Machine Learning for Heterogeneous Network Traffic Classification: A Comparative Study Across Four Datasets

Qeis Khaled Qeis[1] and Guang Yang[1,2][0000−0002−3787−9616]

[1] Norwegian University of Life Sciences, Ås
[2] Western Norway University of Applied Sciences, Bergen

Abstract. Machine learning is widely used for network traffic classification, application identification, and intrusion detection. However, existing studies commonly evaluate classifiers on a single dataset or traffic domain, limiting the extent to which their conclusions generalize across heterogeneous classification tasks. This study compares commonly used machine-learning classifiers across four network traffic datasets: CIC-UNSW-NB15 for attack-category classification, CIC-Darknet2020 for application-type and security-label classification, ISCXVPN2016 for VPN and non-VPN application classification, and CESNET-QUIC22 for encrypted QUIC application classification. The evaluated classifiers include Logistic Regression, Linear Support Vector Machine, K-Nearest Neighbors, Random Forest, gradient-boosted trees, and Multi-Layer Perceptron. To improve comparability and reduce evaluation bias, the experimental procedure uses stratified data partitioning, training-only preprocessing and oversampling, fixed random seeds, and class-sensitive evaluation metrics. Performance is assessed primarily using macro-averaged F1 score, supplemented by Matthews correlation coefficient, accuracy, and per-class results. Among the classifiers evaluated consistently across all datasets, Random Forest achieved the highest macro F1 in every case, with scores ranging from 0.60 on CIC-UNSW-NB15 to 0.86 on ISCXVPN2016. On CESNET-QUIC22, extending the flow representation with packet-size and histogram features increased Random Forest macro F1 from 0.7216 to 0.8641, while a bidirectional LSTM operating on raw packet sequences achieved 0.7062. These findings indicate that tree-based ensembles provide reliable baselines across heterogeneous traffic-classification tasks and that feature representation can have a greater effect on performance than increasing model complexity.

Keywords: network traffic classification · machine learning · intrusion detection · encrypted traffic · cross-dataset analysis

1 Introduction

Network traffic classification is an important component of network management and cybersecurity. It supports tasks such as application identification, quality-of-service management, intrusion detection, traffic engineering and the identification of encrypted or tunnelled communications. Traditional traffic-classification

approaches commonly relied on transport-layer port numbers or inspection of packet payloads. These methods have become less reliable as applications increasingly use dynamic or shared ports and as encryption protocols such as TLS 1.3 and QUIC make application payloads unavailable to network observers.

Machine-learning methods provide an alternative by learning patterns from observable traffic characteristics. Classifiers can use flow-level features such as packet and byte counts, flow duration, packet-size statistics, inter-arrival times, traffic direction, and TCP flags, many of which remain observable when payloads are encrypted. Consequently, supervised machine learning has been widely applied to intrusion detection, application identification, VPN detection, and encrypted traffic classification.

Numerous algorithms have been evaluated for these tasks, including Logistic Regression, K-Nearest Neighbours, Support Vector Machines, Random Forests, gradient-boosted decision trees, and neural networks. However, reported results are difficult to compare because studies commonly use different datasets, classification objectives, features, preprocessing procedures, class-balancing methods, and evaluation metrics. Thus generalization across heterogeneous traffic environments cannot be established from a single benchmark. Strong performance on one dataset may not transfer to another dataset or classification task.

Class imbalance and evaluation design further complicate such comparisons. Accuracy and weighted metrics can indicate strong aggregate performance even when a classifier performs poorly on rare but important classes. Reported performance can also be inflated by duplicate observations across data partitions, oversampling before partitioning, preprocessing using information from the test set, or repeated test-set evaluation. A meaningful cross-dataset comparison must therefore consider both predictive performance and the methodology used to obtain it.

This paper presents a comparative empirical study of machine-learning classifiers across four heterogeneous network traffic datasets. CIC-UNSW-NB15 represents multiclass intrusion detection; CIC-Darknet2020 supports application-type and security-label classification; ISCXVPN2016 represents application classification under non-VPN and VPN-tunnelled conditions; and CESNET-QUIC22 represents encrypted QUIC application classification. Together, these datasets cover different traffic-analysis objectives, class distributions, feature spaces, and levels of encryption. The study investigates whether classifier performance patterns recur across these settings and how the observed differences relate to class distribution, classification objective, and feature representation.

