

Comparative Classification of DNS-over-HTTPS Traffic on the CIRA-CIC-DoHBrw-2020 Dataset Using Logistic Regression, Random Forest and XGBoost

Ikiomoye Douglas Emmanuel¹

¹Centre for Artificial Intelligence and Multidisciplinary Innovations, Department of Auditing, College of Accounting Sciences, University of South Africa, Pretoria 0002, South Africa.

Received:

July 8, 2026

Revised:

September 5, 2026

Accepted:

September 15, 2026

Published:

October 24, 2026

Corresponding Author:

Author Name*:

Ikiomoye Douglas
Emmanuel

Email*:

emmanid@unisa.ac.za

DOI:

10.63158/journalisi.v8i5.1914

© 2026 Journal of Information Systems and Informatics. This open access article is distributed under a (CC-BY License)



Abstract. DNS over HTTPS (DoH) improves the confidentiality of domain-name resolution but reduces the visibility available to enterprise monitoring and security systems because DNS is carried within encrypted HTTPS traffic. This study compares Logistic Regression, Random Forest, and XGBoost for distinguishing DoH from non-DoH HTTPS traffic while examining predictive performance, computational cost, and model interpretability. The models were evaluated using the cleaned Layer 1 representation of the CIRA-CIC-DoHBrw-2020 dataset, which contains 1,019,117 flow records and 29 statistical predictors. The controlled protocol used three repeated 80:20 stratified development-test splits (seeds 42, 52, and 62), three-fold cross-validation, class-imbalance treatment, held-out testing, execution-time measurement, and permutation importance. Random Forest produced the strongest threshold-dependent results, achieving a macro F1 of 0.9965, MCC of 0.9930, and false-positive rate of 0.0011. XGBoost obtained the highest Average Precision (0.9994) and required a mean training time of 31.40 seconds, approximately 95.8% lower than Random Forest (743.42 seconds). Logistic Regression was computationally efficient but yielded lower DoH precision and a false-positive rate of 0.1197. PacketLengthMode was the most influential feature for both tree-based models. The results demonstrate strong within-benchmark separation, but they do not establish cross-network or operational generalisation.

Keywords: DNS over HTTPS; Encrypted Traffic Classification; Random Forest; XGBoost; Feature Permutation

1. INTRODUCTION

The Domain Name System (DNS) is a fundamental Internet service that translates domain names into network addresses. Conventional DNS exchanges are commonly transmitted without transport encryption, allowing intermediaries to observe or manipulate resolution traffic. DNS over HTTPS (DoH) was standardised in RFC (Request for Comment) 8484 to improve confidentiality and integrity by mapping each DNS query-response pair into an HTTPS exchange protected by Transport Layer Security [1]. Although this protects DNS messages from passive observation and modification while in transit, it also reduces the visibility traditionally available to network administrators and security-monitoring systems [1].

The monitoring challenge arises because DoH is carried over the same HTTPS infrastructure and commonly used transport ports as ordinary encrypted web traffic. Port-based identification is therefore insufficient, while reliance on known resolver addresses may fail to identify private, newly deployed or changing DoH services. DoH can also be abused to bypass organisational DNS controls or conceal malicious communications, including command-and-control and covert data transfer [2]. This creates a particular problem for enterprise security infrastructure because TLS limits content-based deep packet inspection of DNS messages. Consequently, passive analysis of observable flow telemetry provides a potentially less invasive means of identifying DoH without decrypting application payloads. Such analysis can exploit characteristics that remain visible after encryption, including packet-length distributions, timing behaviour, transferred volume and connection duration [2], [3].

Early research demonstrated the feasibility of this approach. [4] analysed extended HTTPS flow information generated through DoH-enabled Firefox, Google Chrome and Cloudflare and evaluated five machine-learning classifiers. Their findings demonstrated that DoH could be identified with high accuracy within the constructed experimental environment. However, the study primarily established the feasibility of encrypted-flow recognition and did not determine whether equivalent performance would persist across datasets, network environments or alternative partitioning strategies. [5] subsequently introduced CIRA-CIC-DoHBrw-2020 and formulated the problem as a two-layer classification task. Layer 1 distinguishes DoH from non-DoH HTTPS traffic, whereas Layer

