http://dx.doi.org/10.21622/ACE.2026.06.2.2333

A COMPREHENSIVE EVALUATION OF IDS DATASETS AND MODELS WITH A DERIVED FRAMEWORK FOR INFORMED DATASET SELECTION: IDS-RECOMM

Ifrah Sanober and Roohie Naaz Mir

National Institute of Technology Srinagar, Department of Computer Science and Engineering, Hazratbal, Srinagar, Jammu and Kashmir, India

Emails: {ifrah_2022phacse007@nitsri.ac.in, Roohie Naaz Mir: naaz310@nitsri.ac.in}

Received 23 July 2026 - Accepted 09 September 2026 - Published 23 September 2026

ABSTRACT

Benchmark intrusion detection datasets suffer from class imbalance, outdated attack patterns and feature redundancy, yet their influence on model performance is underexplored, particularly for recent datasets such as HIKARI-2021. This study measures how dataset choice affects detection quality and computational cost across machine learning and deep learning models. Ten models, five classical and five deep, were evaluated on NSL-KDD, UNSW-NB15, CIC-IDS2017 and HIKARI-2021 under six configurations per dataset combining two balancing settings with three feature subsets, each repeated over five random seeds for 1,500 runs. Every transform, including recursive feature elimination, was fitted on training data alone. Analysis used repeated-measures ANOVA with effect sizes and bootstrap confidence intervals. Dataset choice explains 9.5% of the variance in F1-score for classical models and 94.3% for deep models. Random Forest attains the highest F1-score on all four datasets. HIKARI-2021 ranks first on accuracy and last on F1-score, a divergence of 21.1 percentage points, because its effective imbalance after deduplication is 34.8:1 rather than 13.7:1. Removing network identifiers changes F1-score by under one percentage point for eight of ten models. Findings are bounded by offline binary classification, fixed hyperparameters and CPU-only measurement, and do not address cross-dataset transfer or multi-class detection. Practitioners can match dataset, model and feature configurations to deployment constraints using IDS-ReComm, a conceptual rule-based scheme derived from these results. This is among the first studies to benchmark HIKARI-2021 against classic IDS datasets under a leakage-free protocol with repeated seeds and a controlled identifier-removal ablation.

Keywords: (Class imbalance; computational efficiency; deep learning; effect size; evaluation metrics; feature selection; HIKARI-2021; intrusion detection; machine learning; Precision-Recall AUC)

1. INTRODUCTION

Intrusion Detection Systems (IDS) play a fundamental role in today's network security environment. Conventionally, signature-based and anomaly-based approaches to intrusion detection are continuously beleaguered with some constraints such as scalability, false alarm rate and incapability of efficiently detecting new attack patterns [1]. Consequently, new research approaches employing ML-based IDS as well as deep learning-based IDS are gaining traction in this regard.

ML approaches are capable of providing high detection accuracy; however, they are heavily dependent on manual feature extraction, training on labelled data, and accuracy may decrease in case of raw, high-dimensional traffic [2], [3]. DL models have arisen from this need for systems that remove dependence on manually extracted features and could effectively learn from the raw data directly [4], [5].

http://dx.doi.org/10.21622/ACE.2026.06.2.2333

The effectiveness of both ML- and DL-based IDS depends fundamentally on the quality of the training dataset. In 2023, attacks targeting Industrial Control Systems associated with critical infrastructure exposed how ML-based IDS trained on outdated datasets can fail to capture current attack patterns [6], [7], showing the importance of dataset selection to effective threat mitigation.

1.1. MOTIVATION AND RESEARCH GAPS

Selecting a suitable dataset is essential for developing IDS models that generalise to current cyber threats [8]. Despite extensive IDS research, two gaps motivate this study: first, there is limited evidence on how modern datasets reflecting encrypted and synthetic attack traffic affect ML and DL performance; second, very few studies compare traditional and deep learning models across multiple datasets including the recently developed HIKARI-2021 dataset [9].

To address these gaps, this study evaluates the benchmark datasets NSL-KDD [10], UNSW-NB15 [11], and CIC-IDS2017 [12], together with HIKARI-2021 [9], using ten ML and DL models. Rather than evaluating models in isolation, the study centres on how dataset properties influence detection accuracy, feature-subset efficiency, and computational cost, using statistical validation to support practical dataset selection guidance.

The scope is deliberately limited to offline, binary classification with fixed hyperparameters, to isolate dataset effects from confounding tuning decisions. All timings were measured on CPU with GPU acceleration disabled, so that classical and deep models are compared on identical terms. This simplifies deployment complexity and may under-represent DL upper-bound performance, but supports attributing performance volatility to dataset properties. Conclusions therefore describe relative performance trends under controlled conditions rather than asserting dataset superiority in real-time or multi-class environments.

Beyond the empirical evaluation, this study introduces a conceptual, rule-based decision-support scheme, IDS-ReComm, which organises the empirical findings into a structured set of selection rules to help practitioners choose a dataset-model-feature configuration suited to specific deployment requirements.

1.2. CONTRIBUTIONS

- A controlled comparison of five machine learning and five deep learning models across four benchmark IDS datasets under six configurations each, repeated over five random seeds, for 1,500 experimental runs.
- A leakage-free evaluation pipeline in which feature selection, encoding, correlation filtering, and standardization are fitted on training data alone, with identical metric definitions applied to both model families.
- Statistical analysis using a repeated-measures design that respects the pairing of models across datasets, reporting effect sizes and bootstrap confidence intervals alongside significance tests.
- Evidence that HIKARI-2021's apparent advantage is metric-dependent, with a quantification of the duplicate structure and effective class imbalance that produces it.
- An identifier-removal ablation on an identical row set, which measures rather than assumes the contribution of network identifier fields.
- IDS-ReComm, a conceptual rule-based scheme that translates the empirical findings into structured deployment guidance.

2. RELATED WORK

The work in this study was completed through systematic literature reviews in Scopus, IEEE

http://dx.doi.org/10.21622/ACE.2026.06.2.2333

Xplore, and Google Scholar (2019-2024) using keywords "IDS dataset evaluation", "intrusion detection benchmarking" and "deep learning intrusion detection". This review favoured literature focused on methodological robustness, dataset applicability, and reproducibility. All related studies have been categorised into three major areas: IDS benchmarking approaches; dataset issues and critique; and comparison of ML/DL approaches.

2.1. IDS BENCHMARKING STUDIES

In [13], the researchers have used a deep learning-based NIDS on NSL-KDD, resulting in a recall of 79.47% and an F1-score of 76.47%. In [14], Kumar et al. used a classification-based model on UNSW-NB15 and developed a testbed dataset, RTNITP18, designed for real time tests. Choudhary and Kesswani [15] made use of deep neural networks on KDD-Cup'99, NSL-KDD, and UNSW-NB15, where accuracy greater than 90% was observed for all three datasets. Ghurab et al. [16] carried out benchmarks on eight IDS datasets and advised current datasets such as CIC-IDS2017 and CSE-CIC-IDS2018 since they represent today's threats.

2.2. DATASET FLAWS AND CRITICISM

Another shortcoming of the most popular IDS datasets is repetition, incorrect labelling and implausible attack cases. Engelen et al. [17] discovered imperfections in CIC-IDS2017, in which the implementation of DoS attacks was poor, while many feature extraction operations were inaccurate, so they put forward a revised CIC-IDS2017. Liu et al. [18] discovered more mislabelled records in CIC-IDS2017 and CSE-CIC-IDS2018; the rate of corrupted data in some categories exceeded 75%, indicating potential instability issues of DL training on such datasets. More issues, such as packet reordering and non-consecutive timestamps, were reported in [19] as drawbacks of CIC-IDS2017.

HIKARI-2021 was similarly reviewed. Fernandes et al. [20] found two duplicate columns and demonstrated that removal of network identifiers (IP addresses) degraded performance by approximately 2%, a result returned to directly in Section 4.5, where the present study tests it under a controlled ablation. Kwon et al. [21] identified significant class imbalance in HIKARI-2021, negatively impacting the model, and found that the addition of sampling boosted detection rate to 99.64%. Data-overlap issues were addressed by Zoghi and Serpen [22] with a novel method of visualising data in the UNSW-NB15 dataset. Layeghy et al. [23] analysed UNSW-NB15, CIC-IDS2017 and TON-IoT against actual traffic, concluding that while the synthetic datasets reflect the same general structure as actual data, they do not contain variety in background traffic.

2.3. COMPARATIVE ML/DL STUDIES AND RESEARCH GAP

Catillo et al. [24] trained IDS models on CIC-IDS2017 and USB-IDS-1 [25], obtaining strong recall (0.97-0.99) on CIC-IDS2017 but struggling with slow attacks on USB-IDS-1, illustrating a common cross-dataset generalisation problem. Jha et al. [26] argued for scalable, dynamically generated IDS datasets to address privacy and repeatability constraints.

Prior work has largely focused on benchmarking individual datasets, identifying dataset flaws, or comparing ML/DL models on select datasets, but no study has methodically compared HIKARI-2021 across multiple ML and DL models alongside classic IDS datasets, nor examined computational trade-offs related to dataset choice under resource constraints. This study addresses that gap by jointly analysing dataset influence on accuracy, feature selection, and computational efficiency to inform dataset selection and model deployment.