This study makes three empirical contributions. First, it provides a common comparison of four classifiers across datasets representing intrusion detection, application identification, VPN and darknet traffic, and encrypted QUIC communication. The results show that Random Forest is the most consistent of the four common classifiers, while also showing that overall and minority-class performance varies substantially between tasks. Second, by applying two classification targets to the same CIC-Darknet2020 flows and feature schema, the study shows that the target definition affects both task difficulty and the fea-

tures used by the fitted models. In particular, port features are prominent for application classification, whereas TCP initial-window and timing features are more important for security classification. Third, the CESNET-QUIC22 experiment compares three model-and-representation configurations on the same test observations. Adding engineered packet-level summaries and histogram features increases Random Forest macro F1 from 0.7216 to 0.8641, while the evaluated bidirectional LSTM on raw packet sequences achieves 0.7062. This result shows that, under the evaluated configurations, enriching the input representation produces a larger gain than replacing the scalar Random Forest with a more complex sequential model. The experimental notebooks and source code are publicly available[3].

The remainder of the paper is organised as follows. Section 2 reviews related work. Section 3 introduces the datasets and classification tasks. Section 4 describes the experimental methodology. Section 5 presents the results, Section 6 discusses the cross-dataset findings and limitations, and Section 7 concludes the paper.

2 Related Work

Machine learning has long been used to classify Internet traffic from flow-level statistics. Early comparative work showed that application traffic could be distinguished without relying exclusively on registered ports, using properties such as packet sizes, durations and byte counts [6,7]. Random Forest is particularly suitable for such tabular data because bootstrap aggregation and random feature selection reduce correlation among individual trees [8]. More recent boosted-tree implementations, including XGBoost and LightGBM, provide strong alternatives by fitting trees sequentially to reduce residual prediction error [9,10].

Encryption has shifted traffic classification from payload signatures toward statistical and behavioural representations. Wang et al. demonstrated end-to-end classification using one-dimensional convolutional networks on encrypted traffic, while Rezaei and Liu surveyed the growing use of deep models for encrypted traffic classification [11,12]. Multimodal methods combine packet, flow and contextual information, but their advantage depends on whether the additional modality provides genuinely new signal [13]. QUIC creates a particularly difficult setting because it encrypts application data and much of the transport handshake. Studies of QUIC traffic nevertheless show that packet direction, size, timing and short packet sequences retain information about application behaviour [14,5].

Evaluation design remains a persistent concern. Sommer and Paxson argued that strong laboratory results do not automatically translate into operational intrusion detection because datasets, normal traffic and error costs differ from deployment conditions [15]. Dataset surveys similarly document substantial variation in collection environments, labelling methods and feature definitions [16].

[3] GitHub repository of the Jupyter Notebooks

Tavallaee et al. showed how duplicate records in KDD Cup 1999 could favour frequent attack patterns and inflate aggregate results [17]. Class imbalance further complicates interpretation: accuracy can remain high when rare but important classes are missed, motivating class-balanced metrics and careful resampling [18].

3 Datasets and Classification Tasks

Four public datasets were selected to represent different traffic-analysis objectives and levels of encryption. Table 1 summarises their use in this study. Each dataset was treated as an independent task; no cross-dataset transfer learning was performed. The resulting performance rankings, metric behaviour, and broad categories of influential features were then compared across tasks.

Table 1. Datasets and classification tasks used in the study.

Dataset	Task	Classes	Input representation
CIC-UNSW-NB15	Normal and attack-category classification	10	CICFlowMeter flow statistics; top 30 features retained
CIC-Darknet2020	Application classification	8	Flow-level statistical features
CIC-Darknet2020	Security/tunnelling classification	4	Flow-level statistical features
ISCXVPN2016	Plain and VPN application classification	14	Timing and volume statistics; top 15 features retained
CESNET-QUIC22	QUIC application classification	7	Eight scalar features, 52-feature extended representation, or PPI sequence

3.1 CIC-UNSW-NB15

CIC-UNSW-NB15 is an augmented version of the original UNSW-NB15 intrusion-detection dataset [1,2]. The original packet captures contain benign activity and nine attack categories: Analysis, Backdoor, Denial of Service, Exploits, Fuzzers, Generic Attacks, Reconnaissance, Shellcode and Worms. The Canadian Institute for Cybersecurity reprocessed the packet captures with CICFlowMeter and matched the extracted flows to the original ground truth using endpoint addresses, ports, protocols, and timestamps. The resulting dataset therefore uses CICFlowMeter flow statistics rather than the original UNSW-NB15 feature schema.