2 distinguishes benign from malicious DoH traffic. The dataset therefore provides an established benchmark for investigating DoH detection independently from malicious DoH characterisation. The approach combined statistical flow variables with time-series features and showed that machine-learning classifiers could exploit observable encrypted-traffic patterns. This research provided an important benchmark for subsequent DoH research, although its two-stage objective differs from studies concerned only with controlled comparison of Layer 1 classifiers. Later studies extended the state of the art through deep-learning methods and independently collected traffic. Subsequent studies broadened the classification approaches and evaluation environments. [6] evaluated conventional and deep-learning classifiers using CIRA-CIC-DoHBrw-2020 and reported strong detection performance. The study emphasised predictive modelling and comparison of machine-learning and neural approaches, rather than jointly examining threshold behaviour, ranking performance, computational cost and model-agnostic feature dependence. [7] collected DoH traffic across multiple services and locations and examined the statistical characteristics that supported classification. Their findings reinforced the usefulness of packet-level and flow-level patterns, while recognising that client, resolver, geographic and protocol configurations may affect detection performance. A separate research direction has sought to improve operational deployability. From an operational perspective, [8] observed that several proposed detectors depend on specialised exporters or features that are difficult to obtain from standard monitoring infrastructure. They developed an approach combining conventional flow telemetry, machine learning, resolver information and active probing, and evaluated it using laboratory and ISP (Internet Service Provider)-derived traffic. This improved practical applicability and demonstrated that DoH detection could be integrated with commonly available monitoring systems. However, its objective and multi-component architecture differ from a controlled comparison of individual classifier families using one fixed statistical feature representation.

A broader comparative evaluation was conducted by [9], subsequently conducted one of the most comprehensive comparative studies of DoH detection. Seven published detectors were reconstructed and evaluated through six experiments addressing transferability, usability and performance longevity. Their results showed that strong within-environment performance did not necessarily persist when models were transferred across datasets or network settings. The study also identified inconsistencies

in earlier evaluation procedures and demonstrated the importance of comparable datasets and reproducible model descriptions. Nevertheless, its purpose was to compare heterogeneous published detector implementations across several deployment scenarios, rather than to isolate the behaviour of three representative model families under one verified Layer 1 protocol.

The reviewed literature therefore establishes that DoH can be identified from encrypted-flow statistics and that tree-based, deep-learning and hybrid detectors can achieve strong benchmark performance. However, there have been limited studies concerned with isolating the comparative behaviour of a linear classifier, a bagging ensemble and a boosting ensemble on the same cleaned CIRA-CIC-DoHBrw-2020 Layer 1 representation while jointly examining threshold-dependent classification performance, imbalance-sensitive ranking measures, false-positive behaviour, computational cost and model-agnostic feature dependence. Differences in datasets, classifier objectives, feature sets, balancing procedures and evaluation designs make direct comparison across earlier studies difficult.

Therefore, this research intends to provide a controlled within-dataset comparison of Logistic Regression, Random Forest and XGBoost using a common statistical feature representation and experimental protocol. The objectives of this research are to: determine how effectively the three classifier families distinguish DoH from non-DoH HTTPS traffic; examine whether their relative performance changes across threshold-dependent, ranking-based and class-imbalance-sensitive evaluation measures; and identify the statistical flow characteristics on which the fitted models depend most strongly. The novelty does not lie in introducing a new classifier or claiming the first comparative study of DoH detection. Instead, the contribution is a controlled model-level benchmark that jointly examines predictive performance, ranking quality, false-positive behaviour, computational efficiency and permutation-based feature dependence under consistent experimental conditions.

Accordingly, the study addresses the following research questions:

- 1) RQ1: How effectively do Logistic Regression, Random Forest and XGBoost distinguish DoH from non-DoH HTTPS traffic using the CIRA-CIC-DoHBrw-2020 Layer 1 dataset?

- 2) RQ2: How does the relative performance of the classifiers vary across threshold-dependent, ranking-based and class-imbalance-sensitive evaluation measures?
- 3) RQ3: Which statistical flow characteristics contribute most strongly to the predictions of the evaluated classifiers?

The study therefore provides a transparent within-dataset benchmark for DoH versus non-DoH HTTPS classification using statistical flow characteristics. It compares predictive quality and computational requirements under a common evaluation procedure and examines the variables on which the fitted classifiers depend. The resulting evidence is intended to establish a reproducible model-level baseline rather than claim operational generalisation, which requires subsequent group-aware, temporal, cross-dataset and live-network evaluation.