3. METHODS

3.1. DATASETS

Four representative benchmarks covering various generations of network traffic and attack styles were used: NSL-KDD, UNSW-NB15, CIC-IDS2017, and HIKARI-2021. Main features of these datasets are shown in Table 1.

http://dx.doi.org/10.21622/ACE.2026.06.2.2333

NSL-KDD [10] enhances KDD-Cup'99 [27] by removing redundant records and handling class imbalance. The NSL-KDD dataset consists of 41 features that represent four major types of attacks (DoS, Probe, R2L and U2R). However, attack patterns represented in this dataset have become outdated and do not precisely describe current traffic.

UNSW-NB15 [11] was created by the Australian Centre for Cyber Security's Cyber Range Lab, using an IXIA PerfectStorm traffic generator. The UNSW-NB15 dataset has 49 features which combine packet and flow-based features; there are nine classes of attacks: Backdoors, Analysis, DoS, Fuzzers, Generic, Exploits, Reconnaissance, Shellcode and Worms. The dataset consists of 175,341 records in the training set and 82,332 records in the test set. UNSW-NB15 represents a varied and reasonably up to date attack set, but also contains several redundant and highly correlated features.

CIC-IDS2017 [12] was generated by the Canadian Institute for Cybersecurity based on real network data collected over five days (3-7 July 2017). This dataset contains 80 features generated using CICFlowMeter across classes including botnet, DDoS, portscan, DoS, web attack and infiltration, and suffers from high class imbalance due to the presence of many missing values, more than 308,000 duplicates and errors in labels as well as feature extraction [17], [18], [19], requiring careful preprocessing.

HIKARI-2021 [9] was proposed as a contemporary successor to CIC-IDS2017, with 86 features [28] spanning six traffic classes (Benign, Background, Probing, Bruteforce-XML, Bruteforce, XMRIGCC CryptoMiner). As distributed, the file contains 555,278 flows of which 37,696 are labelled as attacks, an imbalance ratio of 13.7:1. After removal of row-index and identifier columns and deduplication on the feature set, 491,248 flows remain and the attack class falls to 14,050, giving an effective imbalance of 34.0:1. In other words, 62.7% of the attack flows in the distributed file are exact duplicates of other flows. Section 3.4 explains why this duplication is not detectable while the per-flow identifier is retained, and Section 4.5 reports its consequences for evaluation. Prior work has also reported feature redundancy and an accuracy drop of up to 2% when network identifiers are removed [20], [21], a claim tested directly in Section 4.5.

Table 1: Summary of Datasets Used in this Study

Feature	NSL-KDD	UNSW-NB15	CIC-IDS2017	HIKARI-2021
Year / Creator	2009, Tavallaee et al.	2015, Moustafa and Slay	2017, Sharafaldin et al.	2021, Ferriyan et al.
Features (raw)	41	49	80	86
Records after cleaning	147,788	153,684	2,520,101	491,248
Attack rate after cleaning	47.85%	44.38%	16.89%	2.86%
Effective imbalance	1.1:1	1.3:1	4.9:1	34.0:1
Feature-identical duplicates	727	103,989	307,775	64,030
Of which contradictory labels	98	414	697	34,246
Access	Public [29]	Public [30]	Public [31]	Public [28]

Record counts in Table 1 differ from those commonly reported because deduplication is performed on the feature set after removal of row identifiers. Section 3.4 explains the procedure.

3.2. COMPUTATIONAL ENVIRONMENT AND EXPERIMENTAL DESIGN

The experiments reported here were executed on a Linux high-performance computing cluster at the National Institute of Technology Srinagar, using the SLURM workload manager. The full design comprises 1,500 runs across five datasets, six configurations, ten models and five random seeds, a scale that requires distributed batch execution. All runs were executed on a single compute node, so timings are internally comparable throughout. The node provides 64 physical CPU cores on a single socket and 503 GB of RAM, of which each job was allocated 16 cores and 128 GB. The software stack comprised Python with

http://dx.doi.org/10.21622/ACE.2026.06.2.2333

Keras, scikit-learn, TensorFlow, NumPy, Pandas, Matplotlib and Imbalanced-learn. Although GPU accelerators were present on the node, they were explicitly disabled for all runs, so that classical and deep models are trained and timed on identical CPU hardware and the computational comparison in Sections 4.4 and 5.2 is made on equal terms.

All experiments used binary classification (benign against malicious) to ensure methodological consistency across datasets with different attack taxonomies. Binary modelling is widely used in early-stage IDS evaluation to assess dataset separability and sensitivity to feature selection; multi-class analysis is beyond the present scope.

Each dataset was evaluated under two balancing settings (imbalanced; SMOTE-balanced) and three feature settings, yielding six configurations per dataset. The three feature settings are: all features surviving the correlation filter; the top 20 selected by recursive feature elimination with a Random Forest estimator; and the top 10 selected from those 20. The selections are nested, so the three settings form an ordered sequence rather than three unrelated subsets. Each configuration was repeated with five random seeds, giving 1,500 runs in total across the four datasets and the identifier-removal variant described in Section 4.5 (Table 2, Fig. 1).

Table 2: Dataset Configurations

Setup	Configuration
Imbalanced	Original class distribution, no balancing.
Balanced	Class imbalance corrected using SMOTE [32], applied to the training split only.
Feature settings	(1) All features surviving the correlation filter; (2) top 20 by RFE; (3) top 10, selected from the 20.
Repetition	Five random seeds (42 to 46) per configuration.

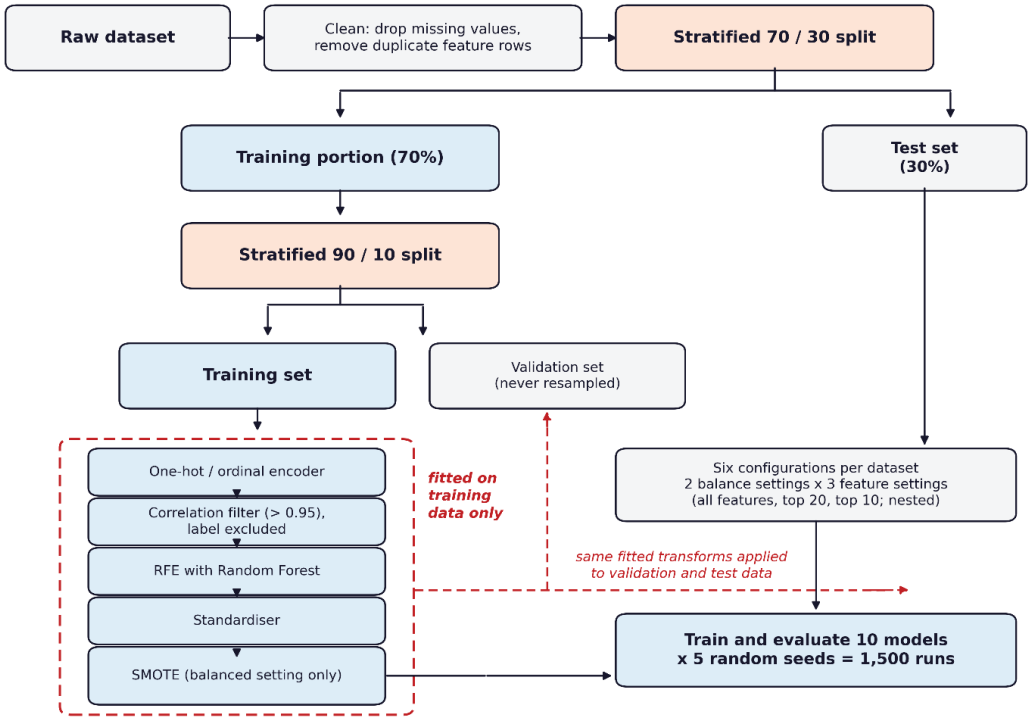


Figure 1: Experimental pipeline. Every transform is fitted on the training split alone and applied to validation and test data.

3.3. MODELS AND HYPERPARAMETERS

Five ML models (Decision Tree, Random Forest, Gaussian Naive Bayes, K-Nearest Neighbours, Support Vector Machine) and five DL models (Gated Recurrent Unit, Long Short-Term Memory,

http://dx.doi.org/10.21622/ACE.2026.06.2.2333

Multi-layer Perceptron, SimpleRNN, Convolutional Neural Network) were evaluated (Table 3). Fixed hyperparameters and architectures were used across all datasets to isolate dataset-model sensitivity from model-specific tuning effects, rather than to achieve absolute optimum performance for every pairing. DT, RF, and GNB used scikit-learn defaults; KNN used k = 3; SVM used an RBF kernel with balanced class weights, wrapped in a bagging classifier of ten base estimators. GRU and LSTM used two recurrent layers (20 units, dropout 0.2) with a sigmoid output; MLP used two dense layers of 20 ReLU units; SimpleRNN used a 50-unit recurrent layer followed by a 50-unit ReLU dense layer; CNN used two convolutional layers (32 and 64 filters) with max pooling, a 128-unit dense layer and 0.5 dropout. All DL models used the Adam optimiser and binary cross-entropy loss (Table 4).

