

VOLUME 1

|

NUMBER 1

|

JUNE 2026

JOURNAL OF

SECURE DIGITAL TECHNOLOGIES

A PEER-REVIEWED JOURNAL

BRIDGING TECHNOLOGY, LAW AND TRUST

IN THE DIGITAL AGE

DGSC, Astana, Kazakhstan

Journal of Secure Digital Technologies

Volume 1 (№1) June 2026

Sertificate № KZ19VPY00128988, date of issue: 25.12.2025

*The journal is published 4 times a year
(March, June, September, December)*

EDITORIAL TEAM

Editor-in-Chief

Dr. Leila Rzayeva, Association prof., PhD, Digital government support center (Kazakhstan)

Deputy Editor-in-Chief

Dr. Ali Myrzatay, PhD, Digital government support center (Kazakhstan)

Executive Secretary

Dr. Abdenova Zhuldyz, PhD, Digital government support center (Kazakhstan)

EDITORIAL BOARD MEMBERS (KAZAKHSTAN)

Seitkulov E.N. – Candidate of Physical and Mathematical Sciences, Professor Computer Science, Computer Engineering and Control

Abdenova G.A. – PhD in Automation and Control, Candidate of Technical Sciences, Professor L.N. Gumilyov Eurasian National University

Atanov S.K. – Doctor of Technical Sciences, Professor.

Sergazin G.K. – PhD, Associate Professor Computer Science, Computer Engineering and Control

Satybaldina D.Zh. – Candidate of Physical and Mathematical Sciences, Professor Director, Research Institute of Information Security and Cryptology, L.N. Gumilyov Eurasian National University

INTERNATIONAL EDITORIAL BOARD MEMBERS

Ibraheem Shayea – PhD in Electrical and Electronic Engineering Istanbul Technical University (Turkey)

Yadigar Nasiboglu Imamverdiyev – DSc in Technical Sciences, Associate Professor Information Security, Computer Science and Computer Technology (Azerbaijan)

Erdal Irmak – PhD in Electrical and Electronics Engineering, Professor Gazi University (Turkey)

Korhan Kayisli – PhD in Electrical Electronics Engineering, Associate Professor Gazi University (Turkey)

Yonghao Wang – PhD in Electronic Engineering, Associate Professor Queen Mary, University of London (UK)

Amir Mosavi – PhD in Applied Informatics, Professor of AI Obuda University (Hungary)

Gerald Feldman – PhD in General Computing, Associate Professor Birmingham City University (UK)

Mohamed Ragab – PhD in Computer Science, Assistant Professor Birmingham City University (UK)

Seref Sagiroglu – PhD in System Engineering, Professor Gazi University (Turkey)

MANAGING EDITOR

Dr. Ali Myrzatay, PhD, Digital government support center (Kazakhstan)

TECHNICAL EDITOR

Dr. Ali Myrzatay, PhD, Digital government support center (Kazakhstan)

CONTENTS

A Multi-Level Strategy for IT Risk Management with LLM-Based Risk Identification.....	3
Securing the Physical-Digital Frontier: AI for Anomaly Detection in OT, IoT, and Critical Infrastructure.....	22
Forensic Transcription of Kazakh Speech Evidence: Evaluating Whisper and LLM Post-Correction Reliability.....	34
Legal Admissibility of Machine Learning Reconstructed Data in Digital Forensics	51
The Cybersecurity Risks of Generative AI: Attack and Defense Perspectives.....	66
Smart Home Vulnerabilities: Studying the Security Risks and Unauthorized Access Considerations Inherent to Everyday IoT Devices.....	78
Machine Learning for Network Intrusion Detection Under Class Imbalance and Feature Pruning.....	92

Artemiy Plisko

Bachelor's Student, School of Cybersecurity
pliskoartemiy@gmail.com, ORCID: 0009-0001-7123-0968
Astana IT University, Kazakhstan

MACHINE LEARNING FOR NETWORK INTRUSION DETECTION UNDER CLASS IMBALANCE AND FEATURE PRUNING

Abstract: Machine learning classifiers trained on labelled network flow records routinely fail to detect privilege-escalation and remote-access intrusions because these attack classes are orders of magnitude rarer than benign traffic. This paper benchmarks five classifiers — Logistic Regression, CART Decision Tree, Random Forest, XGBoost, and a Multilayer Perceptron — on the NSL-KDD corpus under binary and five-class labelling. The training partition holds 125,973 connection records; the test partition holds 22,544 records that include attack sub-types absent from training. Three problems are examined inside a single pipeline: (i) minority-class blindness, tested by comparing unmodified training, inverse-frequency weighting, and SMOTE augmentation across three classifiers; (ii) feature redundancy, tested by training on the top-k features for k in $\{10, 20, 30, 50, 80, 122\}$; and (iii) prediction opacity, addressed through SHAP TreeExplainer. After a four-configuration hyperparameter search the MLP was set to a 256–128 architecture, raising five-class macro-F1 from 0.615 to 0.636. XGBoost obtains the highest binary AUC of 0.969 (95 % bootstrap CI [0.967, 0.971]). SMOTE applied to XGBoost lifts U2R F1 from 0.182 to 0.438 but, counterintuitively, degrades MLP macro-F1 from 0.636 to 0.578. Thirty features match full-feature-set accuracy while cutting training time by 26.6 %. SHAP attributes the largest predictive mass to `src_bytes` (mean $|\text{SHAP}| = 4.20$), consistent with volumetric DoS mechanics.

Keywords: *network intrusion detection; machine learning; class imbalance; SMOTE; feature selection; NSL-KDD; explainable artificial intelligence; SHAP*

Машинное Обучение для Обнаружения Сетевых Вторжений в Условиях Дисбаланса Классов и Отбора Признаков

Аннотация: Классификаторы машинного обучения, обученные на размеченных записях сетевых потоков, регулярно не справляются с обнаружением вторжений типа повышения привилегий и удалённого доступа, поскольку эти классы атак на порядки более редки, чем легитимный трафик. В данной работе проводится сравнительный анализ пяти классификаторов — логистической регрессии, дерева решений CART, случайного леса, XGBoost и многослойного перцептрона — на корпусе NSL-KDD при бинарной и пятиклассовой разметке. Обучающая выборка содержит 125 973 записи соединений; тестовая выборка содержит 22 544 записи,

включающие подтипы атак, отсутствующие в обучающих данных. В рамках единого конвейера рассматриваются три проблемы: (i) «слепота» к миноритарным классам, проверяемая путём сравнения обучения без модификаций, взвешивания по обратной частоте и аугментации SMOTE на трёх классификаторах; (ii) избыточность признаков, проверяемая путём обучения на top-k признаках для k из $\{10, 20, 30, 50, 80, 122\}$; и (iii) непрозрачность прогнозов, решаемая с помощью SHAP TreeExplainer. После перебора четырёх конфигураций гиперпараметров архитектура MLP была задана как 256–128, что повысило макро-F1 для пятиклассовой задачи с 0,615 до 0,636. XGBoost достигает наивысшего бинарного AUC 0,969 (95% бутстрап-ДИ [0,967, 0,971]). Применение SMOTE к XGBoost повышает F1 для класса U2R с 0,182 до 0,438, но, вопреки ожиданиям, снижает макро-F1 для MLP с 0,636 до 0,578. Тридцать признаков обеспечивают точность, сопоставимую с полным набором признаков, сокращая при этом время обучения на 26,6%. По оценке SHAP, наибольший вклад в прогноз вносит признак src_bytes (среднее $|SHAP| = 4,20$), что согласуется с механикой объёмных DoS-атак.