2. LITERATURE REVIEW

This section provides a thematic review of encrypted DNS traffic analysis, machine learning detection, feature design, operational monitoring and evaluation reliability. Unlike the Introduction, which establishes the research problem and contribution, the present section critically synthesises how previous studies have approached DoH classification and where their methodological choices limit direct comparison.

1) Statistical Visibility in Encrypted DNS Traffic

DoH protects DNS message contents by carrying resolution traffic within encrypted HTTPS connections, but it does not eliminate all observable network metadata. Packet size, direction, timing, flow duration and transferred volume may remain visible to passive monitoring systems. [10] classified these characteristics as part of the broader inference surface created by encrypted DNS and reviewed their use in normal-traffic profiling, malicious-activity detection and traffic-analysis attacks. The literature therefore treats encryption as altering the available feature space rather than rendering DNS communication statistically indistinguishable from other encrypted traffic.

[11] demonstrated that encrypted DNS traces could support domain inference in both closed-world and open-world settings. Their findings showed that traffic-analysis models

could exploit statistical patterns despite message encryption, although performance was affected by changes in resolver, client, platform and collection context. The study is important because it establishes the persistence of observable structure in encrypted DNS. However, its objective was to infer user activity and visited domains rather than to distinguish DoH from ordinary HTTPS traffic, so its reported results are not directly comparable with binary DoH-detection experiments. [12] examined whether padding could prevent traffic inference from encrypted DNS. Their method combined packet-size and timing sequences and showed that padding alone did not remove all information available to a passive observer. This result supports the use of packet and temporal statistics in defensive detection, but it also illustrates the privacy implications of such monitoring. Collectively, these studies show that the same residual characteristics that allow security systems to identify encrypted DNS may also permit adversaries to infer user behaviour. Privacy inference and security classification must nevertheless be distinguished. Domain fingerprinting attempts to determine which resource a user accessed, whereas DoH detection determines whether an encrypted flow belongs to a particular protocol or security class. The tasks may rely on overlapping packet-length and timing variables, but they differ in labels, data construction, evaluation criteria and operational consequences. Evidence that encrypted DNS leaks statistical information therefore motivates DoH detection but does not by itself establish the effectiveness or generalisability of a particular classifier.

2) Machine-Learning Detection and Benchmark Development

Early DoH research showed that machine-learning classifiers could distinguish DoH from conventional HTTPS using flow statistics rather than payload contents or destination addresses. [4] demonstrated strong recognition using traffic from DoH-enabled browsers and cloudflared, although the evaluation was limited to the constructed environment and did not establish cross-dataset or cross-network generalisation. [5] introduced CIRA-CIC-DoHBrw-2020 as a two-layer benchmark, with Layer 1 separating DoH from non-DoH traffic and Layer 2 distinguishing benign from malicious DoH, making the two tasks methodologically distinct.

Study by [13] compared ten classifiers for malicious DoH detection and used feature selection and explainability to improve interpretation, but their focus aligns more closely with Layer 2. [6] compared conventional machine-learning and deep-learning approaches

on CIRA-CIC-DoHBrw-2020 and reported strong performance, although the study placed less emphasis on jointly assessing threshold errors, ranking measures, computational cost and model-agnostic feature importance. [7] showed that packet-level and flow-level statistics remained useful across independently collected DoH traffic from different services and locations, though their results are not directly comparable because of differences in data generation and feature construction. More recently, [14] proposed a packet-level real-time system for malicious DoH tunnel detection, reflecting a shift towards operational monitoring, but this differs from the present study's-controlled comparison of three classifier families on the Layer 1 flow representation.

3) Feature Economy, Interpretation and Operational Monitoring

Feature engineering aims to balance predictive performance with the collection, storage and processing costs of high-dimensional traffic representations. [15] proposed XTS, combining cost-sensitive XGBoost, Sequential Forward Evaluation and TreeSHAP to retain strong performance with a compact feature set. Their influential variables included packet-length characteristics and endpoint addresses, although the use of source and destination addresses limits direct comparison with address-independent flow analysis. [8] examined the problem from an operational perspective, combining standard telemetry, machine learning, resolver information and active probing to improve deployability. However, this multi-component approach differs from comparing individual classifier families under an identical feature representation. Overall, previous studies show a trade-off between statistical richness and operational efficiency, while feature importance remains dependent on the model, dataset and correlated predictors. Packet-length, timing and flow-volume variables have repeatedly supported DoH and malicious-tunnel classification, but no universally optimal feature set has been established [7], [13], [14], [15]. This supports the use of model-agnostic interpretation for comparing classifiers, while recognising that feature importance does not imply causality or guarantee stability across datasets.