Table 3: Models Employed in this Study

Type	Models
ML	Decision Tree (DT), Random Forest (RF), Gaussian Naive Bayes (GNB), K-Nearest Neighbours (KNN), Support Vector Machine (SVM)
DL	Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), Multi-layer Perceptron (MLP), SimpleRNN, Convolutional Neural Network (CNN)

Table 4: Hyperparameters

Model	Key parameters	Tuning
DT / RF / GNB	scikit-learn defaults	None
KNN	n_neighbors = 3	Fixed
SVM	RBF kernel, balanced class weights, 10 base estimators, training capped at 100,000 stratified instances	Fixed
GRU / LSTM	2 layers (20 units), dropout 0.2, sigmoid output	Fixed arch.
SimpleRNN	Dense(50, ReLU) after 50 recurrent units, sigmoid output	Fixed arch.
MLP	Dense(20)-Dense(20)-Sigmoid	Fixed arch.
CNN	Conv(32)-MaxPool-Conv(64)-Dense(128)-Dropout(0.5)-Sigmoid	Fixed arch.
All DL	Adam optimiser, binary cross-entropy, batch 2000, max 200 epochs, early stopping (patience 10) on validation loss	Fixed

Recurrent and convolutional architectures are not obviously suited to tabular flow records, which have no temporal or spatial ordering among their columns. They are nonetheless applied to such records routinely in the IDS literature, usually without justification. One purpose of a controlled comparison is to test whether that practice is warranted. The multi-layer perceptron included here makes no sequential assumption and serves as the reference point for that test; Section 4.2 reports the outcome.

Support vector machine training was capped at a stratified sample of 100,000 instances, applied uniformly to all four datasets so that no dataset receives special treatment. Within the bagging ensemble of ten base estimators this corresponds to 10,000 instances per estimator. The cap was necessary because an uncapped RBF-kernel implementation on CIC-IDS2017 required in excess of 30 hours for a single configuration in preliminary runs, which is not compatible with a five-seed design. Reported SVM results and timings therefore understate an uncapped implementation.

Epoch counts are not set per dataset. All deep models use a single stopping rule: a maximum of 200 epochs at batch size 2000, with early stopping on validation loss after ten epochs without improvement and restoration of the best weights. The validation split is drawn from the training data before any resampling, so it remains real and unbalanced. Under this rule the number of epochs actually used varies by dataset: the mean is 156.8 epochs on NSL-KDD and 167.1 on UNSW-NB15 against 99.8 on CIC-IDS2017 and 80.1 on HIKARI-2021. The larger datasets converge in fewer epochs.

http://dx.doi.org/10.21622/ACE.2026.06.2.2333

3.4. DATA PREPROCESSING

Preprocessing follows a fixed order in which no step downstream of the train-test split observes a test row:

- Cleaning. Rows with missing or non-finite values are removed. Duplicate rows are then removed on the feature set, retaining the first occurrence.
- Splitting. A stratified 70/30 train-test split is drawn, followed by a stratified 90/10 split of the training portion into training and validation data. The validation split is taken before any resampling, so it remains real and unbalanced.
- Encoding. A one-hot encoder is fitted on the training rows and applied to validation and test data, with categories unseen in training ignored. Categorical columns with more than 200 distinct values are ordinal-encoded instead, since one-hot encoding of the HIKARI-2021 address fields would produce approximately 10,900 dense columns.
- Correlation filtering. Pairwise absolute correlations above 0.95 are computed on the training features only, with the label excluded, and redundant columns are dropped.
- Feature selection. Recursive feature elimination with a Random Forest estimator is fitted on the training rows only, and the resulting selection is applied to validation and test data.
- Scaling. A standardiser is fitted on the training rows and applied to validation and test data by transformation, not refitting.
- Resampling. SMOTE is applied to the training split alone. Validation and test data are left untouched.

Deduplication on the feature set rather than the full row has a consequence worth stating. Where duplicate detection is performed while per-row identifiers are still present, those identifiers are unique by construction and no duplicate can be found. With identifiers removed, UNSW-NB15 loses 40.4% of its rows and HIKARI-2021 11.5% as exact duplicates, as recorded in Table 1.

A further property of HIKARI-2021 emerges from the same check. Of its 64,030 feature-identical flows, 34,246 carry contradictory labels, meaning the same feature vector appears both as benign and as an attack. The corresponding counts for NSL-KDD, UNSW-NB15 and CIC-IDS2017 are 98, 414 and 697. Removing every member of an ambiguous group would leave HIKARI-2021 with 3,290 attack flows at an imbalance of 137.9:1, too few to support the present design, so the first occurrence is retained. Because the distributed file is ordered by class, this resolution is order-dependent, a limitation recorded in Section 5.5.

3.5. EVALUATION METRICS AND STATISTICAL VALIDATION

Every model produces predicted labels and predicted probabilities, and all metrics are computed from those arrays by the same library functions, so that classical and deep results are directly comparable. Precision, recall and F1-score are reported for the positive (attack) class, with macro-averaged variants recorded alongside. Accuracy, balanced accuracy, Matthews correlation coefficient, ROC-AUC and Precision-Recall AUC are also reported. PR-AUC is included because it is informative under heavy imbalance in a way that ROC-AUC is not, a point Section 4.5 develops.

The unit of observation for statistical analysis is one combination of model, feature setting and random seed, measured across datasets. These observations are paired rather than independent, because the same ten models are re-measured on every dataset, and treating them as independent inflates the F statistic. Results are therefore reported as repeated-measures ANOVA with the model configuration as subject and dataset as the within-subject factor, alongside the one-way result for comparability.

http://dx.doi.org/10.21622/ACE.2026.06.2.2333

Each F statistic is accompanied by eta-squared and omega-squared. Each mean and each pairwise difference is accompanied by a bootstrap 95% confidence interval from 10,000 resamples, and Holm correction is applied across the pairwise family. Effect sizes are reported because a significant result is not necessarily a meaningful one: deep learning PR-AUC on CIC-IDS2017 exceeds that on NSL-KDD by 0.12 percentage points at p = 0.011, a difference with no operational consequence.

4. RESULTS

All reported values are means over five random seeds. Figures show one standard deviation as error bars.

4.1. ACCURACY

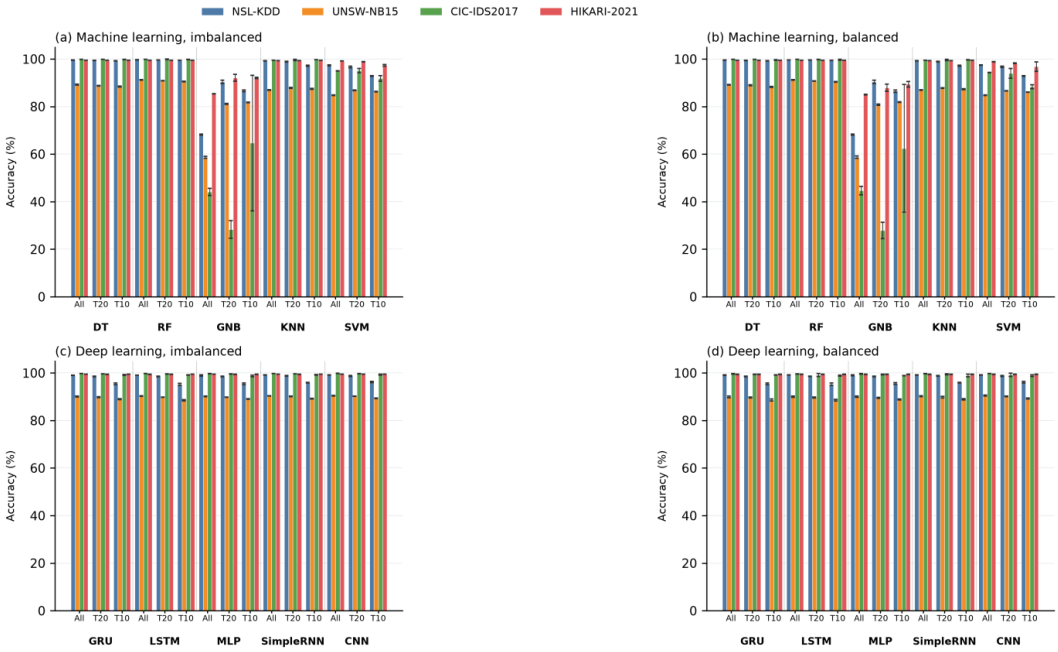


Figure 2: Accuracy across datasets: (a) ML models, imbalanced; (b) ML models, balanced; (c) DL models, imbalanced; (d) DL models, balanced.

Deep learning models reach 99.48% mean accuracy on CIC-IDS2017 and 99.44% on HIKARI-2021, against 97.77% on NSL-KDD and 89.74% on UNSW-NB15. Classical models reach 97.31% on HIKARI-2021, 94.96% on NSL-KDD, 87.81% on CIC-IDS2017 and 85.37% on UNSW-NB15. Balancing changes little: the mean difference in F1-score between the SMOTE-balanced and imbalanced settings is under one percentage point across the grid, which is expected given that NSL-KDD and UNSW-NB15 are close to balanced after deduplication.

Random Forest attains the highest accuracy of any model on all four datasets, at 99.65% on NSL-KDD, 91.23% on UNSW-NB15, 99.93% on CIC-IDS2017 and 99.49% on HIKARI-2021. Gaussian Naive Bayes is the weakest classical model throughout, falling to 44.15% on CIC-IDS2017 with all features (Fig. 2).