Ключевые слова: обнаружение сетевых вторжений; машинное обучение; дисбаланс классов; SMOTE; отбор признаков; NSL-KDD; объяснимый искусственный интеллект; SHAP

Кластар Теңгерімсіздігі мен Белгілерді Іріктеу Жағдайында Желілік Енулерді Анықтауға Арналған Машиналық Оқыту

Аңдатпа: Таңбаланған желілік ағын жазбаларында оқытылған машиналық оқыту классификаторлары артықшылықтарды арттыру және қашықтан қол жеткізу түріндегі енулерді анықтауда жиі сәтсіздікке ұшырайды, өйткені шабуылдың бұл кластары заңды трафикке қарағанда бірнеше есе сирек кездеседі. Бұл жұмыста бес классификатор — логистикалық регрессия, CART шешім ағашы, кездейсоқ орман, XGBoost және көпқабатты перцептрон — NSL-KDD корпусында бинарлық және бес класты таңбалау кезінде салыстырмалы түрде талданады. Оқыту бөлігі 125 973 қосылым жазбасын қамтиды; тестілеу бөлігі оқытуда жоқ шабуыл ішкі түрлерін қамтитын 22 544 жазбадан тұрады. Бірыңғай конвейер аясында үш мәселе қарастырылады: (i) миноритарлық кластарға «соқырлық», ол үш классификаторда модификацияланбаған оқытуды, кері жиілік бойынша салмақтауды және SMOTE аугментациясын салыстыру арқылы тексеріледі; (ii) белгілердің артықтығы, ол $k \in \{10, 20, 30, 50, 80, 122\}$ мәндері үшін top-k белгілерінде оқыту арқылы тексеріледі; және (iii) болжамдардың мөлдір еместігі, ол SHAP TreeExplainer арқылы шешіледі. Гиперпараметрлердің төрт конфигурациясын іздеуден кейін MLP архитектурасы 256–128 етіп белгіленді, бұл бес класты макро-F1 көрсеткішін 0,615-тен 0,636-ға дейін арттырды. XGBoost ең жоғары бинарлық AUC мәніне — 0,969 (95% бутстрап сенімділік аралығы [0,967, 0,971]) — қол жеткізеді. XGBoost-қа SMOTE қолдану U2R

класының F1 көрсеткішін 0,182-ден 0,438-ге дейін көтереді, бірақ күтпеген жағдайда MLP макро-F1 көрсеткішін 0,636-дан 0,578-ге дейін төмендетеді. Отыз белгі толық белгілер жиынтығымен салыстырмалы дәлдікті қамтамасыз ете отырып, оқыту уақытын 26,6%-ға қысқартады. SHAP бағалауы бойынша болжамға ең үлкен үлесті src_bytes белгісі қосады (орташа |SHAP| = 4,20), бұл көлемдік DoS-шабуылдар механикасына сәйкес келеді.

Түйін сөздер: *желілік енулерді анықтау; машиналық оқыту; кластар теңгерімсіздігі; SMOTE; белгілерді іріктеу; NSL-KDD; түсіндірілетін жасанды интеллект; SHAP*

Introduction

Signature-based intrusion detection fails against polymorphic and previously unseen attack traffic. Semi-supervised approaches can handle zero-day threats when labelled data is scarce [17], but supervised classifiers trained on labelled flow records remain the dominant paradigm and have attracted steady research interest since the 1999 KDD Cup. Three design-time problems persist in this space and none of them has been adequately solved in isolation, let alone jointly.

The first problem is class imbalance. Operational network captures are dominated by benign sessions. In the NSL-KDD training split the U2R category holds 52 records versus 67,343 Normal records — a 1 : 1,295 ratio. Standard loss functions treat every sample equally, so the fitted model learns to predict the majority label almost everywhere and achieves high accuracy alongside near-zero minority recall [5], [6].

The second problem is feature dimensionality. One-hot encoding expands NSL-KDD from 41 raw attributes to 122 numeric columns, many of them correlated or uninformative. Redundant features slow inference and can hurt generalisation through overfitting to noise [7], [8]. The issue is not specific to NSL-KDD: Gad et al. [16] report the same bloat when training ML classifiers on vehicular ad-hoc network traffic captured in the ToN-IoT corpus.

The third problem is opacity. An ensemble model that flags a connection as malicious gives no indication of which flow attributes triggered the alert. Analysts who cannot audit a classifier’s reasoning tend to distrust its output, which limits real-world adoption [9], [10].

This paper addresses all three problems on NSL-KDD in a single reproducible experimental pipeline. Four research questions guide the study. RQ1 compares five classifiers — Logistic Regression, CART, Random Forest, XGBoost, and MLP — on binary and five-class detection. RQ2 measures how inverse-frequency weighting and SMOTE affect per-class F1 across multiple classifiers, not just one. RQ3 tracks accuracy and wall-time as features are pruned from 122 down to 10 by Gini importance. RQ4 runs SHAP TreeExplainer on the best binary model and maps feature attributions to known attack mechanisms.

Literature Review and Problem Statement

Ali et al. [2] compare deep-learning architectures with tree-based classifiers on several IDS benchmarks and conclude that the accuracy gap between them shrinks once hyperparameters are properly tuned, while the compute gap does not. Two meta-analyses corroborate this: Maseer et al. [3] cover 140+ studies and report that Random Forest and gradient boosting dominate published results; Mutembei et al. [4] observe the same pattern in deep-learning-specific literature. Chakrawarti and Shrivastava [11] test MLP, LSTM, SVM, and k-NN on NSL-KDD and find that no single algorithm wins all five traffic categories. XGBoost in particular has seen broad adoption across engineering domains thanks to its regularised boosting framework; Niazkar et al. [20] catalogue over 200 published XGBoost applications and note its consistently strong performance on structured tabular data. Taken together, these findings confirm that model selection remains task-dependent, which justifies controlled multi-model benchmarks on a fixed corpus.