4) Evaluation Reliability and Generalisability

Study by [9] reconstructed seven published DoH detectors and evaluated them across transferability, usability and performance longevity. Their results showed that strong within-environment performance could decline across datasets or network settings, while incomplete feature descriptions and poorly specified model configurations also

limited reproducibility. [16] reported a similar problem in broader intrusion detection, where strong within-dataset performance did not reliably transfer across datasets because of differences in features, labels and traffic distributions.

[17] further showed that variations in network conditions and TCP behaviour can alter encrypted-traffic representations and reduce classifier robustness, even when the underlying application class remains unchanged. Evaluation reliability also depends on partitioning strategy, because random record-level splits may place related flows from the same session or capture in both training and test sets, potentially producing optimistic estimates. These findings reinforce the distinction between within-dataset discrimination and broader generalisation. Repeated record-level evaluation remains useful for controlled comparison, but it cannot replace external, temporal or group-aware validation [9], [16], [17].

5) Critical Synthesis and Research Positioning

The literature clearly contains comparative DoH research, explainable detection systems and operational monitoring proposals. The present study therefore does not claim the first DoH comparison or introduce a novel classifier. Its contribution is narrower and methodological. Previous studies do not consistently compare a linear model, a bagging ensemble and a boosting ensemble on the same cleaned CIRA-CIC-DoHBrw-2020 Layer 1 records while jointly examining macro F1, MCC, Average Precision, false-positive behaviour, execution time and model-agnostic permutation importance. The study addresses this gap through a controlled within-dataset comparison of Logistic Regression, Random Forest and XGBoost. All three models are evaluated using a common statistical feature representation, partitioning procedure, imbalance treatment and test framework. This design isolates differences between representative model families without introducing heterogeneous detector architectures, endpoint-address intelligence or different datasets into the primary comparison. The resulting evidence should be interpreted as a reproducible benchmark rather than a solution to the generalisation problem identified by [9], [16], [17]. Its value lies in establishing a transparent model-level baseline and identifying predictive, computational and interpretive differences that can later be tested through group-aware, temporal, cross-dataset and operational evaluation.

Table 1. Synthesis of representative encrypted-DNS and DoH studies

Study	Primary focus	Methodological contribution	Limitation relative to the present study
Vekshin et al. [4]	DoH recognition	Demonstrated flow-based machine-learning detection	Limited cross-environment evaluation
MontazeriShatoori et al. [5]	DoH detection and characterisation	Introduced the two-layer CIRA-CIC-DoHBrw-2020 benchmark	Layer 1 and Layer 2 address different tasks
Alzighaibi [6]	Machine-learning and deep-learning comparison	Evaluated neural and conventional classifiers	Limited combined treatment of errors, cost and model-agnostic interpretation
Wala et al. [7]	Independent DoH classification	Confirmed packet and timing patterns across services and locations	Different collection and feature context
Jerabek et al. [8]	Operational detection	Combined standard telemetry, resolver intelligence and probing	Multi-component detector rather than isolated model comparison
Jerabek et al. [9]	Comparative detector evaluation	Examined transferability, usability and longevity	Compared heterogeneous detector architectures
Siby et al. [11]	Encrypted-DNS privacy inference	Demonstrated persistent statistical visibility	Domain inference rather than binary DoH detection
Bushart and Rossow [12]	Padding and traffic analysis	Showed that padding does not eliminate size and timing leakage	Focused on privacy inference
Irénée et al. [15]	Explainable feature reduction	Combined cost-sensitive XGBoost, feature selection and TreeSHAP	Included endpoint-address variables
D'hooge et al. [16]	Cross-dataset generalisation	Demonstrated deterioration across intrusion datasets	Broader intrusion detection rather than DoH

Study	Primary focus	Methodological contribution	Limitation relative to the present study
Behnke et al. [13]	Feature engineering for malicious DoH	Compared classifiers, feature selection and explainability	Focused on malicious activity detection
Xie et al. [17]	Robust TLS classification	Demonstrated sensitivity to network and TCP variation	Addressed general encrypted traffic
Moure-Garrido et al. [14]	Real-time malicious-tunnel detection	Evaluated lightweight packet-level detection	Different task and operational unit