4.2. PRECISION, RECALL AND F1-SCORE

Mean F1-score by dataset separates the two families more clearly than accuracy does. For classical models the order is NSL-KDD at 93.55%, CIC-IDS2017 at 84.10%, UNSW-NB15 at 81.44% and HIKARI-2021 at 75.77%. For deep models it is CIC-IDS2017 at 98.45%, NSL-KDD at 97.66%, HIKARI-2021 at 90.93% and UNSW-NB15 at 88.38%.

http://dx.doi.org/10.21622/ACE.2026.06.2.2333

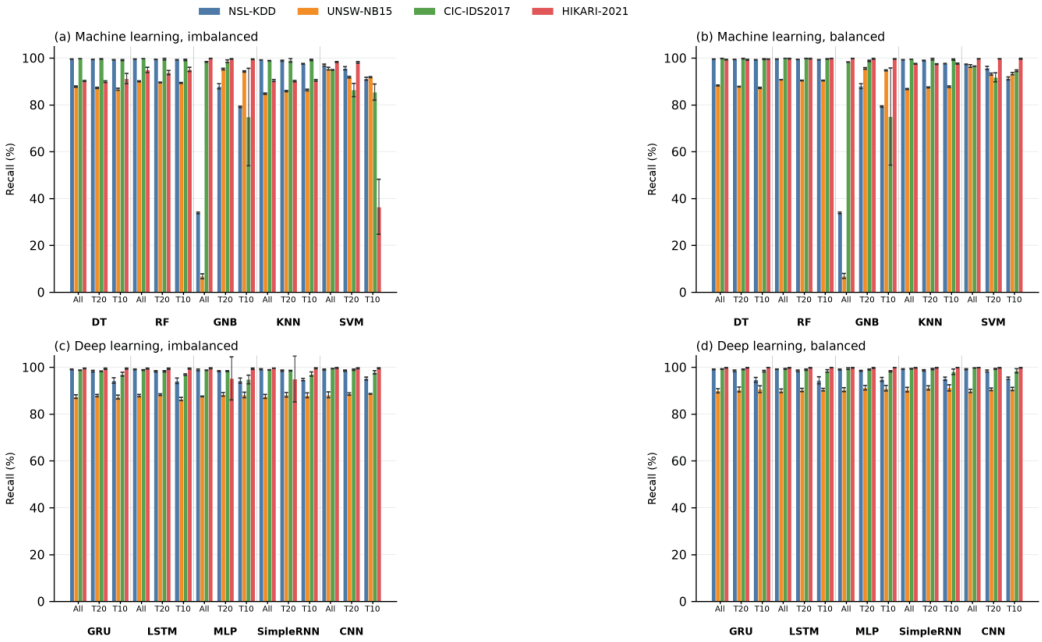


Figure 3: Recall across datasets: (a) ML, imbalanced; (b) ML, balanced; (c) DL, imbalanced; (d) DL, balanced.

The component metrics show where these differences originate. Precision is consistently the stronger of the two for classical models on HIKARI-2021, while recall collapses, which is the signature of a classifier favouring the majority class under heavy imbalance (Figs. 3 and 4). Gaussian Naive Bayes shows the most extreme case, reaching 99.37% precision at 33.93% recall on NSL-KDD with all features.

Family means are misleading if read as a ranking of approaches. The best classical model outperforms the best deep model on every dataset: Random Forest reaches 99.63% against 99.16% for CNN on NSL-KDD, 90.13% against 89.33% on UNSW-NB15, 99.79% against 99.52% on CIC-IDS2017, and 91.84% against 91.33% on HIKARI-2021. The deep learning family mean is higher only because Gaussian Naive Bayes and the support vector machine depress the classical mean. Comparisons between families should therefore be made model by model.

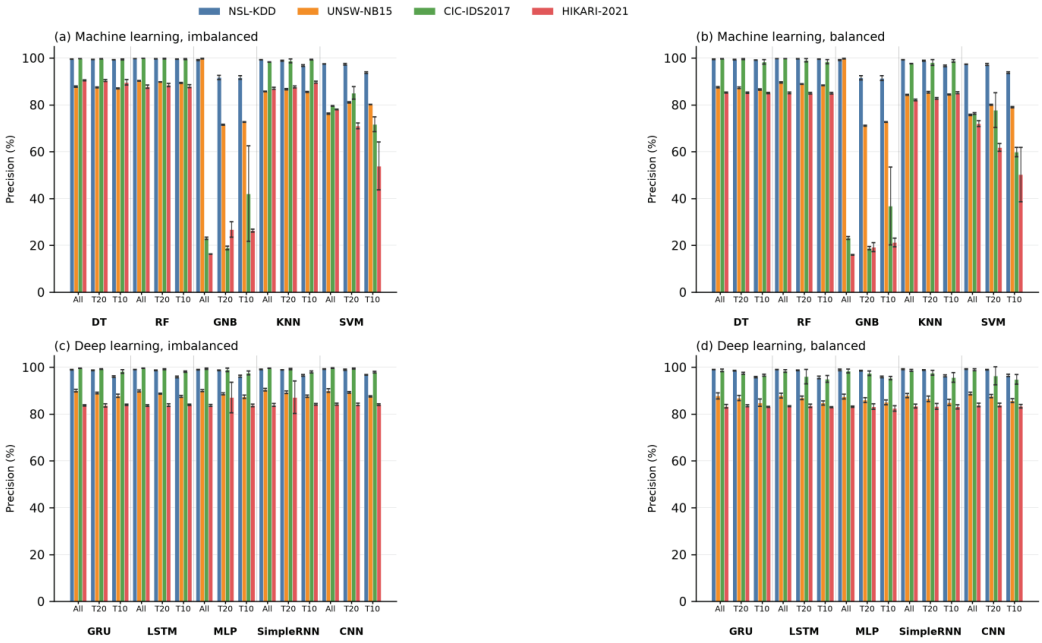


Figure 4: Precision across datasets: (a) ML, imbalanced; (b) ML, balanced; (c) DL, imbalanced; (d) DL, balanced.

http://dx.doi.org/10.21622/ACE.2026.06.2.2333

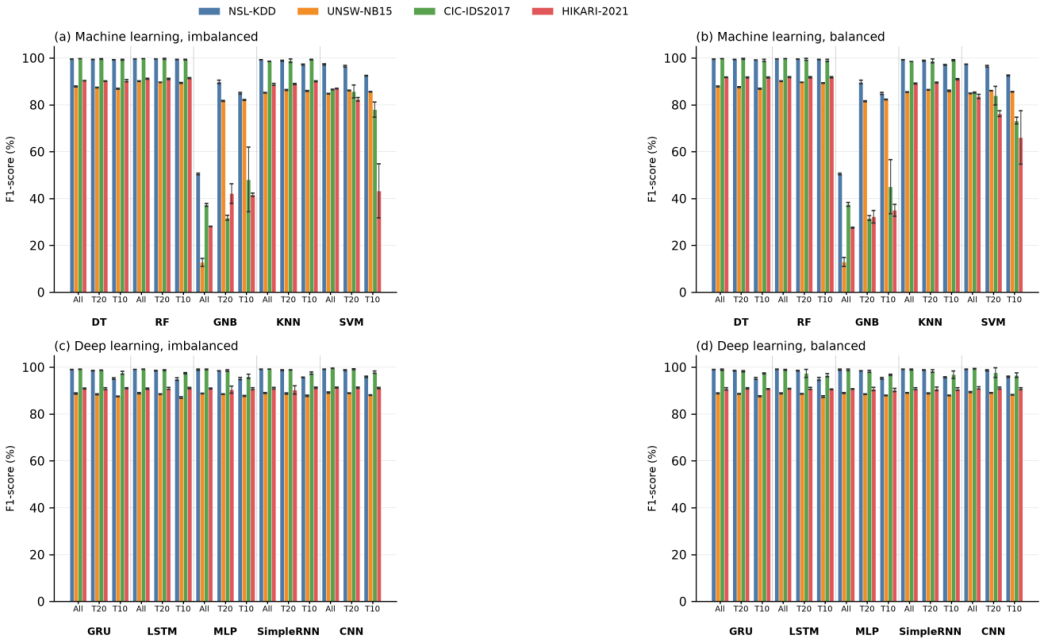


Figure 5: F1-score across datasets: (a) ML, imbalanced; (b) ML, balanced; (c) DL, imbalanced; (d) DL, balanced.

On the question of architecture, the multi-layer perceptron performs within roughly one percentage point of the recurrent models on every dataset while training in a fraction of the time, at 76.04 seconds against 201.19 for GRU and 204.40 for LSTM on CIC-IDS2017. The sequential structure these architectures assume does not appear to recover information from tabular flow records that a plain feed-forward network cannot.

Feature reduction degrades performance monotonically for both families. Mean F1-score falls from 98.9% with all features to 98.3% at twenty features and 95.5% at ten for deep models, and from 96.8% to 96.5% and 94.4% for classical models (Fig. 5).

4.3. THRESHOLD-FREE MEASURES AND STATISTICAL VALIDATION

Dataset choice has a significant effect on F1-score for both families, but the magnitudes differ by an order of magnitude. For classical models, the repeated-measures analysis gives $F(3,222) = 35.85$ at $p = 6.2e\text{-}19$ with an eta-squared of 0.095. For deep models it gives $F(3,222) = 2648.27$ at $p = 1.9e\text{-}173$ with an eta-squared of 0.943. Dataset choice thus accounts for 9.5% of the variance in classical model performance and 94.3% in deep model performance (Table 5).