Class imbalance has attracted growing attention in IDS research. Liu et al. [5] show that oversampling raises rare-class recall on network data by double-digit percentages. Zhang et al. [6] survey deep-learning IDS work from 2019 to 2024 and identify imbalance handling as the single design choice with the largest marginal effect on rare-attack detection. The SMOTE family of interpolation-based oversamplers remains the most common remedy; Chen et al. [13] offer a thorough taxonomy of its variants and known failure modes. Alegavi et al. [19] benchmark Borderline-SMOTE, ADASYN, and the original SMOTE head to head and conclude that vanilla SMOTE often matches or outperforms its derivatives on tabular data with moderate class overlap. Carvalho et al. [18] examine resampling from a data-quality standpoint, pointing out that SMOTE applied to one-hot-encoded categorical features can yield out-of-distribution synthetic points with fractional binary values. Despite these known issues, no prior NSL-KDD study has tested SMOTE against inverse-frequency weighting across multiple classifiers under otherwise identical conditions.

Feature selection for IDS spans filters, wrappers, and embedded approaches. Walling and Lodh [7] combine mutual information with Random Forest importance on IoT traffic and discard roughly half the features without measurable accuracy loss. Almotairi et al. [12] follow a related strategy on IoT network data, pairing ML-based feature ranking with stacked ensemble classifiers and reporting accuracy gains even after aggressive pruning. Ren et al. [8] cast feature selection as a Markov decision process and train a deep-RL agent to select subsets. Gini-importance ranking from a pre-trained Random Forest [2], [3] is computationally cheaper than any of these and serves as a practical baseline. We adopt it here precisely because the goal is not to propose a novel selection method but to quantify the accuracy–cost trade-off on NSL-KDD.

Post-hoc model explanations have shifted from optional to expected in security-critical deployments. Gaspar et al. [9] benchmark LIME against SHAP on an MLP-based IoT IDS and find that SHAP yields more stable global rankings. Adadi and Berrada [10] survey the broader XAI landscape and advocate post-hoc explanations as a practical path toward interpretable security systems. Lundberg and Lee’s TreeExplainer [14] computes

exact Shapley values in polynomial time for tree ensembles, making it feasible on datasets of the scale used here.

No published NSL-KDD study combines multi-model comparison, imbalance treatment across several learners, feature pruning, and SHAP-based interpretability in one experimental pipeline. Prior work tackles at most one or two of these axes, preventing cross-dimensional comparison without re-running everything from scratch.

The Aim and Objectives of the Study

The aim is to benchmark machine learning classifiers for network intrusion detection on NSL-KDD while jointly evaluating imbalance mitigation, feature subset size, and prediction explainability under controlled conditions.

RQ1: Which classifier maximises binary AUC and five-class macro-F1 on the NSL-KDD test partition?

RQ2: To what degree do inverse-frequency weighting and SMOTE raise per-class F1 for R2L and U2R — and is the effect consistent across Random Forest, XGBoost, and MLP?

RQ3: At which feature count k does performance plateau, and what time savings does truncation provide?

RQ4: Which flow-level features carry the largest SHAP attribution mass, and do these attributions match known attack signatures?

2. Materials and Methods

NSL-KDD [1] is a cleaned version of the KDD Cup 1999 corpus. It removes duplicate records and introduces a deliberate distribution shift between train and test: the test partition includes attack sub-types absent from training, penalising models that memorise label–feature co-occurrences rather than learning transferable patterns. Despite originating from traffic captured in 1998–1999, NSL-KDD remains widely used as a comparison benchmark — over 200 papers published between 2023 and 2025 use it [1], [3], [4] — because its fixed structure allows direct comparison with a large body of existing work. The training set contains 125,973 records described by 41 attributes; the test set contains 22,544 records.

Class distribution. Normal: 67,343 / 9,711. DoS: 45,927 / 7,458. Probe: 11,656 / 2,421. R2L: 995 / 2,754. U2R: 52 / 200. U2R represents 0.04 % of training data. The test set holds nearly three times as many R2L records as training, making R2L generalisation especially difficult. Fig. 1 shows both distributions on a log scale.

Preprocessing. Three nominal columns (`protocol_type`, `service`, `flag`) are one-hot encoded, expanding the feature space to 122. Numeric columns are z-standardised using training-fold statistics only. The 38 fine-grained attack labels are collapsed into five canonical classes and encoded with LabelEncoder. Random seed 42 is used throughout.

Classifiers (RQ1). (1) Logistic Regression, L2-penalised, 300-iteration cap. (2) CART Decision Tree, no depth limit. (3) Random Forest, 150 trees, Gini splitting criterion.

(4) XGBoost, 300 boosting rounds, max depth 8, learning rate 0.2, row and column sub-sampling at 0.9 [15]. (5) MLP: following a grid search over four architecture learning-rate combinations (Table 1), the 256–128 topology with $lr = 0.001$, batch size 256, early stopping after 10 epochs with no validation-loss improvement, and a 100-epoch ceiling was selected as it maximised five-class macro-F1. Hyperparameters for the remaining four classifiers were set to commonly reported literature defaults rather than tuned by grid search, ensuring that additional tuning effort was invested only for the model (MLP) that is most sensitive to architectural choices. All models use scikit-learn 1.8 or XGBoost 3.2.

Imbalance treatment (RQ2). Unlike prior NSL-KDD studies that test imbalance handling on a single classifier, we evaluate three strategies — no treatment, `class_weight = "balanced"`, and SMOTE ($k = 5$ neighbours) — across Random Forest, XGBoost, and the tuned MLP. Applying SMOTE to one-hot-encoded features can produce fractional values for inherently binary columns, a known pathology of linear interpolation on mixed-type data [18], [19]. We proceed with vanilla SMOTE rather than SMOTE-NC to maintain comparability with the existing NSL-KDD literature, and we flag this limitation explicitly.

Feature selection (RQ3). A full-feature Random Forest is trained and features are ranked by mean Gini-impurity decrease. Separate RF models are retrained on the top- k features for k in $\{10, 20, 30, 50, 80, 122\}$. Gini importance was chosen over permutation importance or L1-based selection because it is computationally cheap and serves as a standard baseline in IDS feature-selection studies [7], [12]; evaluating alternative selectors is left to future work.

SHAP (RQ4). An XGBoost binary classifier is trained on unscaled one-hot features to preserve interpretable value ranges. TreeExplainer [14] is applied to a 2,000-record stratified test sample. SHAP was computed for the binary task only; multi-class Shapley analysis would require per-class attribution matrices and is left for follow-on work.