The released dataset contains 447,915 flows and 76 input features. Its 358,332 benign flows constitute approximately 80% of the observations because the dataset producers retained all extracted malicious flows but sampled the benign traffic. This proportion is therefore a dataset-construction choice rather than an estimate of operational attack prevalence. Removing nine constant

columns and 142,313 duplicate records left 305,602 flows and 67 candidate features. A Random Forest feature selector fitted to the training partition retained the 30 highest-ranked features. The final stratified split contained 244,481 training and 61,121 test observations.

3.2 CIC-Darknet2020

CIC-Darknet2020 [3] combines traffic from ISCXTor2016 and ISCXVPN2016. It contains ordinary, Tor and VPN traffic generated by applications including browsing, audio and video streaming, chat, email, file transfer, peer-to-peer software and voice over IP. Every flow has two targets: an application label and a security label indicating Non-Tor, NonVPN, Tor or VPN traffic. This dual-label structure enables two classification objectives to be studied using the same flow representation.

Inconsistent spelling and capitalisation in the application labels were normalised to eight classes: Audio-Streaming, Browsing, Chat, Email, File-Transfer, P2P, VoIP and Video-Streaming. Flow identifiers, endpoint IP addresses, and timestamps were removed, while ports and protocols were retained. After duplicate removal, the dataset was reduced from 158,616 to 119,577 flows. A shared 80/20 split, stratified by the application label, produced 95,661 training and 23,916 test flows. Missing and infinite values were handled using preprocessing fitted to the training partition. For the application task, training-only SMOTE [19] increased the training set to 97,906 observations.

3.3 ISCXVPN2016

ISCXVPN2016 [4] contains seven application types—Browsing, Chat, File Transfer, Mail, P2P, Streaming and VoIP—captured in both plain and OpenVPN-tunnelled conditions, producing 14 classes. The 23 available predictors consist of timing, byte-volume, rate, active-period and idle-period statistics. They therefore enable application classification without relying on payload content or inner application ports.

The raw data contained 59,706 observations. Removing 18,978 duplicates left 40,728 flows, which were divided into 32,582 training and 8,146 test observations using stratified sampling. A Random Forest feature selector fitted to the training partition retained 15 features. Some timing variables use -1 as a sentinel value when a statistic cannot be computed. Their measured importance may therefore represent both timing behaviour and the absence of sufficient packets in a particular direction.

3.4 CESNET-QUIC22

CESNET-QUIC22 [5] was collected from the CESNET national research and education network in the Czech Republic. The present study uses traffic from 5 November 2022. The selected daily file contains 2,453,828 flows, of which

1,441,868 were mapped to seven target categories: STREAMING, BROWSING, CHAT, FT, MAIL, P2P and VOIP. Categories that could not be mapped unambiguously to these targets were excluded.

In addition to standard bidirectional flow statistics, CESNET-QUIC22 provides packet-level information (PPI) and packet histograms. The PPI data record the inter-arrival time, direction and size of up to the first 30 packets in each flow. The dataset also provides summary variables derived from these sequences, including the number of recorded packets, sequence duration and estimated round trips. Packet histograms describe the distributions of packet sizes and inter-packet times in both communication directions. These fields allow traffic to be represented using either aggregate flow characteristics, engineered packet-level summaries or packet sequences.

We configured three representations for this dataset, and we compare them from the 1,436,422 flows for which PPI information was available. A stratified 80/20 split produced a shared test partition of 287,285 flows, ensuring that differences among the scalar Random Forest, extended Random Forest, and bidirectional LSTM were measured on the same observations.

4 Experimental Methodology

4.1 Evaluation Design

Each dataset was evaluated as an independent supervised multiclass task. CIC-Darknet2020 was evaluated under two target definitions: application type and security/tunnelling category. Models trained on one dataset were not tested on another because the datasets have different feature schemas, labels and collection conditions.

Dataset-level cleaning, label normalisation and exact-duplicate removal, where applicable, were performed before partitioning. A stratified 80/20 split with a fixed random seed then formed training and held-out test partitions. Scaling, feature selection and oversampling were fitted using only the training partition and were subsequently applied to the test partition without refitting. The test observations retained their original class distribution and were never oversampled.

Model configurations were fixed without using test-set performance. Each classifier was trained on the corresponding training partition and evaluated on the same held-out observations as the other classifiers for that task. This design supports controlled within-task comparison.

