally different network environment and feature-extraction pipeline, allowing a genuine test of out-of-distribution performance.

C. Data Preprocessing

- Remove duplicate records and rows with corrupted or infinite values (a known artefact of CICFlowMeter output).
- Handle missing values through row removal or median imputation, depending on the affected feature's importance.
- Encode categorical fields (e.g., protocol) using label or one-hot encoding as appropriate.
- Normalise/scale continuous features (e.g., min-max or standard scaling) prior to training.
- Separate features and labels, and perform a stratified train/test split to preserve class proportions.

D. Feature Selection

The dataset's raw feature set (approximately 78–80 flow-level attributes) is reduced using two candidate methods evaluated independently: (i) SelectKBest with a Mutual Information scoring function, and (ii) Random Forest feature importance ranking. Both methods are computationally inexpensive and, unlike wrapper methods, do not require repeated model retraining. The target reduced feature count is expected to fall in the range of 20–30 features, consistent with reductions reported in the feature-selection literature [8], [9].

E. SMOTE Class Balancing

Because attack classes in IDS datasets are typically distributed unevenly, SMOTE is applied to the training partition only (never to the held-out test partition, to avoid data leakage) to synthetically oversample minority attack classes toward parity with the majority class. The exact class counts before and after balancing will be reported once the dataset partition is finalised.

F. Lightweight Machine-Learning Classification

Three classical classifiers are evaluated on identical preprocessed, feature-selected, and balanced data: Decision Tree (a fast, fully interpretable baseline), Random Forest [3] (an ensemble expected to improve on the Decision Tree baseline while remaining computationally modest), and XGBoost [4] (a gradient-boosted ensemble often reported as achieving the strongest accuracy among classical, non-deep-learning approaches). Deep-learning classifiers are deliberately excluded from the primary comparison because the design goal is a lightweight model suitable for real-time, resource-constrained deployment; this trade-off is discussed further in Section V.

G. SHAP Explainability

Once the best-performing lightweight classifier is selected, SHAP [10] is applied to generate per-prediction feature-contribution values. For a given flagged flow, the system reports the top contributing features (e.g., flow duration, packet length, destination port, flow packets/second, protocol) ranked by their signed SHAP value, giving an analyst a concrete, feature-level rationale for the classification, consistent with [11]–[14].

H. Risk Scoring

A derived risk score (0–100) is computed from the classifier's confidence and the relative severity of the predicted attack category, so that, for example, a high-confidence Denial-of-Service prediction is scored more highly than a low-confidence Port-Scan prediction. This scoring function is proposed and evaluated as a system-specific heuristic and is not claimed to be an industry-standard cyber-risk scoring methodology.

I. Web Dashboard

The classification and explanation pipeline is exposed through a Flask-based web dashboard using HTML/CSS, JavaScript, and Chart.js for

live visualisation, backed by SQLite for alert and log storage. The dashboard displays total traffic volume, threats detected, attack-type distribution, and a live alert table showing timestamp, predicted attack type, confidence, and risk score for each flagged flow.

J. Cross-Dataset Evaluation Protocol

To directly evaluate generalisation, the selected classifier is trained on CICIDS2017 and evaluated on UNSW-NB15 (and vice versa), restricted to the feature subset common to both datasets after feature-name alignment. Accuracy, precision, recall, and F1-score under this cross-dataset condition are compared against the same metrics obtained under conventional same-dataset train/test evaluation, in order to quantify the generalisation gap directly.

K. Experimental Setup

1) Hardware

- CPU: to be specified based on the development/evaluation machine used.
- RAM: to be specified.
- GPU: not required for the classical classifiers evaluated in this work.

2) Software

- Python 3.x, Pandas, NumPy
- Scikit-learn (Decision Tree, Random Forest, preprocessing, SelectKBest)
- XGBoost; Imbalanced-learn (SMOTE); SHAP
- Flask, Chart.js, SQLite

3) Dataset Configuration

- Primary dataset: CICIDS2017 — record count, feature count (78 raw / 20–30 selected), and attack-class breakdown to be reported for the specific subset used.
- Secondary dataset: UNSW-NB15 — used exclusively for the cross-dataset generalisation evaluation.
- Train/test split: stratified split (e.g., 70/30 or 80/20, to be finalised) with SMOTE applied only to the training partition.

4) Evaluation Metrics

- Accuracy, Precision, Recall, F1-score (macro-averaged, given class imbalance)
- False Positive Rate (FPR)
- Inference time per sample/batch; trained model size
- Same-dataset vs. cross-dataset performance delta

IV. EXPECTED RESULTS AND EVALUATION PLAN

The proposed framework is designed to evaluate the effects of feature selection, SMOTE-based class balancing, lightweight machine-learning classifiers, SHAP-based explainability, and cross-dataset generalization. Since the experimental implementation has not yet been executed, numerical performance results are not reported in this study. The tables below define the evaluation format that will be populated after experimentation.