3. METHODS

3.1. Research Design and Classification Task

The experimental process comprised dataset verification, stratified data partitioning, training-data-based preprocessing, cross-validation, final testing and post-hoc model interpretation. Figure 1 presents the complete operational workflow. Following dataset and feature verification, the data were divided into an 80% development partition and a 20% held-out test partition using stratification. Preprocessing parameters and class-imbalance adjustments were derived exclusively from the development data. Three-fold stratified cross-validation was then performed within the development partition to assess each fixed model configuration. After cross-validation, each classifier was refitted using the complete development partition and evaluated once on the corresponding untouched test partition. Permutation importance was subsequently calculated using held-out observations, after which the results were aggregated across the three fixed random seeds.

3.2. Dataset and Feature Preparation

The study used the Layer 1 component of the CIRA-CIC-DoHBrw-2020 dataset. The original dataset follows a two-layer structure. Layer 1 distinguishes DoH from non-DoH traffic, while Layer 2 differentiates benign DoH from malicious DoH traffic [5]. Only Layer 1 was selected because the present study concerns the detection of DoH within encrypted HTTPS traffic rather than the subsequent characterisation of malicious DoH activity. The analysed file was the cleaned parquet representation, L1-DoH-NonDoH.parquet. The official dataset documentation confirms the Layer 1 and Layer 2

distinction and describes the browser, resolver and tunnelling environments used to generate the original traffic [18].

Inspection of the parquet file identified 1,019,117 flow records, comprising 750,456 non-DoH observations, representing 73.6% of the dataset, and 268,661 DoH observations, representing 26.4%. Each observation contained 29 numerical predictors and one target variable, Label. The predictors represented flow duration, transmitted and received byte volumes and rates, packet-length statistics, packet-time statistics and request-response-time statistics. The dataset contained no missing values, infinite numerical values, exact duplicate records, non-numeric predictors, constant features or near-constant features in which a single value accounted for at least 99.9% of the observations. Consequently, no imputation, duplicate removal or feature deletion was required. The target variable was encoded as 1 for DoH and 0 for non-DoH. The dataset inspection also generated a SHA-256 file hash to preserve the identity of the exact file used in the experiments.