Table 5: Effect of Dataset Choice on F1-Score by Model Family

Family	One-way F	eta-squared	Repeated-measures F	p
Machine learning	10.31	0.095	F(3,222) = 35.85	6.2e-19
Deep learning	1621.41	0.943	F(3,222) = 2648.27	1.9e-173

The dataset effect on classical models is statistically significant rather than absent; what distinguishes the families is the size of the effect, not its presence. The finding is better supported by effect size than by a significance threshold alone.

http://dx.doi.org/10.21622/ACE.2026.06.2.2333

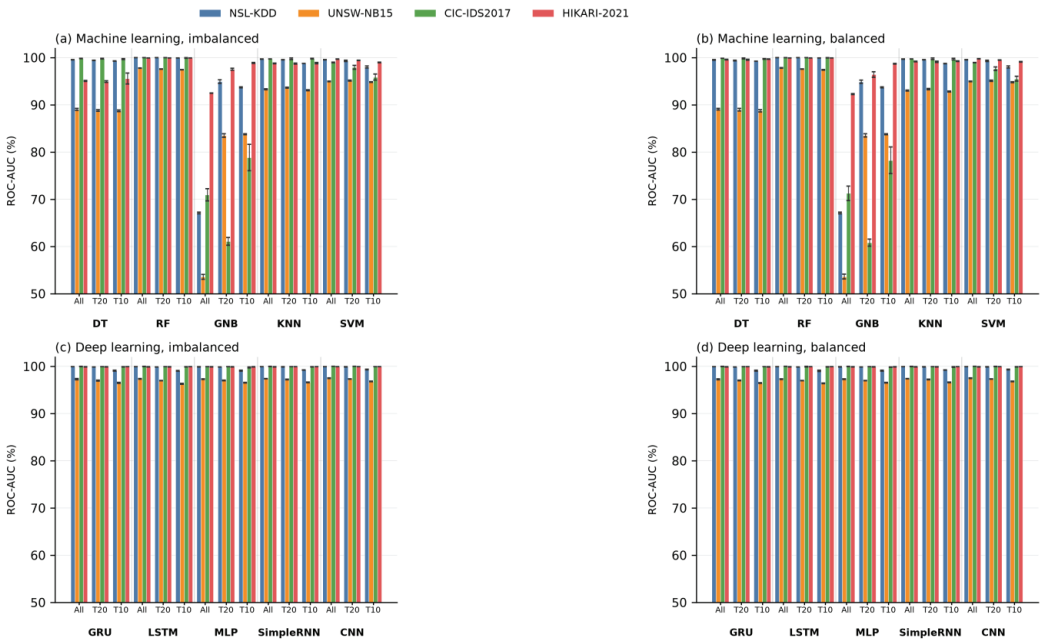


Figure 6: ROC-AUC across datasets: (a) ML, imbalanced; (b) ML, balanced; (c) DL, imbalanced; (d) DL, balanced.

ROC-AUC is uninformative for cross-dataset comparison in this study. All deep learning values fall between 0.9701 and 0.9995, a span of 0.038 (Fig. 6), within which several pairwise differences are significant but negligible: CIC-IDS2017 exceeds NSL-KDD by 0.28 percentage points at p = 1.6e-08. PR-AUC separates the datasets more usefully and is reported in Fig. 7.

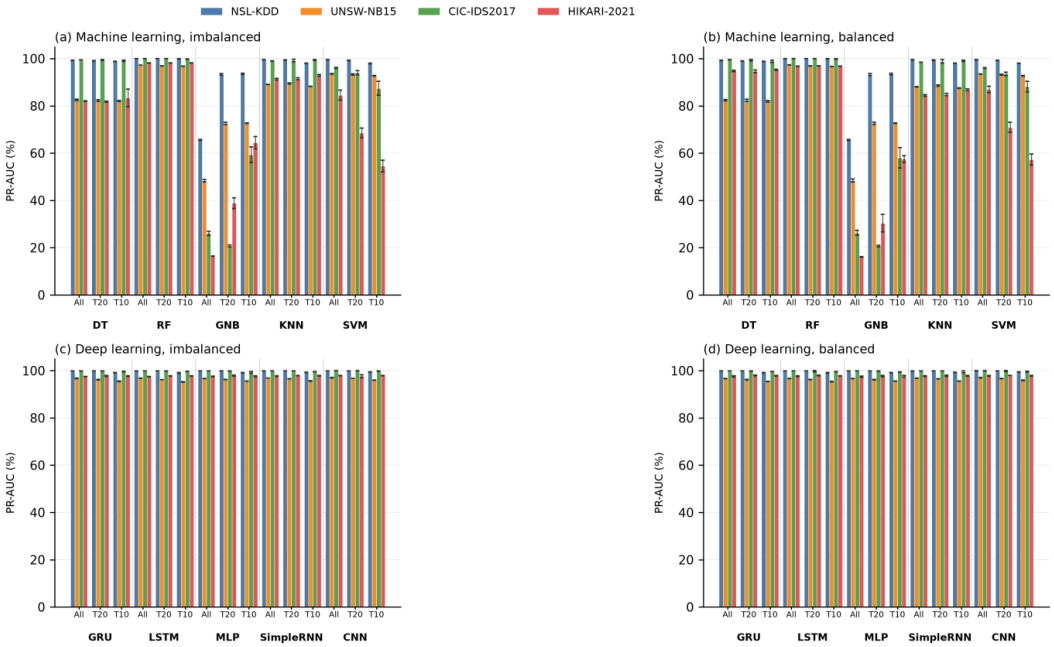


Figure 7: Precision-Recall AUC across datasets: (a) ML, imbalanced; (b) ML, balanced; (c) DL, imbalanced; (d) DL, balanced.

4.4. COMPUTATIONAL COST

Dataset choice significantly affects computational cost for both families. The effect is proportionally larger for deep models, with an eta-squared of 0.636 for inference against 0.188 for classical models, because deep architectures are similar in cost to one another while classical models are not, so model choice dominates dataset choice on the classical side.

http://dx.doi.org/10.21622/ACE.2026.06.2.2333

In absolute terms the largest inference costs are classical. Mean inference time on CIC-IDS2017 is 47.24 seconds for classical models against 0.69 seconds for deep models. The cost is concentrated in two models: K-Nearest Neighbours takes 370.89 seconds and the support vector machine 72.27 seconds on CIC-IDS2017 with all features, both scaling with training-set size at prediction time. Random Forest, which is also the most accurate model on every dataset, requires 0.43 seconds (Figs. 8 and 9).

Training cost runs the other way. Deep models require 257.14 seconds on average on CIC-IDS2017 against 10.62 seconds for classical models. The practical reading is that classical models are cheap to train and can be expensive to query, while deep models are expensive to train and cheap to query (Table 6).

Table 6: Mean Training and Inference Time by Dataset and Model Family (seconds, CPU only)

Family	Measure	NSL-KDD	UNSW-NB15	CIC-IDS2017	HIKARI-2021
Machine learning	Training	1.18	3.20	10.62	1.27
Machine learning	Inference	0.88	1.96	47.24	2.88
Deep learning	Training	28.49	34.04	257.14	44.02
Deep learning	Inference	0.21	0.22	0.69	0.27

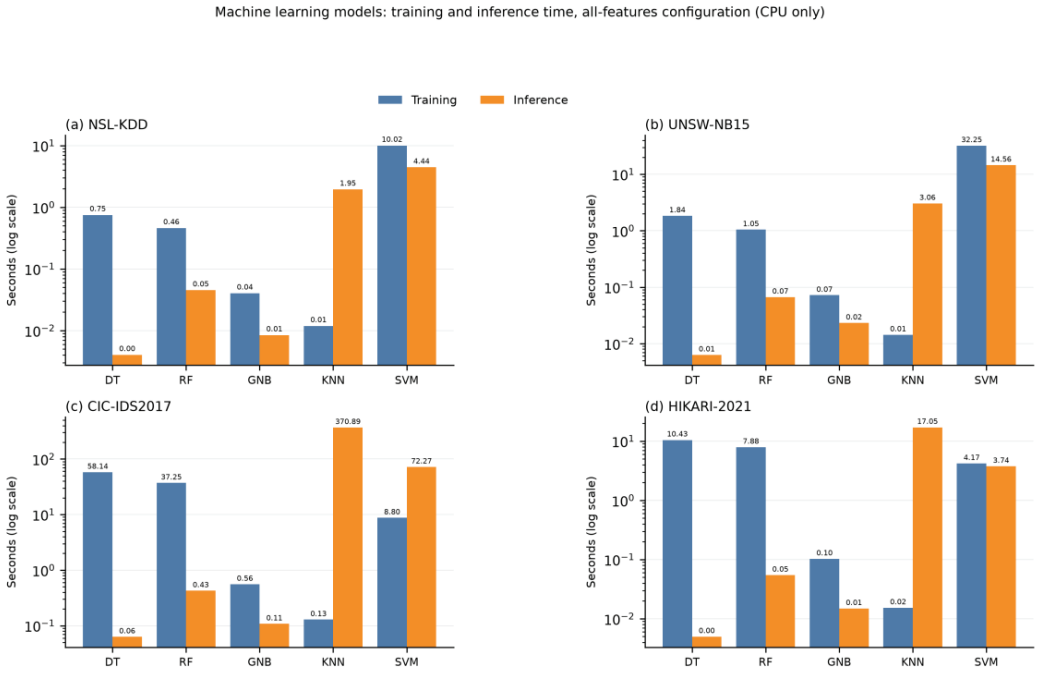


Figure 8: Training and inference time, ML models, all features, CPU only: (a) NSL-KDD; (b) UNSW-NB15; (c) CIC-IDS2017; (d) HIKARI-2021.