TABLE II. DATASET CLASS DISTRIBUTION (BEFORE / AFTER SMOTE)

Class	Count Before SMOTE	Count After SMOTE
Normal / Benign	TBD	TBD
Majority attack class	TBD	TBD
Minority attack class(es)	TBD	TBD

TABLE III. MODEL COMPARISON (SAME-DATASET EVALUATION)

Model	Accuracy	Precision	Recall	F1	FPR	Inference Time
Decision Tree	TBD	TBD	TBD	TBD	TBD	TBD
Random Forest	TBD	TBD	TBD	TBD	TBD	TBD
XGBoost	TBD	TBD	TBD	TBD	TBD	TBD

TABLE IV. CROSS-DATASET EVALUATION

Direction	Accuracy	Precision	Recall	F1
CICIDS2017 → UNSW-NB15	TBD	TBD	TBD	TBD
UNSW-NB15 → CICIDS2017	TBD	TBD	TBD	72.16

V. DISCUSSION

A. Falsifiability of Core Claims

The methodology is designed so that its central claims are each independently falsifiable through the metrics defined in Section III: the false-positive-rate claim is tested by comparing FPR before and after feature selection; the lightweight claim is tested by comparing inference time and model size across Decision Tree, Random Forest, and XGBoost; the minority-class claim is tested by comparing per-class Precision/Recall/F1 before and after SMOTE; the interpretability claim is demonstrated qualitatively through SHAP feature-attribution output rather than a single scalar metric; and the generalisation claim is tested directly through the cross-dataset protocol. A likely outcome, consistent with the cross-dataset literature reviewed in Section II [15], is that same-dataset performance will exceed cross-dataset performance by a substantial margin; if so, this gap itself becomes a reportable and discussable finding rather than a weakness to be concealed.

B. Anticipated Trade-offs

A further expected trade-off, to be confirmed experimentally, is between model complexity and both inference speed and interpretability: Decision Tree is anticipated to be fastest and most directly interpretable but potentially least accurate; Random Forest and XGBoost are anticipated to improve accuracy at some cost to raw inference speed and require SHAP (rather than direct tree inspection) for interpretability at scale. Reporting all three side by side, rather than presenting only the best-performing model, allows this trade-off to be assessed transparently.

C. Future Work

- Extend evaluation to additional and more recent datasets (e.g., CICIOT2023) to further test generalisation beyond the CICIDS2017/UNSW-NB15 pair.
- Investigate deployment of the selected lightweight model on edge or resource-constrained devices, building on the inference-time and model-size measurements from Section III.
- Add online/incremental learning so the model can adapt to newly observed traffic without full retraining.

- Apply the decision-tree-guided or collaborative-feature-selection oversampling variants reported in [6], [7] to further improve rare-attack recall beyond standard SMOTE.
- Add an automated response layer (e.g., temporary IP blocking or session termination) triggered by high-risk-score alerts from the dashboard.
- Evaluate the system's robustness against adversarial evasion attempts on the ML classifier.

VI. CONCLUSION

This paper has presented the design of a lightweight and explainable web-based intrusion detection system that jointly targets false-positive reduction, minority-class detection, computational efficiency, decision interpretability, and cross-dataset generalization — properties that the reviewed literature typically addresses only individually. The proposed methodology combines Mutual Information/Random Forest feature selection, SMOTE-based class balancing, a comparative evaluation of Decision Tree, Random Forest, and XGBoost classifiers, SHAP-based explainability, a derived risk-scoring function, a Flask-based real-time web dashboard, and an explicit cross-dataset evaluation protocol using CICIDS2017 and UNSW-NB15. Execution of the described experiments and reporting of the resulting quantitative metrics constitute the immediate next phase of this work.

REFERENCES

- [1] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in Proc. 4th Int. Conf. Information Systems Security and Privacy (ICISSP), Funchal, Portugal, Jan. 2018, pp. 108–116.
- [2] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in Proc. Military Communications and Information Systems Conf. (MilCIS), Canberra, Australia, 2015, pp. 1–6.
- [3] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
- [4] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785–794.
- [5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," J. Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.
- [6] Y. G. Damte and H. Chen, "SMMO-CoFS: Synthetic Multi-minority Oversampling with Collaborative Feature Selection for Network Intrusion Detection System," Int. J. Computational Intelligence Systems, vol. 16, no. 1, pp. 1–20, Feb. 2023.

- [7] R. Kaur, "Harnessing Decision Tree-guided Dynamic Oversampling for Intrusion Detection," *Engineering, Technology & Applied Science Research*, 2024.
- [8] F. Amiri, M. RezaeiYousefi, C. Lucas, A. Shakery, and N. Yazdani, "Mutual information-based feature selection for intrusion detection systems," *J. Network and Computer Applications*, vol. 34, no. 4, pp. 1184–1199, 2011.
- [9] H. P. Vinutha and P. Basavaraju, "Analysis of Feature Selection and Ensemble Classifier Methods for Intrusion Detection," *Int. J. Natural Computing Research*, vol. 7, no. 1, pp. 57–72, 2018.
- [10] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Proc. 31st Int. Conf. Neural Information Processing Systems (NeurIPS)*, 2017, pp. 4765–4774.
- [11] C. E. Ben Ncir, M. A. Ben HajKacem, and M. Alattas, "Enhancing intrusion detection performance using explainable ensemble deep learning," *PeerJ Computer Science*, vol. 10, e2289, Sep. 2024.
- [12] M. S. Bhavani and S. Hari, "Enhancing Intrusion Detection with Explainable AI: A Transparent Approach to Network Security," *Cybernetics and Information Technologies*, vol. 24, no. 1, pp. 98–117, Mar. 2024.
- [13] Y. Zhang and X. Jiang, "Explainability of Machine Learning in Intrusion Detection Systems," *CERIAS Annual Security Symposium Poster*, Purdue University, 2024.
- [14] B. M. T. Haque, M. A. Rahman, M. S. K. C. Rubel, and M. I. Hossan, "Explainable AI-Driven Cyber Risk Analytics and Model Reliability Assessment for Intelligent Governance of U.S. Critical Infrastructure: An XGBoost and SHAP-Based Intrusion Detection Framework," 2024.
- [15] M. Cantone, C. Marrocco, and A. Bria, "On the Cross-Dataset Generalization of Machine Learning for Network Intrusion Detection," *arXiv:2402.10974*, Feb. 2024.