4.2 Classifiers

The principal cross-dataset comparison includes Logistic Regression, K-Nearest Neighbours, linear Support Vector Machine (SVM) and Random Forest. A Multi-Layer Perceptron and gradient-boosted tree implementations were evaluated in additional dataset-specific experiments. Together, these models represent linear,

instance-based, margin-based, bagged-tree, boosted-tree and neural approaches. Random Forest provides the principal ensemble baseline: its trees are fitted to bootstrap samples, consider random feature subsets at each split and aggregate their class predictions.

The exact classifier set differs slightly between datasets because of computational requirements and feature modality. Therefore, the cross-dataset analysis focuses on the four classifiers evaluated on every dataset and identifies performance patterns shared across the datasets.

4.3 Feature Selection and Imbalance Handling

For CIC-UNSW-NB15 and ISCXVPN2016, Random Forest mean-decrease-in-impurity rankings computed from the training partition were used to retain 30 and 15 features, respectively. The resulting feature subsets were then applied unchanged to the test partitions. CIC-Darknet2020 retained its cleaned flow-statistics representation, while CESNET-QUIC22 used the predefined scalar, extended and sequential representations described below.

Scale-sensitive models used a StandardScaler fitted to the training partition. SMOTE was used only for CIC-Darknet2020 and CESNET-QUIC22, whose training partitions contained severely underrepresented classes. For CIC-Darknet2020, application classes below 5,000 training observations were increased to that threshold; the same rule was applied separately to the security target. For CESNET-QUIC22, the three largest training classes were capped at 50,000 observations and the rare VOIP class was increased from 988 to 5,000 observations. No oversampling was applied to CIC-UNSW-NB15 or ISCXVPN2016. These operations affected only training data; all reported metrics were calculated on test data with their original distributions.

4.4 CESNET Model and Representation Configurations

Using the flow and packet information described in Section 3.4, three model-and-input configurations were evaluated on the shared PPI-eligible test partition. The first configuration used a Random Forest with eight scalar flow features. The second used an extended Random Forest with 52 features, combining the scalar variables with PPI summary statistics and packet-size histograms. The third used a bidirectional LSTM with the inter-arrival time, direction and packet size of the first 30 packets, represented as a 30×3 matrix. Shorter sequences were zero-padded. The LSTM processed each sequence in both directions before multiclass prediction through a softmax output layer.

The extended Random Forest and LSTM do not receive identical inputs. Their comparison therefore evaluates complete model-and-representation configurations rather than isolating classifier architecture. The experiment asks whether the selected raw packet sequence offers an advantage over engineered distributional summaries under the evaluated configurations.

4.5 Evaluation Metrics

Macro F1 is the principal metric for cross-task interpretation:

$$F1_{\text{macro}} = \frac{1}{K}\sum_{k=1}^{K}\frac{2P_kR_k}{P_k + R_k}, \tag{1}$$

where K is the number of classes and P_k and R_k are the precision and recall for class k. Macro F1 weights every class equally and is therefore sensitive to failure on rare categories. Matthews correlation coefficient(MCC) is used as a complementary measure based on the complete confusion matrix. Accuracy and weighted F1 are also reported for comparison with earlier work, while confusion matrices and class-level scores support interpretation of minority-class behaviour.

4.6 Feature Importance

Random Forest feature importance is calculated as mean decrease in Gini impurity. For a split at node t, the decrease is

$$\Delta I(t) = I(t) - \frac{N_L}{N_t}I(t_L) - \frac{N_R}{N_t}I(t_R), \tag{2}$$

where $I(t)$ denotes parent impurity, t_L and t_R are the child nodes, and N_L, N_R and N_t are their sample counts. Reductions assigned to a feature are accumulated across splits and trees and then normalised.

These values describe how strongly the fitted forest relied on each variable; they do not demonstrate causality. Mean-decrease-in-impurity importance may favour continuous or high-cardinality variables, while correlated features can divide their importance. We therefore emphasise broad feature categories and substantial cross-task differences rather than small changes in individual ranks.

4.7 Cross-Dataset Analysis

Performance is first assessed independently for every task. The cross-dataset analysis then compares classifier ordering, differences between aggregate and class-balanced metrics, and influential feature categories. Since the datasets differ in labels, collection environments and feature spaces, raw scores are not treated as measurements of one common problem. The analysis instead identifies observations that recur across the evaluated settings and distinguishes them from dataset-specific findings.