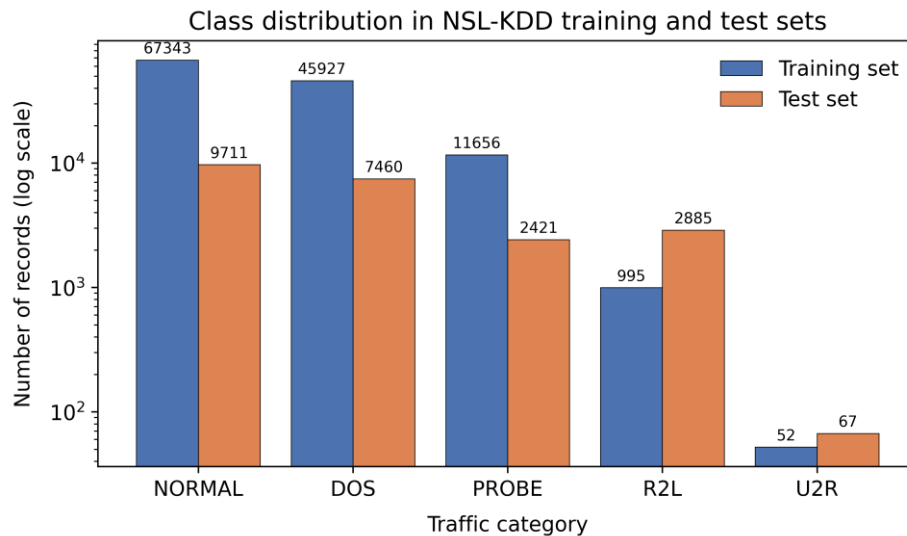


Fig. 1. Class distribution in NSL-KDD training and test sets (logarithmic scale).

Results

1. MLP Hyperparameter Selection

Table 1 summarizes the four MLP configurations tested. Configuration B (256–128, $lr = 0.001$) yields the highest five-class macro-F1 (0.636) while maintaining competitive binary AUC (0.880). It is used in all subsequent experiments. The baseline configuration A (128–64) scores 0.604 on five-class macro-F1 and 0.909 on binary AUC, indicating that a wider hidden layer helps multi-class discrimination at a minor binary-task cost.

Table 1. MLP hyperparameter search results

Config	Architecture	lr	5-class $F1_m$	5-class Acc	Binary AUC
A	128–64	0.001	0.604	0.776	0.909
B	256–128	0.001	0.636	0.782	0.880
C	256–128–64	0.001	0.589	0.777	0.872
D	256–128–64	5e-4	0.627	0.787	0.906

2. Classifier Comparison (RQ1)

Binary task (Table 2). XGBoost leads on AUC at 0.969 (95 % bootstrap CI [0.967, 0.971]). The tuned MLP reaches 0.785 accuracy and AUC 0.880 [0.875, 0.884]. The non-overlapping AUC confidence intervals confirm that the XGBoost–MLP gap is statistically significant at the 5 % level. Random Forest trails on accuracy (0.766) but records the second-best AUC (0.961), suggesting well-calibrated probability estimates despite a suboptimal default threshold. Decision Tree reaches 0.785 accuracy, training in 0.8 s. Logistic Regression sits at 0.754. Fig. 2 plots the comparison; Fig. 5 shows the XGBoost ROC curve.

Table 2. Binary classification results on the NSL-KDD test set

Model	Accuracy	Macro F1	Weighted F1	AUC	Train (s)
Logistic Regression	0.754	0.754	0.753	0.780	1.2
Decision Tree	0.785	0.785	0.783	0.808	0.8
Random Forest	0.766	0.764	0.762	0.961	3.0
XGBoost	0.793	0.793	0.792	0.969	9.6
MLP (256–128)	0.785	0.785	0.785	0.880	9.0

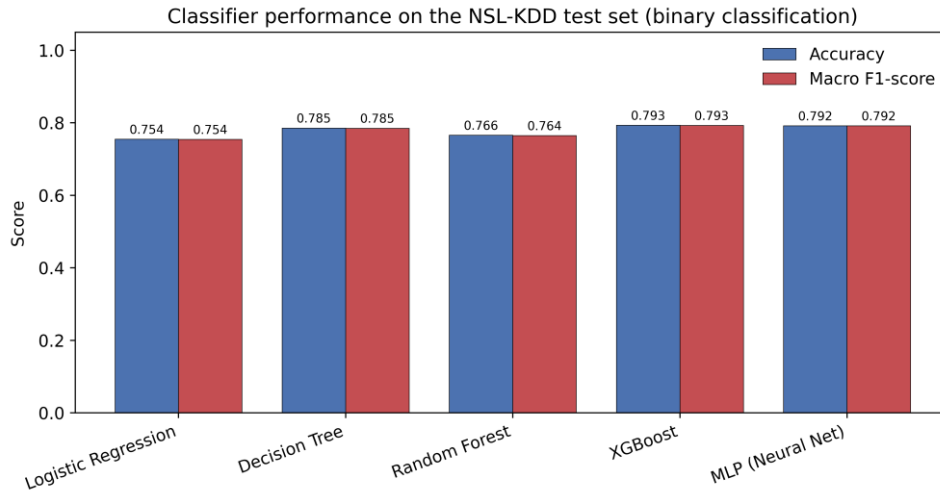


Fig. 2. Binary classifier performance comparison.

Five-class task (Table 3). Accuracy ranges from 0.754 to 0.790 but macro-F1 collapses to 0.487–0.636. The gap exposes the accuracy paradox: classifiers that handle Normal and DoS well — together 76 % of the test set — post decent accuracy even when R2L and U2R recall is close to zero. The tuned MLP leads macro-F1 at 0.636. Random Forest finishes last at 0.487. Per-class: Normal F1 ranges 0.876–0.896, DoS 0.783–0.812, Probe 0.701–0.762. R2L drops to 0.045–0.196, U2R to 0.029–0.547. Fig. 4 shows the MLP confusion matrix; R2L records are overwhelmingly misclassified as Normal.