http://dx.doi.org/10.21622/ACE.2026.06.2.2333

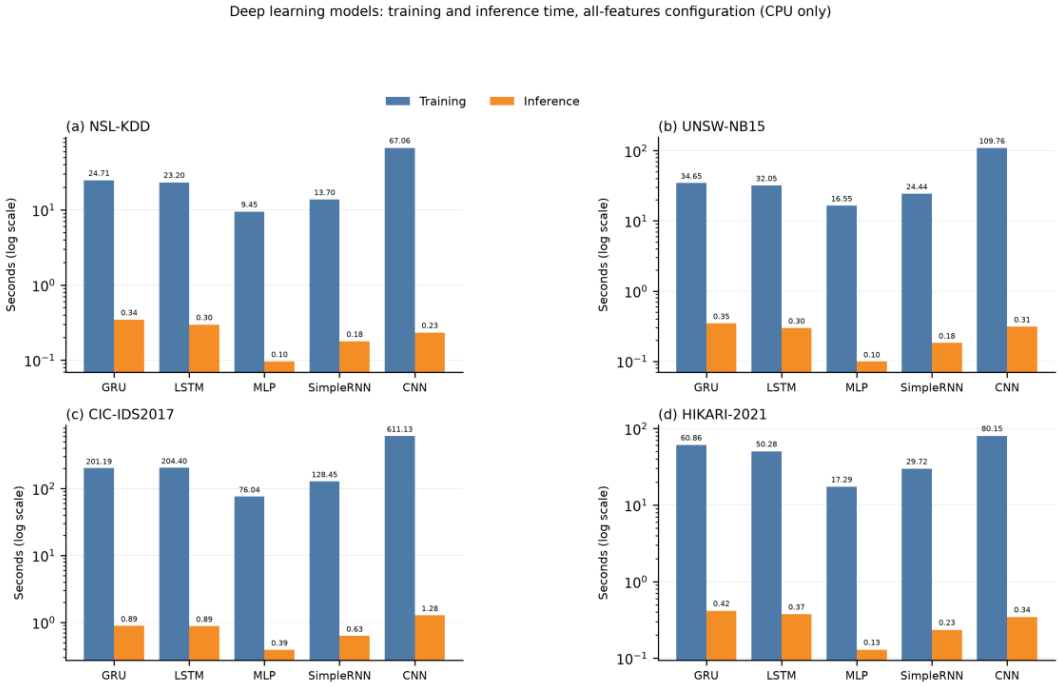


Figure 9: Training and inference time, DL models, all features, CPU only: (a) NSL-KDD; (b) UNSW-NB15; (c) CIC-IDS2017; (d) HIKARI-2021.

4.5. HIKARI-2021: METRIC DIVERGENCE AND THE IDENTIFIER QUESTION

HIKARI-2021 ranks first among the four datasets on accuracy and last or third on F1-score. The gap between its mean accuracy and mean F1-score is 21.1 percentage points for classical models and 8.6 for deep models. The equivalent gaps for the other three datasets are 1.4, 3.8 and 3.9 points for classical models and 0.1, 1.2 and 1.2 for deep models.

The cause is the dataset's effective class imbalance. After correct deduplication the ratio is 34.0:1 rather than the 13.7:1 implied by the distributed file, so a classifier that predicts the majority class attains high accuracy while detecting few attacks. Accuracy conceals this and F1-score and PR-AUC reveal it. A benchmark can therefore appear to be the strongest of a set under one metric and the weakest under another, and the difference is a property of the benchmark rather than of the models evaluated on it (Fig. 10).

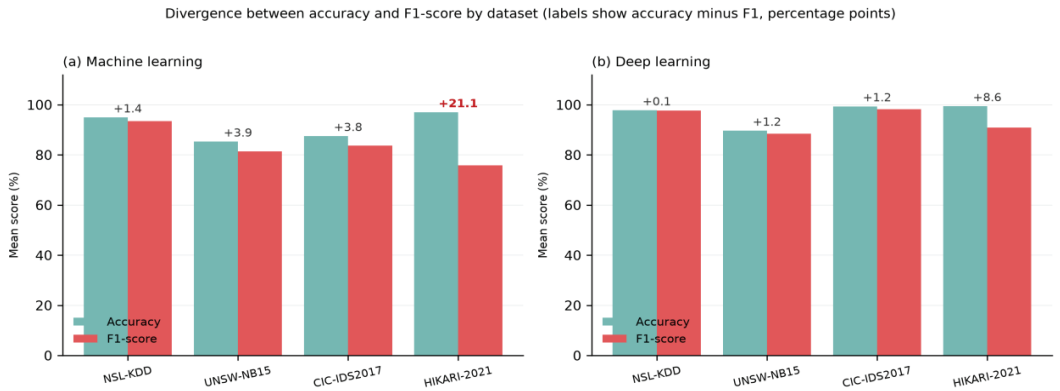


Figure 10: Divergence between accuracy and F1-score by dataset. Labels show accuracy minus F1-score in percentage points.

Fernandes et al. [20] attributed part of HIKARI-2021's separability to network identifier fields. To test this, the dataset was re-run with the four identifier columns (source and destination addresses and ports) removed from an identical row set, so that only the columns differ between conditions. For eight of ten models the change in mean F1-score is below one percentage point, and K-Nearest Neighbours improves slightly (Table 7, Fig. 11).

http://dx.doi.org/10.21622/ACE.2026.06.2.2333

Table 7: Mean F1-Score on HIKARI-2021 With and Without Network Identifier Columns

Model	With identifiers	Identifiers removed	Change (pp)
Random Forest	91.50	91.37	+0.13
K-Nearest Neighbours	89.53	89.72	-0.19
Decision Tree	91.00	90.57	+0.43
CNN	91.13	90.58	+0.55
SimpleRNN	90.81	90.23	+0.58
LSTM	90.86	90.00	+0.86
GRU	90.86	89.98	+0.88
MLP	90.64	89.76	+0.88
Support Vector Machine	73.10	57.10	+16.00
Gaussian Naive Bayes	34.44	15.79	+18.65

The two exceptions are the models least able to handle high-cardinality ordinal features, and both are unstable on the affected configuration: the support vector machine records a standard deviation of 24.01 on a mean of 28.48 in the top-20 setting. Their sensitivity is attributed to encoding rather than to leakage.

A different form of leakage is present in the distributed file. It carries two row-index columns and a per-flow identifier, and it is ordered by class, so a sequential index predicts the label almost perfectly. Those columns are removed in the present study. Their removal, together with correct duplicate detection, accounts for the difference between the results reported here and performance figures reported for this dataset in the prior work.

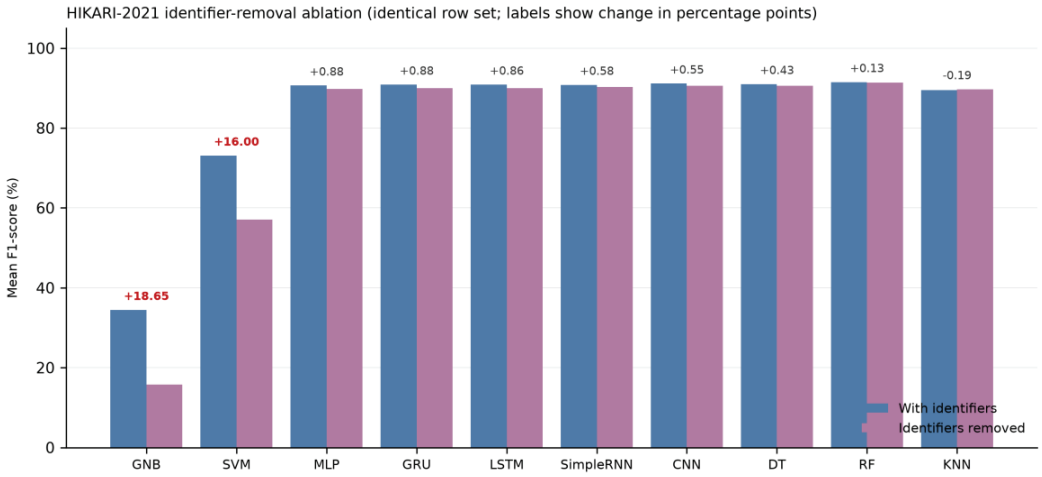


Figure 11: HIKARI-2021 identifier-removal ablation on an identical row set. Labels show the change in mean F1-score in percentage points.

5. DISCUSSION

These results clarify how dataset selection affects IDS model performance. The central pattern is that classical models are markedly less sensitive to dataset choice than deep models, with dataset accounting for 9.5% against 94.3% of variance in F1-score. A practitioner choosing a classical model can expect its behaviour to transfer more predictably across benchmarks than a deep model's.

Dataset recency is not by itself evidence of benchmark quality. HIKARI-2021 is the most recent of the four datasets examined, and its advantage is confined to accuracy, the metric least suited to a benchmark at 34.0:1 imbalance.