5 Results

5.1 Classifier Performance Across Tasks

Table 2 compares the four classifiers evaluated consistently across the four principal scalar-feature tasks. Random Forest achieved the highest macro F1 in every

case, ranging from 0.60 on CIC-UNSW-NB15 to 0.86 on ISCXVPN2016. KNN ranked second in all four tasks, while Logistic Regression and linear SVM performed particularly poorly on ISCXVPN2016 and CESNET-QUIC22. This ordering is consistent with a benefit from modelling nonlinear interactions among timing, volume and directional features; however, differences in model configuration and sensitivity to class imbalance may also contribute.

Table 2. Macro F1 for the four classifiers common to the principal scalar-feature experiments. Best values in each row are bold.

Task	LR	Linear SVM	KNN	RF
CIC-UNSW-NB15 attack category	0.30	0.19	0.53	**0.60**
CIC-Darknet2020 application	0.4829	0.4690	0.7248	**0.8187**
ISCXVPN2016 application/VPN	0.19	0.18	0.66	**0.86**
CESNET-QUIC22 scalar	0.1689	0.1326	0.4858	**0.7083**

The larger model sets qualify this pattern. On CIC-UNSW-NB15, XGBoost produced slightly higher accuracy and weighted F1 than RF (0.9113 and 0.9095 versus 0.9077 and 0.9067), but a slightly lower macro F1 (0.59 versus 0.60). On CIC-Darknet2020 application classification, RF and HistGradientBoosting both achieved macro F1 of 0.8187, while HistGradientBoosting had marginally higher MCC (0.8683 versus 0.8630). On ISCXVPN2016, LightGBM and XGBoost each reached macro F1 of 0.87, compared with 0.86 for RF.

The CIC-Darknet2020 security task was evaluated separately using RF. It achieved macro F1 of 0.9562, MCC of 0.9675 and accuracy of 0.9816, compared with application-task macro F1 of 0.8187 on the same test observations. Although the tasks have different class counts and imbalance profiles, the contrast shows that apparent difficulty depends strongly on the target definition: the four tunnelling/security categories were more readily separated than the eight application categories.

5.2 Class Imbalance and Class-Level Behaviour

Aggregate performance concealed important class-level weaknesses. On CIC-UNSW-NB15, RF obtained accuracy of 0.9077 and weighted F1 of 0.9067, but macro F1 was only 0.60. Its F1 scores for the rare Analysis, Backdoor and Worms classes were 0.31, 0.65 and 0.43, respectively. The high weighted result therefore reflects performance on the dominant Normal class and more frequent attacks rather than uniformly strong attack-category recognition.

CIC-Darknet2020 showed a different error pattern. For application classification, RF achieved F1 of 0.99 for P2P and 0.95 for Browsing. VoIP recall was 0.86, but precision was only 0.44, producing F1 of 0.58 because many Chat and Email flows were predicted as VoIP. By contrast, the smaller Tor class in the

security task achieved F1 of 0.91 after training-only SMOTE, although its recall of 0.87 remained the lowest among the four security categories.

On ISCXVPN2016, RF achieved F1 of 0.98 for VoIP and 0.96 for VPN-VoIP, a result consistent with regular bidirectional timing remaining informative under VPN encapsulation. VPN-Chat and VPN-Mail were the most difficult categories, each with F1 of 0.77. In the original mapped CESNET-QUIC22 data, STREAMING outnumbered VOIP by approximately 530:1. After training-only capping and oversampling, RF achieved F1 of 0.82 for STREAMING but only 0.43 for VOIP. Thus, resampling increased minority-class representation during training but did not eliminate feature-space overlap or the limitations of small real minority-class support.

5.3 Feature Importance Across Classification Objectives

Table 3 presents the most important features identified by the RF models. Since each dataset has different features and a separately trained model, the importance values cannot be compared directly across datasets. Instead, we examine which types of features are commonly important and how their importance changes when the same data are used for different classification tasks.

Table 3. Leading mean-decrease-in-impurity features for the Random Forest models.