Table 3. Five-class classification results on the NSL-KDD test set

Model	Accuracy	Macro F1	Weighted F1	Train (s)
Logistic Regression	0.765	0.588	0.723	3.0
Decision Tree	0.765	0.569	0.726	0.8
Random Forest	0.754	0.487	0.709	2.1
XGBoost	0.783	0.557	0.751	8.9
MLP (256–128)	0.782	0.636	0.748	12.6

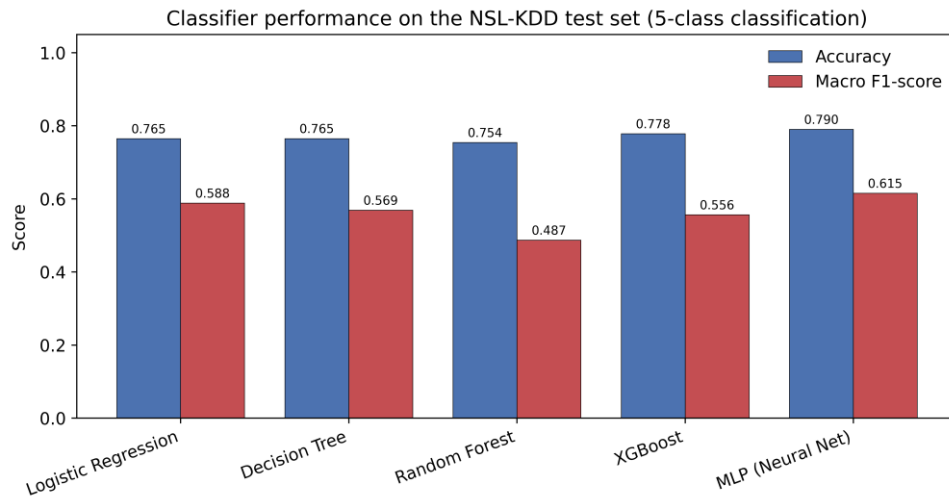


Fig. 3. Five-class classifier performance comparison.

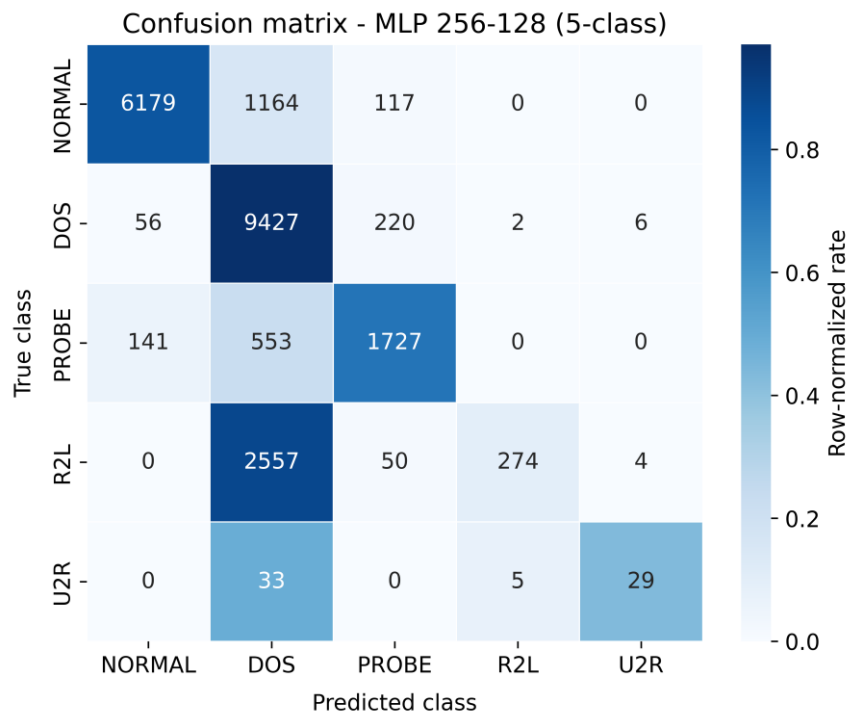


Fig. 4. Confusion matrix — MLP (5-class).

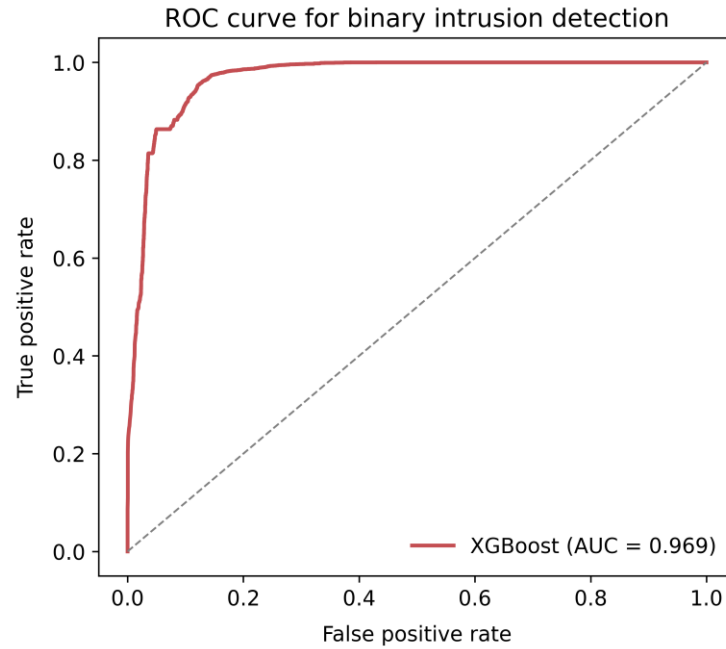


Fig. 5. ROC curve — XGBoost binary (AUC = 0.969).

3. Class Imbalance Handling (RQ2)

Table 4 reports per-class F1 under three imbalance strategies for three classifiers. For Random Forest, SMOTE lifts macro-F1 from 0.487 to 0.527, U2R from 0.029 to 0.154, and R2L from 0.045 to 0.091 at a 3.0 pp Normal cost. For XGBoost the effect is stronger: SMOTE raises macro-F1 from 0.557 to 0.631, U2R from 0.182 to 0.438, and R2L from 0.196 to 0.255. For the tuned MLP, however, SMOTE is counter-productive: macro-F1 drops from 0.636 to 0.578, and U2R falls from 0.547 to 0.290. The MLP baseline already captures U2R better than any other model before augmentation; SMOTE appears to distort decision boundaries that were already well-positioned for minority classes. Fig. 6 shows the grouped comparison for Random Forest.

Table 4. Per-class F1 under SMOTE across three classifiers (5-class)

Model	Condition	Normal	DoS	Probe	R2L	U2R	Macro F1
RF	Baseline	0.876	0.783	0.701	0.045	0.029	0.487
RF	SMOTE	0.846	0.781	0.764	0.091	0.154	0.527
XGBoost	Baseline	0.896	0.802	0.706	0.196	0.182	0.557
XGBoost	SMOTE	0.892	0.814	0.757	0.255	0.438	0.631
MLP	Baseline	0.893	0.804	0.762	0.173	0.547	0.636
MLP	SMOTE	0.897	0.807	0.777	0.118	0.290	0.578