A central methodological choice, fixed hyperparameters and architectures across datasets, was intended to attribute performance differences primarily to dataset characteristics rather than model-specific tuning. This increases comparative validity but will not achieve absolute optimum performance for every model-dataset pairing.

http://dx.doi.org/10.21622/ACE.2026.06.2.2333

5.1. RE-EVALUATING TRADITIONAL ML MODELS

The assumption that classical models perform poorly under class imbalance and feature selection is not supported here. Random Forest attains the highest F1-score on all four datasets while requiring 0.43 seconds for inference on the largest, and Decision Tree is second on three of four. K-Nearest Neighbours and the support vector machine remain accurate but are the most expensive models to query, which bears directly on their suitability for real-time deployment. The default preference for deep models in IDS work is difficult to justify on these benchmarks.

5.2. COMPUTATIONAL EFFICIENCY AND DEPLOYMENT TRADE-OFFS

Scalability remains a central constraint for IDS deployment in resource-constrained environments such as IoT and edge computing. K-Nearest Neighbours requires 370.89 seconds to classify the CIC-IDS2017 test set against 1.28 seconds for the convolutional network, so it is not a low-latency choice at scale despite its low training cost. Random Forest combines the highest F1-score on every dataset with inference cost comparable to the cheapest deep models, and is the appropriate default for latency-sensitive deployment. Where a deep model is required, the multi-layer perceptron offers the lowest training and inference cost of the five at close to the accuracy of the recurrent architectures.

5.3. THE IDS-RECOMM CONCEPTUAL SCHEME

The dataset-model sensitivity differences and computational trade-offs established above form the basis for IDS-ReComm, a conceptual scheme intended as a decision-support aid for choosing an appropriate dataset, model and feature configuration based on deployment context. IDS-ReComm is a structured expression of the empirical findings of Section 4 and the guidance in Section 5.2 as a set of if-then selection rules, not a learned, adaptive, or software-implemented recommendation engine; a working implementation and its independent evaluation are proposed as future work (Fig. 12, Section 6).



Figure 12: IDS-ReComm workflow: dataset characteristics and operational requirements combine with empirically derived dataset-model sensitivity patterns and rule-based decision logic to produce configuration recommendations.

IDS-ReComm has four components. The Input Layer takes user-specified constraints such as computational budget, prioritised metrics, imbalance tolerance and real-time preference. The Rule-Based Selection Engine applies heuristic deduction and weighted scoring drawn from the empirical results of Section 4. The Knowledge Base is a repository of the statistical and performance data from this study's experiments. The Output Layer returns the recommended configuration together with supporting performance figures, statistical validation, and applicable caveats, so that recommendations remain auditable rather than a black box (Table 8, Fig. 13).

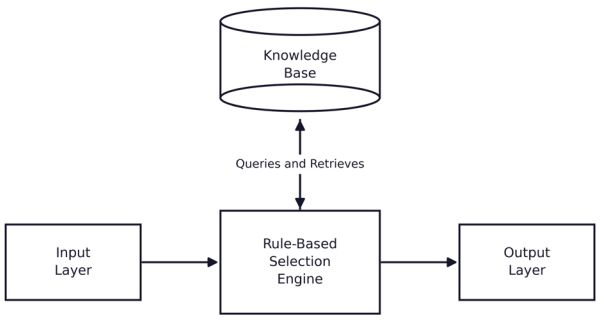


Figure 13: Components of IDS-ReComm: Input Layer, Rule-Based Selection Engine, Knowledge Base, and Output Layer.

http://dx.doi.org/10.21622/ACE.2026.06.2.2333

Table 8: Example Rule Specifications in IDS-ReComm

Condition	Recommended configuration
Low compute budget; latency-sensitive Random Forest with all features.	Highest F1-score on all four datasets and 0.43 s inference on the largest. Avoid KNN and SVM, whose inference cost scales with training-set size (370.89 s and 72.27 s on CIC-IDS2017).
High resource availability; F1 priority	Random Forest, or CNN, where a deep model is required. On CIC-IDS2017 these reach 99.79 and 99.52 mean F1-score respectively.
Heavily imbalanced target traffic	Report PR-AUC and F1-score, not accuracy. On HIKARI-2021 at 34.0:1 the two diverge by 21.1 percentage points for classical models.
Constrained feature budget	Prefer 20 features over 10. Mean F1-score falls by roughly 0.6 points from all features to 20, and by a further 2.8 points from 20 to 10.
Real-time or streaming deployment Advisory flag recommending pilot	validation, since all findings reflect offline evaluation only.

5.4. RECOMMENDATIONS FOR DATASET SELECTION

Table 9: Overall Dataset Assessment and Deployment Guidance

Dataset	Recommended models	Key observation
NSL-KDD	RF, DT, KNN	Highest classical F1-score of the four at 93.55 mean. Near-balanced, so accuracy and F1 agree to within 1.4 points. Suited to lightweight comparative work.
UNSW-NB15	RF; CNN where a deep model is required	Lowest deep learning F1-score of the four at 88.38. RF leads CNN by 0.81 points.
CIC-IDS2017	RF, DT; CNN for deep comparison	Highest deep learning F1-score at 98.45, but RF still leads at 99.79. Largest computational cost of the four in both training and inference.
HIKARI-2021	RF; interpret with care	First on accuracy and last on F1-score. Effective imbalance 34.0:1 after deduplication. Report PR-AUC and F1, and remove index and identifier columns before use.

Table 9 summarises this guidance by dataset. For academic research, dataset selection should align with research objectives, and reported metrics should suit the dataset's class balance rather than default to accuracy. For real-world deployment, CIC-IDS2017 and UNSW-NB15 need substantial preprocessing, and any HIKARI-2021-trained model should be validated on traffic with a realistic class balance before deployment claims are made. For model optimisation, feature selection should be strategic rather than aggressive, since the drop from twenty features to ten is roughly five times larger than the drop from all features to twenty.

5.5. LIMITATIONS

- Controlled rather than operational testing. Results reflect offline conditions; real-time validation in realistic network settings remains future work.
- Binary classification only, with fixed hyperparameters and architectures. Absolute optimum performance for each model and dataset pairing was not sought.
- Support vector machine training was capped at 100,000 stratified instances, so reported SVM results and timings understate an uncapped implementation.
- All timings reported here were measured on a single cluster node configuration with GPU acceleration disabled. Absolute values will differ on other hardware, and the relative comparisons between model families should be read as specific to CPU-only execution.
- Deduplication retains the first occurrence within a group of feature-identical flows. Because HIKARI-2021 is ordered by class, this resolution is order-dependent, and 34,246 of its flows carry contradictory labels.
- Deep models in the ten-feature setting frequently reached the 200-epoch cap without early stopping, so those results should be read as a lower bound rather than converged performance.

http://dx.doi.org/10.21622/ACE.2026.06.2.2333

- Cross-dataset transfer, adversarial robustness and multi-class detection are outside the present scope.
- IDS-ReComm is conceptual. It has not been implemented, and its rules are derived from four datasets under a single experimental protocol.

6. CONCLUSION

This study evaluated five machine learning and five deep learning models across four benchmark IDS datasets under six configurations each, repeated over five random seeds, using a pipeline in which every transform is fitted on training data alone. Dataset choice accounts for 9.5% of the variance in F1-score for classical models and 94.3% for deep models, confirming that classical models are markedly less dataset-sensitive while showing the effect to be small rather than absent. Random Forest attains the highest F1-score on all four datasets and among the lowest inference cost.

HIKARI-2021 ranks first on accuracy and last on F1-score. Its effective class imbalance after correct deduplication is 34.0:1 rather than 13.7:1, and 62.7% of its attack flows are exact duplicates, a structure that cannot be detected while a per-flow identifier defeats any duplicate check. An ablation on an identical row set shows that removing network identifiers changes F1-score by under one percentage point for eight of ten models. These observations bear on how IDS benchmarks are selected and reported as much as on how models are chosen.

Future work should extend the evaluation to multi-class settings and cross-dataset transfer, implement and validate IDS-ReComm rather than leaving it conceptual, and examine whether the duplicate and labelling structure documented here for HIKARI-2021 appears in other recent benchmarks.

ACKNOWLEDGMENTS

The authors acknowledge the High-Performance Computing facility at the National Institute of Technology Srinagar, on which all experiments reported in this study were executed. No external funding was received for this research.

DECLARATIONS

The authors declare no conflict of interest. The manuscript is original and unpublished, all necessary acknowledgements have been made, and all persons entitled to authorship are included and have approved the manuscript.

DATA AVAILABILITY

The complete experimental pipeline, submission scripts and automated consistency audit are available at https://github.com/ifrahnaqash/ids-recomm-evaluation. Per-run predicted labels and probabilities for all 1,500 runs are archived at https://doi.org/10.5281/zenodo.22485401. The four datasets analysed are not redistributed, as their licences rest with their original providers, and are publicly available from the sources cited in Table 1.

REFERENCES

[1] M. Alkasassbeh and S. Al-Haj Baddar, "Intrusion Detection Systems: A State-of-the-Art Taxonomy and Survey," *Arabian Journal for Science and Engineering*, vol. 48, no. 8, pp. 10021-10064, Nov. 2022, doi: 10.1007/s13369-022-07412-1.