Task	Five highest-ranked features
CIC-UNSW-NB15	FWD Init Win Bytes (0.0991); Fwd Seg Size Min (0.0910); Bwd Packets/s (0.0594); Bwd Init Win Bytes (0.0544); Flow IAT Mean (0.0424)
CIC-Darknet application	Idle Max (0.0830); Dst Port (0.0697); Idle Min (0.0539); Idle Mean (0.0539); Src Port (0.0502)
CIC-Darknet security	FWD Init Win Bytes (0.0899); Bwd Init Win Bytes (0.0823); Idle Max (0.0711); Idle Mean (0.0492); Idle Min (0.0417)
ISCXVPN2016	flowBytesPerSecond (0.0962); min_biat (0.0920); min_flowiat (0.0867); max_flowiat (0.0766); min_fiat (0.0742)
CESNET-QUIC22 scalar	BYTES_REV (0.2389); BYTES (0.1656); DURATION (0.1324); PACKETS_REV (0.1246); PPI_DURATION (0.1167)

The CIC-Darknet2020 comparison provides the most controlled view of target dependence because both objectives use the same observations and feature schema. Ports were prominent in the application ranking but absent from the security top ten, whereas the forward and backward TCP initial-window variables occupied the first two positions for security classification. Six features appeared in both top-ten lists, including three idle-duration measures and two flow inter-arrival-time measures. The fitted models therefore relied on an overlapping

timing core together with target-specific predictors, cautioning against reusing one feature subset for multiple objectives without validation.

For ISCXVPN2016, the five leading variables were timing or throughput statistics. This is consistent with application information remaining observable after VPN encapsulation, although the -1 sentinel used for unavailable timing measurements means that some importance may represent packet availability rather than timing magnitude alone. In the CESNET scalar model, reverse byte volume was the leading feature, followed by total bytes, duration and reverse packet count; the fitted forest therefore relied primarily on directional volume and duration.

5.4 Representation Comparison on CESNET-QUIC22

Table 4 compares the three CESNET configurations on the same 287,285 PPI-eligible test flows. The scalar RF score of 0.7216 differs from the 0.7083 reported in Table 2 because the latter belongs to the principal scalar experiment, whereas this comparison uses the subset shared by all three configurations. On the shared partition, the extended RF improved macro F1 by 0.1425 over the scalar RF. The bidirectional LSTM achieved higher accuracy and MCC than the scalar RF but lower macro F1, reflecting weaker precision for rare P2P and VOIP traffic.

Table 4. CESNET-QUIC22 model-and-representation comparison on the shared test partition.

Configuration	Macro F1	MCC	Accuracy
Scalar RF, 8 features	0.7216	0.6975	0.7872
Extended RF, 52 features	**0.8641**	**0.8556**	**0.8995**
Bidirectional LSTM, 30×3 PPI	0.7062	0.7340	0.8069

The extended model's leading variables were minimum PPI packet size (0.0716), maximum PPI packet size (0.0569) and reverse bytes (0.0532). Six packet-size histogram bins also appeared among its top 15 features. This shift indicates that the distribution of packet sizes carried information not available from aggregate byte and packet totals. Under the evaluated configurations, engineered packet-distribution summaries were more effective than either the scalar baseline or direct recurrent processing of the first 30 packets. Because the LSTM and extended RF received different representations, the comparison does not establish an inherent superiority of RF over sequence models.

5.5 Summary of Findings

Three findings recur across the experiments. First, tree ensembles formed the leading group on the evaluated tabular tasks, with RF providing the most consistent result among the four common classifiers. Second, high accuracy or weighted

F1 did not imply strong minority-class recognition. Third, timing and volume features recurred across datasets, but their specific importance depended on the feature extractor and target. The results therefore support neither a universally optimal classifier nor a transferable feature ranking; performance reflects the interaction among dataset, objective, representation and evaluation metric.

6 Discussion and Limitations

6.1 Tree Ensembles as Strong Tabular Baselines

The most consistent cross-task result is the strength of tree ensembles on flow-level tabular features. RF led the four common classifiers on every principal task, while XGBoost, LightGBM and HistGradientBoosting matched or slightly exceeded it in some expanded comparisons. No boosted implementation dominated across all datasets. The evidence therefore supports tree ensembles as a strong model family and RF as a useful baseline, rather than identifying a universally optimal algorithm.

The gap between the tree ensembles and the linear baselines is consistent with useful decision boundaries depending on nonlinear interactions among timing, volume and direction. KNN captured some of this structure, but the feed-forward MLP did not perform better than the tree ensembles. However, these results apply only to the models and settings evaluated in this study and do not mean that nonlinear models will always outperform linear models.

6.2 Metrics and Minority Classes

CIC-UNSW-NB15 provides the clearest illustration of why accuracy should not be reported alone. Accuracy and weighted F1 were both above 0.90, whereas macro F1 was 0.60 and several rare attack classes remained weak. Weighted measures were dominated by the frequent Normal class and common attacks, while macro F1 exposed uneven class-level performance. This distinction is operationally important because rare attacks may carry greater cost than their frequency suggests.