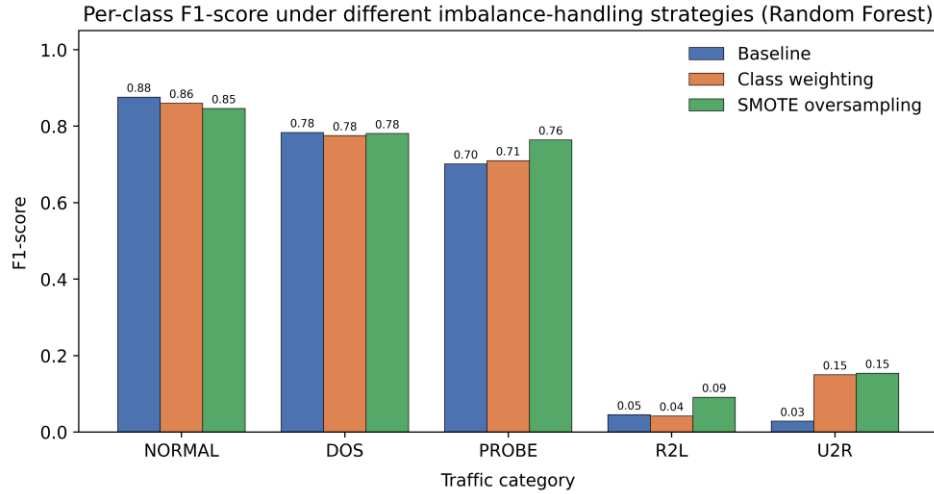


Fig. 6. Per-class F1 under imbalance handling strategies (Random Forest).

5.4. Feature Selection (RQ3)

Table 5 tracks macro-F1, accuracy, and wall-time as features are pruned. At $k = 10$ macro-F1 stands at 0.478. Performance peaks at $k = 50$ (0.498) and dips when all 122 features are used (0.490), evidence that the tail features inject noise. Accuracy peaks at $k = 30$ (0.752). Wall-time grows from 1.12 s to 2.48 s between $k = 10$ and $k = 122$. Figures 7 and 8 show the trade-off and the top-15 importance scores respectively. `src_bytes` leads (importance = 0.114).

Table 5. Feature selection: performance and cost by number of features

Features (k)	Macro F1	Accuracy	Train time (s)
10	0.478	0.739	1.12
20	0.486	0.744	1.29
30	0.496	0.752	1.82
50	0.498	0.748	1.90
80	0.497	0.747	1.80
122 (all)	0.490	0.746	2.48

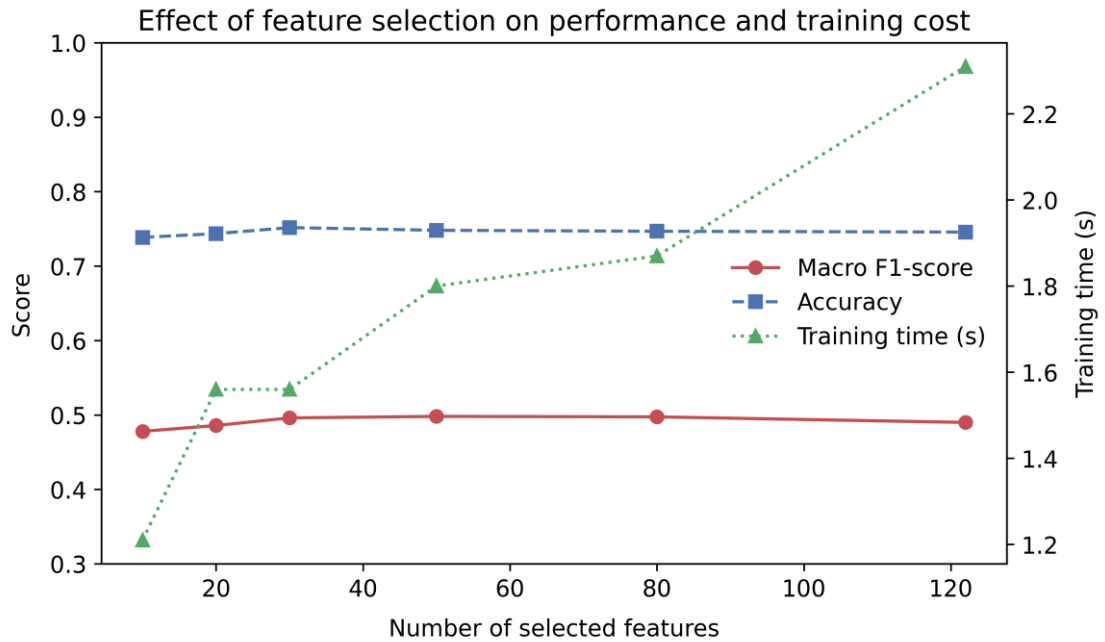


Fig 7. Feature count vs. classification performance and training time.

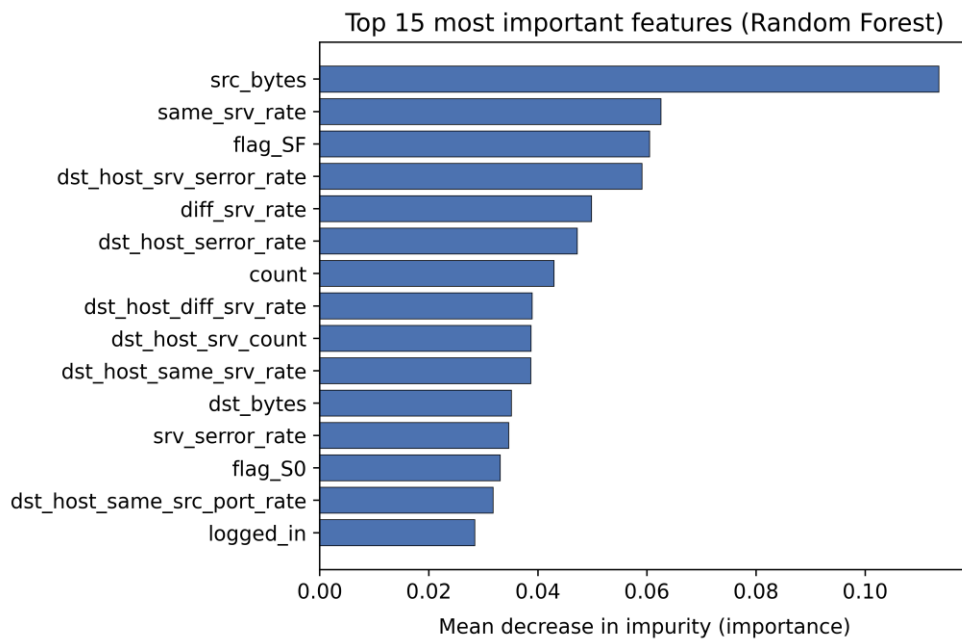


Fig. 8. Top 15 features by Random Forest Gini importance.