[2] G. Kocher and G. Kumar, "Machine Learning and Deep Learning Methods for Intrusion Detection Systems: Recent Developments and Challenges," *Soft Computing*, Jun.

http://dx.doi.org/10.21622/ACE.2026.06.2.2333

2021, doi: 10.1007/s00500-021-05893-0.

[3] Md. A. Talukder *et al.*, "Machine Learning-Based Network Intrusion Detection for Big and Imbalanced Data Using Oversampling, Stacking Feature Embedding and Feature Extraction," *Journal of Big Data*, vol. 11, no. 1, p. 33, Feb. 2024, doi: 10.1186/s40537-024-00886-w.

[4] T. T. Nguyen and V. J. Reddi, "Deep Reinforcement Learning for Cyber Security," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 8, pp. 1–17, 2021, doi: 10.1109/tnnls.2021.3121870.

[5] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network Intrusion Detection System: A Systematic Study of Machine Learning and Deep Learning Approaches," *Transactions on Emerging Telecommunications Technologies,* vol. 32, no. 1, pp. 1–29, Oct. 2020, doi: 10.1002/ett.4150.

[6] D. Parsons, "Five Startling Findings in 2023'S ICS Cybersecurity Data," SANS Institute. Accessed: Sep. 22, 2024. [Online]. Available: https://www.sans.org/blog/five-startling-findings-2023-ics-cybersecurity-data

[7] R. M. Augusto and R. M. Augusto, "Strengthening ICS/OT Cyber Resilience: Learning From 2023'S Cybersecurity Incidents From Dragos' Report," Industrial Cyber. Accessed: Sep. 22, 2024. [Online]. Available: https://industrialcyber.co/expert/strengthening-ics-ot-cyber-resilience-learning-from-2023s-cybersecurity-incidents-from-dragos-report/

[8] J. O. Nehinbe, "A Critical Evaluation of Datasets for Investigating IDSs and IPSs Researches," in *2011 IEEE 10th International Conference on Cybernetic Intelligent Systems (CIS)*, Sep. 2011.

[9] A. Ferriyan, A. H. Thamrin, K. Takeda, and J. Murai, "Generating Network Intrusion Detection Dataset Based on Real and Encrypted Synthetic Attack Traffic," *Applied Sciences,* vol. 11, no. 17, p. 7868, Aug. 2021, doi: 10.3390/app11177868.

[10] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Jul. 2009, pp. 1–6.

[11] N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set)," in *2015 Military Communications and Information Systems Conference (MilCIS),* Nov. 2015, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7348942?casa_token=MCoyi5t44IQAAAAA:ZCODekObKojal3RTAApfo0ej9wNT7DtfDPGqq90k5CLbEyosH4XYCMu--0c2pObMagliyUGMwMk

[12] I. Sharafaldin, A. Gharib, A. H. Lashkari, and A. A. Ghorbani, "Towards a Reliable Intrusion Detection Benchmark Dataset," *Software Networking*, no. 1, pp. 177–200, 2017, doi: 10.13052/jsn2445-9739.2017.009.

[13] C. Zhang, F. Ruan, L. Yin, X. Chen, L. Zhai, and F. Liu, "A Deep Learning Approach for Network Intrusion Detection Based on NSL-KDD Dataset," in *2019 IEEE 13th International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, Oct. 2019, pp. 41–45.

[14] V. Kumar, D. Sinha, A. K. Das, S. C. Pandey, and R. T. Goswami, "An Integrated Rule Based Intrusion Detection System: Analysis on UNSW-NB15 Data Set and the Real Time Online Dataset," *Cluster Computing,* vol. 23, no. 2, pp. 1397–1418, Oct. 2019, doi: 10.1007/s10586-019-03008-x.

[15] S. Choudhary and N. Kesswani, "Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15

http://dx.doi.org/10.21622/ACE.2026.06.2.2333

Datasets Using Deep Learning in IoT," *Procedia Computer Science*, vol. 167, pp. 1561–1573, 2020, doi: 10.1016/j.procs.2020.03.367.

[16] M. Ghurab, G. Gaphari, F. Alshami, R. Alshamy, and S. Othman, "A Detailed Analysis of Benchmark Datasets for Network Intrusion Detection System," *Asian Journal of Research in Computer Science*, pp. 14–33, Apr. 2021, doi: 10.9734/ajrcos/2021/v7i430185.

[17] G. Engelen, V. Rimmer, and W. Joosen, "Troubleshooting an Intrusion Detection Dataset: The CICIDS2017 Case Study," in *2021 IEEE Security and Privacy Workshops (SPW)*, May 2021, pp. 7–12. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9474286

[18] L. Liu, G. Engelen, T. Lynar, D. Essam, and W. Joosen, "Error Prevalence in NIDS Datasets: A Case Study on CIC-IDS-2017 and CSE-CIC-IDS-2018," in *2022 IEEE Conference on Communications and Network Security (CNS)*, Oct. 2022, pp. 254–262.

[19] M. Lanvin, P.-F. Gimenez, Y. Han, F. Majorczyk, L. Mé, and É. Totel, "Errors in the CICIDS2017 Dataset and the Significant Differences in Detection Performances It Makes," *Lecture Notes in Computer Science*, pp. 18–33, 2023, doi: 10.1007/978-3-031-31108-6_2.

[20] R. Fernandes, J. Silva, Ó. Ribeiro, I. Portela, and N. Lopes, "The Impact of Identifiable Features in ML Classification Algorithms With the HIKARI-2021 Dataset," in *2023 11th International Symposium on Digital Forensics and Security (ISDFS)*, May 2023, pp. 1–5.

[21] D. Kwon, R.-M. Neagu, P. Rasakonda, J. T. Ryu, and J. Kim, "Evaluating Unbalanced Network Data for Attack Detection," in *Proceedings of the 2023 on Systems and Network Telemetry and Analytics, in SNTA '23. New York, NY, USA: Association for Computing Machinery*, Jul. 2023, pp. 23–26.

[22] Z. Zoghi and G. Serpen, "UNSW-NB15 Computer Security Dataset: Analysis Through Visualization," *Security and Privacy*, vol. 7, no. 1, Jun. 2023, doi: 10.1002/spy2.331.

[23] S. Layeghy, M. Gallagher, and M. Portmann, "Benchmarking the Benchmark — Comparing Synthetic and Real-World Network IDS Datasets," *Journal of Information Security and Applications*, vol. 80, p. 103689, Feb. 2024, doi: 10.1016/j.jisa.2023.103689.

[24] M. Catillo, A. Del Vecchio, A. Pecchia, and U. Villano, "Transferability of Machine Learning Models Learned From Public Intrusion Detection Datasets: The CICIDS2017 Case Study," *Software Quality Journal*, vol. 30, no. 4, pp. 955–981, Mar. 2022, doi: 10.1007/s11219-022-09587-0.

[25] M. Catillo, A. Del Vecchio, L. Ocone, A. Pecchia, and U. Villano, "USB-IDS-1: A Public Multilayer Dataset of Labeled Network Flows for IDS Evaluation," in *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, Jun. 2021, pp. 1–6.

[26] H. Jha, M. Khanna, H. Jhawar, and R. Jindal, "Performance Analysis of Deep Neural Network for Intrusion Detection Systems," *Lecture Notes in Networks and Systems*, pp. 445–455, 2023, doi: 10.1007/978-981-99-3758-5_41.

[27] S. J. Stolfo, W. Lee, W. Fan, A. Prodromidis, and P. K. Chan, "Index of /Databases/Kddcup99," Uci.edu. [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/

[28] A. Ferriyan, A. H. Thamrin, J. Murai, and K. Takeda, "HIKARI-2021: Generating Network Intrusion Detection Dataset Based on Real and Encrypted Synthetic Attack Traffic," *zenodo.org*, 2024, doi: 10.5281/zenodo.5199540.

http://dx.doi.org/10.21622/ACE.2026.06.2.2333

[29] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "NSL-KDD," www.kaggle.com. Accessed: Jun. 13, 2024. [Online]. Available: https://www.kaggle.com/datasets/hassan06/nslkdd

[30] N. Moustafa and J. Slay, "The UNSW-NB15 Dataset | UNSW Research," research.unsw.edu.au. Accessed: Jun. 13, 2024. [Online]. Available: https://research.unsw.edu.au/projects/unsw-nb15-dataset

[31] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Index of /CICDataset/CIC-IDS-2017/Dataset," 174.165.80. Accessed: Jun. 13, 2024. [Online]. Available: http://205.174.165.80/CICDataset/CIC-IDS-2017/Dataset/

[32] D. Elreedy and A. F. Atiya, "A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for Handling Class Imbalance," *Information Sciences*, vol. 505, pp. 32–64, Dec. 2019, doi: 10.1016/j.ins.2019.07.070.

[33] X. Deng, Q. Liu, Y. Deng, and S. Mahadevan, "An Improved Method to Construct Basic Probability Assignment Based on the Confusion Matrix for Classification Problem," *Information Sciences*, vol. 340–341, pp. 250–261, May 2016, doi: 10.1016/j.ins.2016.01.033.

[34] F. Gorunescu, *Data Mining: Concepts, Models and Techniques*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[35] G. Sameera, R. V. Vardhan, and K. V. S. Sarma, "Binary Classification Using Multivariate Receiver Operating Characteristic Curve for Continuous Data," *Journal of Biopharmaceutical Statistics*, vol. 26, no. 3, pp. 421–431, May 2015, doi: 10.1080/10543406.2015.1052479.