Increasing minority-class counts through SMOTE did not remove the limitations of scarce real examples or overlapping features. CESNET-QUIC22 contained only 988 real VOIP training flows before oversampling. Its weak precision and F1 show that balancing sample counts does not guarantee a reliable minority decision boundary. For operationally important rare classes, collecting more genuine observations is preferable to relying primarily on synthetic examples.

Macro F1 and MCC provided complementary views. The CESNET LSTM improved accuracy and MCC over the scalar RF but reduced macro F1 because false positives harmed rare-class precision. Model selection should therefore be tied to operational error costs rather than to a single aggregate metric.

6.3 Feature Importance Is Task Dependent

Timing, traffic volume and direction were among the important feature categories in several datasets. This suggests that these characteristics can still provide useful information when packet payloads are unavailable. However, the rankings of individual features changed between tasks. In CIC-Darknet2020, for example, port numbers ranked highly for application classification but did not appear among the ten most important features for security classification. In contrast, TCP initial-window features became more important for the security task. Since both tasks used the same observations and feature schema, the difference is mainly related to the classification target and its class distribution. A feature set selected for one task should therefore be evaluated again before it is used for another task.

Feature-importance values should also be interpreted carefully. They show which variables were used by the fitted RF models, but they do not show that those variables have a causal effect on the traffic class. Mean-decrease-in-impurity importance may favour variables with many possible split points. Another issue occurs in ISCXVPN2016, where unavailable timing values are represented by -1. In this case, an important timing feature may partly indicate whether enough packets were available to calculate the value. Future work could use permutation importance or feature-group ablation to check whether the same patterns remain.

6.4 Representation Versus Architecture

The CESNET-QUIC22 results show that the choice of input representation had a large effect on classification performance. When PPI summary statistics and packet-size histograms were added, RF macro F1 increased from 0.7216 to 0.8641. The bidirectional LSTM, which used the first 30 packets of each flow, achieved a macro F1 of 0.7062. Packet-size minimum and maximum values, together with several histogram bins, were among the most important features in the extended RF. This indicates that the distribution of packet sizes contained useful information that was missing from the eight-feature scalar representation.

However, this experiment is not a direct comparison between RF and LSTM architectures. The extended RF used engineered statistical features, whereas the LSTM received a truncated sequence of packet-level values. The observed difference may therefore be caused by the input representation, preprocessing, model architecture or a combination of these factors. Based on the present experiment, we can conclude only that the engineered packet-distribution features gave the best result among the three configurations tested. A direct comparison of model architectures would require all models to use the same packet-level input.

6.5 Limitations

Several limitations should be considered when interpreting the results. First, the four datasets represent different classification problems, feature extractors, collection environments and class definitions. Comparing relative patterns across

them is informative, but raw macro F1 values are not measurements of one common task. The study is therefore a cross-dataset comparison, not cross-dataset transfer learning.

Second, the reported results are based on one stratified train–test split for each dataset. Using the same test observations provides a fair comparison among models within a task, but it does not show how much the results would change under another split. Small differences, particularly those between RF and the boosted-tree models, should therefore be treated with caution.

The experimental settings were also not completely uniform across datasets. Different feature-selection thresholds and imbalance-handling procedures were used because the datasets have different structures. For CIC-Darknet2020, computational limitations required the linear SVM and conventional Gradient Boosting models to be trained on smaller stratified subsets. Only RF was evaluated for the CIC-Darknet2020 security task. These differences limit the extent to which all classifier results can be compared directly.

7 Conclusion

This paper compared machine-learning classifiers across four heterogeneous network traffic datasets covering intrusion detection, application identification, VPN and darknet traffic, and encrypted QUIC communication. Among the four classifiers common to the principal scalar experiments, RF achieved the highest macro F1 on every task, while boosted-tree models matched or slightly exceeded it in some extended comparisons. Tree ensembles were therefore the most consistently competitive model family for the evaluated tabular features.

The experiments also showed that aggregate performance depends on task definition, class distribution and feature representation. Accuracy and weighted F1 concealed weak minority-class recognition in CIC-UNSW-NB15, while VOIP remained difficult in CESNET-QUIC22 despite resampling. Feature-importance rankings changed between the CIC-Darknet application and security targets, supporting objective-specific feature validation. For QUIC traffic, the extended RF using packet-level summaries and histograms achieved macro F1 of 0.8641, compared with 0.7216 for the scalar RF and 0.7062 for the evaluated bidirectional LSTM. This comparison supports representation design as a central modelling decision, but not a general architectural ranking.