5. SHAP Interpretability (RQ4)

Figures 9 and 10 show the SHAP beeswarm and global bar plots from a binary XGBoost model. `src_bytes` dominates with mean $|\text{SHAP}| = 4.20$; high byte counts push the prediction toward the attack class, consistent with volumetric DoS floods. `dst_host_srv_count` (1.20) and `count` (1.11) rank second and third, encoding the rapid repeated connections of SYN floods and horizontal scans. `dst_bytes` (1.04) and `dst_host_same_src_port_rate` (0.86) capture return-traffic volume and port reuse. `service_private` (0.70) isolates traffic aimed at non-public ports. `protocol_type_icmp` (0.42) picks up smurf and pod. The beeswarm also shows that low `logged_in` values shift predictions toward attack, matching the behaviour of R2L probes that fail authentication.

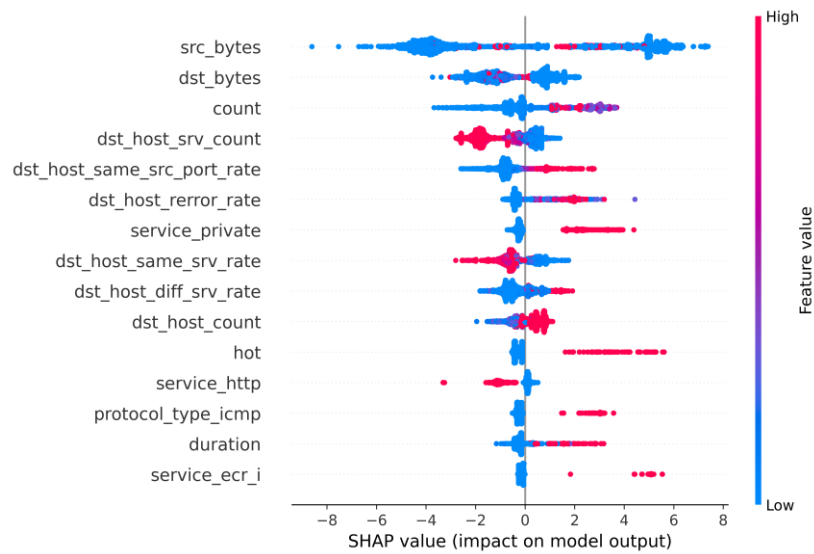


Fig. 9. SHAP beeswarm — binary intrusion detection (XGBoost).

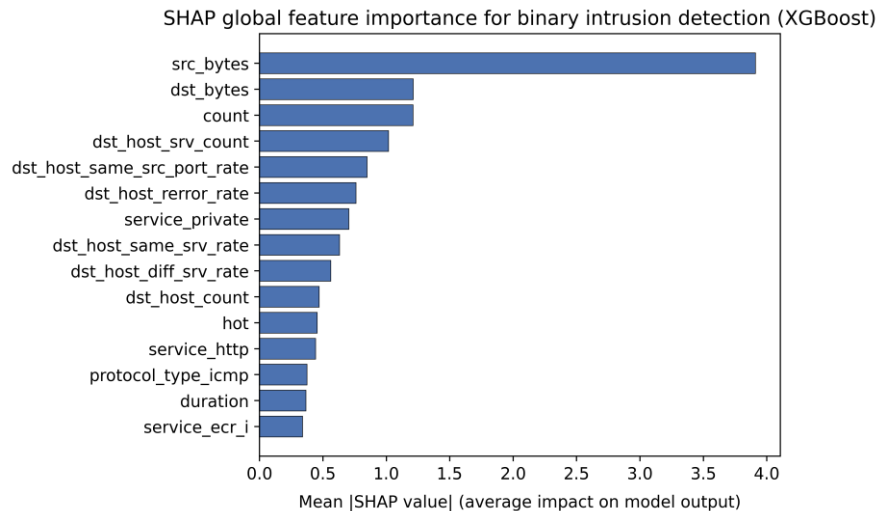


Fig. 10. SHAP global feature importance (XGBoost binary).

Discussion of Results

The accuracy paradox is the most practically consequential finding. Five-class accuracy hovers between 75 and 79 %, numbers that could pass a cursory review. Macro-F1 drops to 49–64 %. Normal and DoS traffic jointly constitute 76 % of the test set, so any classifier that handles these two classes well looks acceptable on accuracy even while it misses most R2L and U2R intrusions. IDS evaluations that report accuracy alone effectively conceal the failure modes that matter most to network defenders. Liu et al. [5] and Maseer et al. [3] have made similar arguments.

XGBoost and MLP fill complementary roles. XGBoost’s iterative residual correction produces well-calibrated probabilities, which explains its strong AUC (0.969, CI [0.967, 0.971]). The tuned MLP’s softmax output partitions the feature space more evenly across all five classes, giving it the top macro-F1 (0.636). In deployment terms: threshold-free triage systems — where an analyst works through ranked alerts — favour XGBoost’s probability ordering. Categorical alert pipelines that assign each flow an attack label benefit more from the MLP.

The cross-model SMOTE comparison produced a noteworthy negative result. SMOTE boosted Random Forest and XGBoost consistently (RF: 0.487 $\rightarrow$ 0.527; XGB: 0.557 $\rightarrow$ 0.631), but it degraded the tuned MLP from 0.636 to 0.578. The MLP baseline already captures U2R reasonably well (F1 = 0.547) owing to its ability to learn nonlinear class boundaries from limited data. SMOTE appears to distort those boundaries by flooding the minority manifold with synthetic points that lie along interpolation lines rather than along the true data surface. This observation aligns with the cautions raised by Carvalho et al. [18] regarding synthetic-point quality on one-hot features. The practical takeaway is that SMOTE should not be applied blindly; its benefit depends on the learner.

Feature selection flattens between $k = 30$ and $k = 50$ and the full 122-feature model actually underperforms the 30-feature one on macro-F1 (0.490 vs. 0.496). Tail features contribute near-zero impurity reduction and appear to inject noise. For resource-constrained IDS deployments — edge switches, IoT gateways, vehicular nodes [16] — trimming to 30 features cuts feature-extraction and inference cost by roughly 75 % with no accuracy penalty.

SHAP attributions map directly onto known attack signatures. `src_bytes` — the dominant feature — is the volumetric footprint of bandwidth-exhaustion DoS. Connection-count features capture the burst patterns of floods and port sweeps. `service_private` and `protocol_type_icmp` flag traffic classes that rarely appear in legitimate enterprise sessions. These explanations can seed lightweight pre-filter rules in a layered architecture: a simple threshold on `src_bytes` alone would intercept a large fraction of DoS traffic before it reaches the classifier.

Limitations. NSL-KDD was built from traffic captured in 1998–1999 and repackaged in 2009 [1]. Modern networks carry encrypted TLS 1.3 sessions, HTTP/2 multiplexed

streams, and cloud-native east–west traffic that differ structurally from the KDD feature set. Replication on a contemporary corpus such as Edge-IIoTset (2022) or CICIOT2023, or on the ToN-IoT dataset used by Gad et al. [16], is needed before these findings can be generalised. U2R’s 52 training samples confine SMOTE to a tiny convex hull. The SHAP analysis was limited to binary detection; multi-class attributions could reveal class-specific feature interactions.