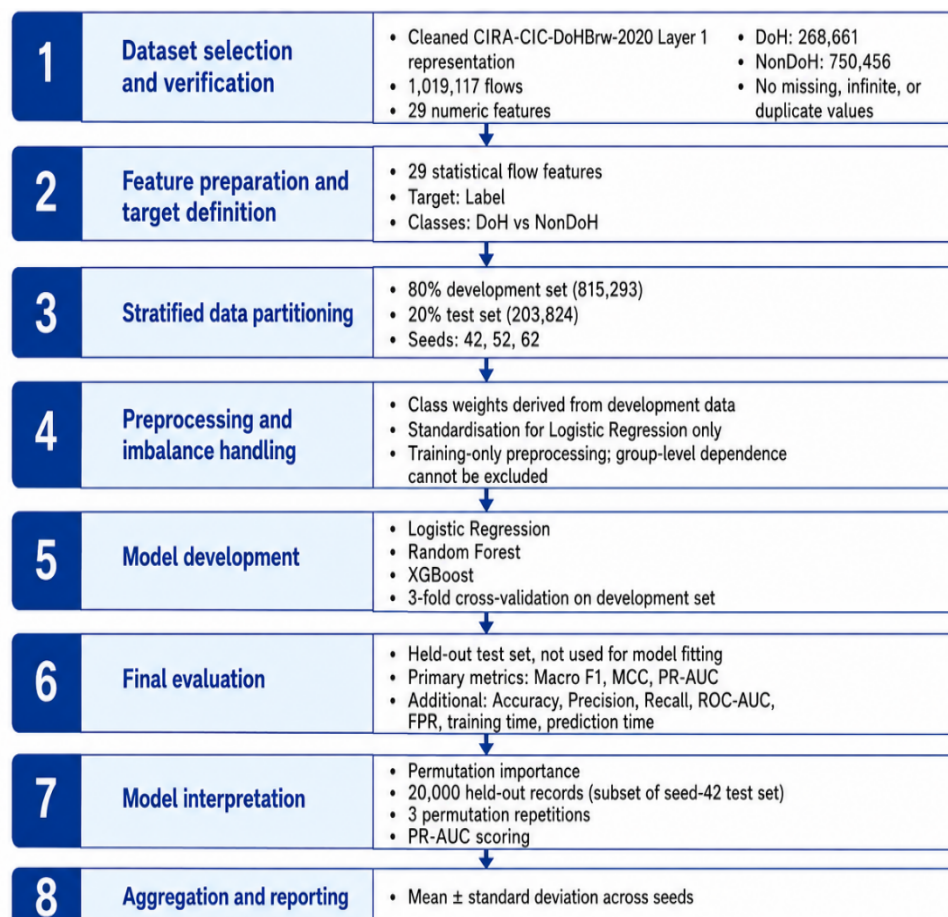


Figure 1. Experimental methodology for DoH versus non-DoH traffic classification

3.3. Data Partitioning and Preprocessing

The experiment was repeated independently using the fixed random seeds 42, 52 and 62. For each seed, the data were randomly partitioned using an 80:20 stratified split. This allocated 815,293 observations to the development set and 203,824 observations to the untouched test set while maintaining the original DoH and non-DoH proportions.

Within each development partition, stratified three-fold cross-validation was performed to estimate the performance and stability of each fixed model configuration. The test partition was not used during cross-validation, preprocessing estimation, model fitting or class-weight calculation. After cross-validation, each classifier was fitted using the complete development partition and evaluated once on the corresponding untouched test partition. The three-seed design therefore produced three repeated stratified hold-out evaluations using fixed random seeds. The executed protocol retained the complete dataset rather than using a reduced sample.

Class imbalance was addressed without removing majority-class observations or creating synthetic samples. Logistic Regression used inverse-frequency balanced class weights. Random Forest used *balanced_{subsample}* weighting, under which class weights were recalculated within each bootstrap sample. For XGBoost, $Scale_{pos_{weight}}$ was calculated separately from each development partition as:

$$Scale_{pos_{weight}} = \frac{N_{NonDoH}}{N_{DoH}} \quad (1)$$

Where N_{NonDoH} and N_{DoH} denote the number of negative and positive observations in the relevant development partition, respectively.

Standardisation was applied only to Logistic Regression because its estimated coefficients are sensitive to differences in feature scale. A *StandardScaler* was placed within the Logistic Regression pipeline so that the feature means, and standard deviations were estimated only from the relevant training fold or full development partition. The fitted transformation was subsequently applied to the corresponding validation or test observations. Random Forest and XGBoost received the predictors on their original numerical scales because tree-based split selection does not require standardised variables.

The cleaned dataset did not contain capture identifiers, user identifiers, session identifiers, timestamps, IP addresses or resolver identifiers. A group-aware partition based on capture, user or browsing session could therefore not be constructed. The record-level stratified split was necessary, although potentially related flows may have occurred in different partitions. This was treated as a threat to internal validity rather than being addressed through an unsupported grouping assumption.

3.4. Model Development and Configuration

The evaluated classifiers were selected to provide a comparison between a linear decision boundary and two established nonlinear ensemble methods. Random Forest combines predictions from multiple randomly constructed decision trees [19]. Whereas XGBoost applies regularised sequential tree boosting and includes computational mechanisms for efficient large-scale learning [20]. The configurations were fixed before final testing and applied consistently across the three repetitions. No result from the untouched test partitions was used to modify the settings. Table 2 shows the executed model configurations for Logistics Regression, Random Forest and XGBoost.

Table 2. Executed model configurations

Model	Configuration
Logistic Regression	(C=1.0); lbfgs solver; maximum 1,000 iterations; balanced class weights; standardisation within the modelling pipeline
Random Forest	100 trees; Gini criterion; unrestricted maximum depth; minimum split size of 2; minimum leaf size of 1; square-root feature sampling; bootstrap sampling; balanced subsample class weights
XGBoost	200 boosting rounds; learning rate of 0.05; maximum depth of 5; minimum child weight of 1; row subsampling of 0.8; column subsampling of 0.8; gamma of 0; L1 regularisation of 0; L2 regularisation of 1; binary logistic objective; log-loss evaluation; histogram tree method; development-derived $Scale_{posweight}$

For each seed and model, three-fold cross-validation was first conducted using the development data. The estimator was then refitted on the full development partition and evaluated on the untouched test partition. The random seed controlled the data split and, where supported by the estimator, model randomness. Parallel processing was enabled for Random Forest and XGBoost to reduce execution time on the complete dataset. These

settings match the executed experiment rather than the earlier five-seed and five-fold proposal.