Overall, no single benchmark can establish universal superiority of a classifier or feature set. Future work should use repeated, grouped or temporal evaluation, independent deployment captures and controlled architecture comparisons on identical packet-level inputs.

References

1. Moustafa, N., Slay, J.: UNSW-NB15: A comprehensive data set for network intrusion detection systems. In: MilCIS 2015, pp. 1–6. IEEE (2015). https://doi.org/10.1109/MilCIS.2015.7348942

2. Mohammadian, H., Habibi Lashkari, A., Ghorbani, A.A.: Poisoning and evasion: Deep learning-based NIDS under adversarial attacks. In: 21st Annual International Conference on Privacy, Security and Trust. IEEE (2024). Dataset: https://www.unb.ca/cic/datasets/cic-unsw-nb15.html
3. Habibi Lashkari, A., Kaur, G., Rahali, A.: DIDarknet: A contemporary approach to detect and characterize darknet traffic using deep image learning. In: ICCNS 2020, pp. 1–13. ACM (2020). https://doi.org/10.1145/3442520.3442521
4. Draper-Gil, G., Habibi Lashkari, A., Mamun, M.S.I., Ghorbani, A.A.: Characterization of encrypted and VPN traffic using time-related features. In: ICISSP 2016, pp. 407–414. SciTePress (2016). https://doi.org/10.5220/0005740704070414
5. Luxemburk, J., Hynek, K., Cejka, T., Lukacovic, A., Siska, P.: CESNET-QUIC22: A large one-month QUIC network traffic dataset from backbone lines. Data in Brief **46**, 108888 (2023). https://doi.org/10.1016/j.dib.2023.108888
6. Nguyen, T.T.T., Armitage, G.: A survey of techniques for Internet traffic classification using machine learning. IEEE Communications Surveys & Tutorials **10**(4), 56–76 (2008). https://doi.org/10.1109/SURV.2008.080406
7. Williams, N., Zander, S., Armitage, G.: A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. ACM SIGCOMM Computer Communication Review **36**(5), 5–16 (2006). https://doi.org/10.1145/1163593.1163596
8. Breiman, L.: Random forests. Machine Learning **45**(1), 5–32 (2001). https://doi.org/10.1023/A:1010933404324
9. Chen, T., Guestrin, C.: XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). Association for Computing Machinery, New York, NY, USA, 785–794. https://doi.org/10.1145/2939672.2939785
10. Ke, G., Meng, Q., Finley, T., et al.: LightGBM: A highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems 30, pp. 3146–3154 (2017)
11. Wang, W., Zhu, M., Wang, J., Zeng, X., Yang, Z.: End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In: IEEE ISI 2017, pp. 43–48 (2017). https://doi.org/10.1109/ISI.2017.8004872
12. Rezaei, S., Liu, X.: Deep learning for encrypted traffic classification: An overview. IEEE Communications Magazine **57**(5), 76–81 (2019). https://doi.org/10.1109/MCOM.2019.1800819
13. Aceto, G., Ciuonzo, D., Montieri, A., Pescapè, A.: MIMETIC: Mobile encrypted traffic classification using multimodal deep learning. Computer Networks **165**, 106944 (2019). https://doi.org/10.1016/j.comnet.2019.106944
14. Luxemburk, J., Hynek, K., Cejka, T.: Encrypted traffic classification: The QUIC case. In: TMA 2023, pp. 1–10. IFIP (2023). https://doi.org/10.23919/TMA58422.2023.10199052
15. Sommer, R., Paxson, V.: Outside the closed world: On using machine learning for network intrusion detection. In: IEEE Symposium on Security and Privacy, pp. 305–316 (2010). https://doi.org/10.1109/SP.2010.25
16. Ring, M., Wunderlich, S., Scheuring, D., Landes, D., Hotho, A.: A survey of network-based intrusion detection data sets. Computers & Security **86**, 147–167 (2019). https://doi.org/10.1016/j.cose.2019.06.005
17. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A detailed analysis of the KDD CUP 99 data set. In: IEEE CISDA 2009, pp. 1–6 (2009). https://doi.org/10.1109/CISDA.2009.5356528

18. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering **21**(9), 1263–1284 (2009). https://doi.org/10.1109/TKDE.2008.239
19. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research **16**, 321–357 (2002). https://doi.org/10.1613/jair.953