Conclusion

Five classifiers were benchmarked on NSL-KDD under binary and five-class labelling. XGBoost produced the strongest binary AUC (0.969, 95 % CI [0.967, 0.971]). The tuned MLP (256–128) achieved the best five-class macro-F1 (0.636), a 3.4 % relative improvement over the untuned baseline (0.615). No single model dominated both tasks.

Cross-model SMOTE testing revealed learner-dependent effects: SMOTE raised XGBoost macro-F1 from 0.557 to 0.631 and U2R F1 from 0.182 to 0.438, but it degraded the MLP from 0.636 to 0.578. The result underscores that oversampling is not universally beneficial and should be validated per classifier.

Thirty features matched or exceeded full-set macro-F1 while cutting wall-time by 26.6 %. The 122-feature model scored lower, providing evidence of noise injection from uninformative tail features.

SHAP attributed the largest predictive mass to `src_bytes`, `dst_host_srv_count`, and `count` — volumetric and connection-rate features that correspond to DoS flood and scan signatures.

Accuracy alone misrepresented classifier quality by 13–17 percentage points relative to macro-F1 in the five-class setting.

Follow-on work should replicate this pipeline on Edge-IIoTset or CICIOT2023, investigate semi-supervised zero-day detection [17], test GAN-based oversampling, and explore federated IDS training across network segments.

References

- [1] Goldschmidt, P., & Chudá, D. (2025). Network intrusion datasets: A survey, limitations, and recommendations. *Computers & Security*. <https://doi.org/10.1016/j.cose.2025.104510>
- [2] Ali, M. L., Thakur, K., Schmeelk, S., Debello, J., & Dragos, D. (2025). Deep learning vs. machine learning for intrusion detection in computer networks. *Applied Sciences*, 15(4), 1903. <https://doi.org/10.3390/app15041903>
- [3] Maseer, Z. K., Kadhim, Q. K., Al-Bander, B., Yusof, R., & Saif, A. (2024). Meta-analysis and systematic review for anomaly network intrusion detection systems. *IET Networks*, 13(5–6), 339–376. <https://doi.org/10.1049/ntw2.12128>
- [4] Mutembei, L. L., Senekane, M. C., & van Zyl, T. (2025). Deep learning-based network intrusion detection systems: A systematic literature review. In *SACAIR 2024* (CCIS, Vol. 2326, pp. 207–234). Springer. https://doi.org/10.1007/978-3-031-78255-8_13

- [5] Liu, L., Wang, P., Lin, J., & Liu, L. (2021). Intrusion detection of imbalanced network traffic based on machine learning and deep learning. *IEEE Access*, 9, 7550–7563. <https://doi.org/10.1109/ACCESS.2020.3048198>
- [6] Zhang, Y., Muniyandi, R. C., & Qamar, F. (2025). A review of deep learning applications in IDS: Overcoming challenges in feature extraction and data imbalance. *Applied Sciences*, 15(3), 1552. <https://doi.org/10.3390/app15031552>
- [7] Walling, S., & Lodh, N. (2024). Network intrusion detection system for IoT security using ML and statistical based hybrid feature selection. *Security and Privacy*, 7(6), e429. <https://doi.org/10.1002/spy2.429>
- [8] Ren, K., Zeng, Y., Cao, Z., & Zhang, Y. (2022). ID-RDRL: A deep reinforcement learning-based feature selection intrusion detection model. *Scientific Reports*, 12, 15594. <https://doi.org/10.1038/s41598-022-19366-3>
- [9] Gaspar, D., Silva, P., & Silva, C. (2024). Explainable AI for intrusion detection systems: LIME and SHAP applicability on multi-layer perceptron. *IEEE Access*, 12, 30164–30175. <https://doi.org/10.1109/ACCESS.2024.3368377>
- [10] Adadi, A., & Berrada, M. (2018). Peeking inside the black box: A survey on explainable AI. *IEEE Access*, 6, 52138–52160. <https://doi.org/10.1109/ACCESS.2018.2870052>
- [11] Chakrawarti, A., & Shrivastava, S. S. (2024). Intrusion classification and detection system using ML models on NSL-KDD dataset. In *Proc. Fifth Int. Conf. Computer and Communication Technologies* (pp. 73–84). Springer. https://doi.org/10.1007/978-981-99-9707-7_8
- [12] Almotairi, A., Atawneh, S., Khashan, O. A., & Khafajah, N. M. (2024). Enhancing intrusion detection in IoT networks using machine learning-based feature selection and ensemble models. *Systems Science & Control Engineering*, 12(1). <https://doi.org/10.1080/21642583.2024.2321381>
- [13] Chen, W., Yang, K., Yu, Z., Shi, Y., & Chen, C. L. P. (2024). A survey on imbalanced learning: Latest research, applications and future directions. *Artificial Intelligence Review*, 57, 137. <https://doi.org/10.1007/s10462-024-10759-6>
- [14] Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30* (pp. 4765–4774). NeurIPS. <https://doi.org/10.48550/arXiv.1705.07874>
- [15] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proc. 22nd ACM SIGKDD* (pp. 785–794). <https://doi.org/10.1145/2939672.2939785>
- [16] Gad, A. R., Nashat, A. A., & Barkat, T. M. (2021). IDS using ML for vehicular ad hoc networks based on ToN-IoT dataset. *IEEE Access*, 9, 142206–142217. <https://doi.org/10.1109/ACCESS.2021.3120626>
- [17] Mbona, I., & Eloff, J. H. P. (2022). Detecting zero-day intrusion attacks using semi-supervised machine learning approaches. *IEEE Access*, 10, 69822–69838. <https://doi.org/10.1109/ACCESS.2022.3187116>

[18] Carvalho, M., Pinho, A. J., & Brás, S. (2025). Resampling approaches to handle class imbalance: A review from a data perspective. *Journal of Big Data*, 12(1). <https://doi.org/10.1186/s40537-025-01119-4>

[19] Alegavi, S., Nemade, B., & Bharadi, V. (2023). A comprehensive review: SMOTE-based oversampling methods for imbalanced classification. *International Journal of Intelligent Systems and Applications in Engineering*, 11(9s), 790–803. Retrieved from <https://ijisae.org/index.php/IJISAE/article/view/3268>

[20] Niazkar, M., Menapace, A., Brentan, B., Piraei, R., Jimenez, D., & Righetti, M. (2024). Applications of XGBoost in water resources engineering: A systematic literature review. *Environmental Modelling & Software*, 174, 105971. <https://doi.org/10.1016/j.envsoft.2024.105971>