3.5. Evaluation and Model Interpretation

Model comparison emphasised measures appropriate for unequal class frequencies. Macro F1, Matthew's correlation coefficient (MCC) and Average Precision (AP) were designated as the primary evaluation measures. For class c , the F1-score was calculated from precision P_c and recall R_c as:

$$F1_c = \frac{2P_cR_c}{P_c+R_c} \quad (2)$$

Macro F1 was subsequently calculated as the unweighted mean of the class-specific F1-scores:

$$\text{Macro F1} = \frac{F1_{\text{DoH}} + F1_{\text{nonDoH}}}{2} \quad (3)$$

This formulation prevented the majority non-DoH class from dominating the combined score. MCC was included because it incorporates all four outcomes of the confusion matrix and remains informative when class frequencies differ [21]:

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (4)$$

where TP , TN , FP and FN denote true positives, true negatives, false positives and false negatives, respectively. Precision-recall performance was summarised using Average Precision (AP). Inspection of the executed experimental code confirmed that AP was calculated from the predicted probabilities using the average-precision implementation rather than trapezoidal integration of the precision-recall curve. AP can be expressed as:

$$AP = \sum_n (R_n - R_{n-1})P_n \quad (5)$$

where P_n and R_n denote precision and recall at successive operating points. AP was prioritised because it summarises positive-class ranking performance while giving particular attention to precision and recall under unequal class frequencies [22].

The additional evaluation measures were accuracy, balanced accuracy, DoH precision, DoH recall, area under the receiver operating characteristic curve (ROC-AUC) and false-positive rate. Training time and prediction time were also recorded to compare computational efficiency. Threshold-dependent measures used a probability threshold of 0.5, whereas AP and ROC-AUC were calculated from predicted probabilities. A confusion matrix was retained separately for every model and seed.

Test results were summarised across the three repeated evaluations using the arithmetic mean, sample standard deviation and a 95% confidence interval based on the t-distribution. Because the three test partitions were drawn from the same dataset and overlapped, these intervals represent descriptive uncertainty across the selected random record-level splits rather than inferential confidence based on three independent datasets. Individual seed results are therefore reported alongside the aggregated values to show stability under the specified partitioning procedure.

Permutation importance was used to examine the dependence of the fitted classifiers on individual predictor variables. The fitted seed-42 version of each classifier was evaluated using the same stratified sample of 20,000 observations drawn from its untouched test partition. Each feature was randomly permuted three times while the remaining predictors were left unchanged. Average Precision was used as the permutation scoring criterion. The resulting decrease in AP was recorded as the feature-importance value, with a larger reduction indicating greater dependence of the fitted classifier on that predictor for maintaining precision-recall ranking performance. Permutation importance was used for comparative interpretation rather than causal explanation. The resulting importance values were interpreted cautiously because correlated flow statistics may contain overlapping predictive information. Consequently, importance may be concentrated in one predictor or distributed across several related predictors when they provide similar information.

3.6. Implementation and Reproducibility

The experiment was implemented in Python. Pandas and NumPy supported data loading and numerical processing, while scikit-learn was used for stratified partitioning, preprocessing pipelines, Logistic Regression, Random Forest, cross-validation, evaluation metrics and permutation importance [23]. XGBoost was used to implement the gradient-

boosting classifier [20]. SciPy supported statistical summaries and confidence-interval estimation, while Matplotlib was used to generate the graphical outputs.

Before model fitting, the experimental programme automatically verified the expected dataset dimensions, predictor count and class distribution. It also checked for missing values, infinite numerical values, exact duplicate records, non-numeric predictors, constant predictors and near-constant predictors. The analysed dataset contained 1,019,117 records, 29 predictors, 750,456 non-DoH observations and 268,661 DoH observations. No missing cells, infinite numerical values or exact duplicate records were identified, and the dataset passed all programmed validation checks. A SHA-256 hash was also generated to preserve the identity of the exact parquet file used in the experiment. Reproducibility was supported through fixed random seeds, explicit classifier configurations, stratified partitioning, training-only preprocessing and class-weight estimation, a recorded dataset hash and saved experimental outputs. The retained outputs included fold-level cross-validation metrics, per-seed held-out test metrics, aggregated performance summaries, confidence intervals, confusion matrices, permutation-importance values, training times and prediction times.

The experimental programme was executed in a Google Colab cloud-based environment running Linux 6.6.122+ on an x86_64 architecture. The allocated runtime used an Intel(R) Xeon(R) CPU operating at 2.20 GHz, with one physical CPU core, two logical processors and 12.67 GB of RAM. The software environment included Python 3.12, scikit-learn 1.6.1 and XGBoost 3.4.1. Parallel processing was enabled for Random Forest and XGBoost where supported.

The cleaned dataset did not contain capture identifiers, user identifiers, browsing-session identifiers, timestamps, IP addresses or resolver identifiers. A group-aware or temporal partition could therefore not be constructed from the available representation. The record-level stratified evaluation was retained as the reproducible comparison protocol, while the possibility that statistically related flows may have occurred across development and test partitions is treated explicitly as a threat to validity in the Discussion.

4. RESULTS AND DISCUSSION

This section presents the predictive, computational and interpretive results obtained from the three repeated stratified hold-out evaluations using seeds 42, 52 and 62. Unless otherwise stated, results are reported as the mean and standard deviation across the three repetitions. To avoid repetition, the section is organised into predictive performance and stability, computational performance, and permutation-based feature importance.

4.1. Predictive Performance and Stability

Table 3 summarises the cross-validation and held-out test performance of the three classifiers. The cross-validation and test results were closely aligned for all models, indicating limited variation across the three specified splits. Logistic Regression produced a cross-validation macro F1 of 0.8524 and a test macro F1 of 0.8524. The corresponding values were 0.9964 and 0.9965 for Random Forest, and 0.9949 and 0.9952 for XGBoost. The consistency between validation and test performance shows that the relative ranking of the models was maintained across the development and held-out evaluation stages. These small standard deviations indicate consistency across the three selected record-level partitions but should not be interpreted as evidence of session-level, temporal, cross-network or cross-dataset robustness. Random Forest achieved the strongest threshold-dependent performance. It obtained the highest mean test accuracy (0.9973), balanced accuracy (0.9958), macro F1 (0.9965) and MCC (0.9930). XGBoost followed closely, with an accuracy of 0.9962, balanced accuracy of 0.9948, macro F1 of 0.9952 and MCC of 0.9903. Logistic Regression was substantially weaker, producing an accuracy of 0.8782, macro F1 of 0.8524 and MCC of 0.7122.

A slightly different pattern appeared in the ranking-based measures. XGBoost achieved the highest AP (0.9994) and ROC-AUC (0.9997), while Random Forest recorded 0.9989 and 0.9993, respectively. Logistic Regression remained clearly lower, with an AP of 0.8856 and ROC-AUC of 0.9485. Accordingly, Random Forest was the strongest model at the fixed 0.5 classification threshold, whereas XGBoost provided the best ranking performance across alternative thresholds.

Class-specific results further separated the models. Random Forest achieved DoH precision of 0.9970 and recall of 0.9927, while XGBoost achieved precision of 0.9939 and recall of 0.9918. Logistic Regression retained moderate DoH recall (0.8726) but lower precision (0.7229). The same pattern was reflected in false-positive behaviour. Random Forest produced the lowest mean false-positive rate (0.0011), followed by XGBoost (0.0022), whereas Logistic Regression produced a substantially higher value (0.1197).

Table 3. Cross-validation and held-out test performance (mean $\pm$ standard deviation)

Model	CV macro F1	Test accuracy	Balanced accuracy	Test macro F1	MCC	AP	ROC- AUC
Logistic Regression	0.8524 $\pm$ 0.9964 $\pm$ 0.0001	0.8782 $\pm$ 0.9973 $\pm$ 0.0000	0.8764 $\pm$ 0.9958 $\pm$ 0.0001	0.8524 $\pm$ 0.9965 $\pm$ 0.0000	0.7122 $\pm$ 0.9930 $\pm$ 0.0001	0.8856 $\pm$ 0.0015 0.9989 $\pm$ 0.0000	0.9485 $\pm$ 0.9993 $\pm$ 0.0000
Random Forest	0.9949 $\pm$ 0.0002	0.9962 $\pm$ 0.0002	0.9948 $\pm$ 0.0002	0.9952 $\pm$ 0.0003	0.9903 $\pm$ 0.0005	0.9994 $\pm$ 0.0001	0.9997 $\pm$ 0.0000

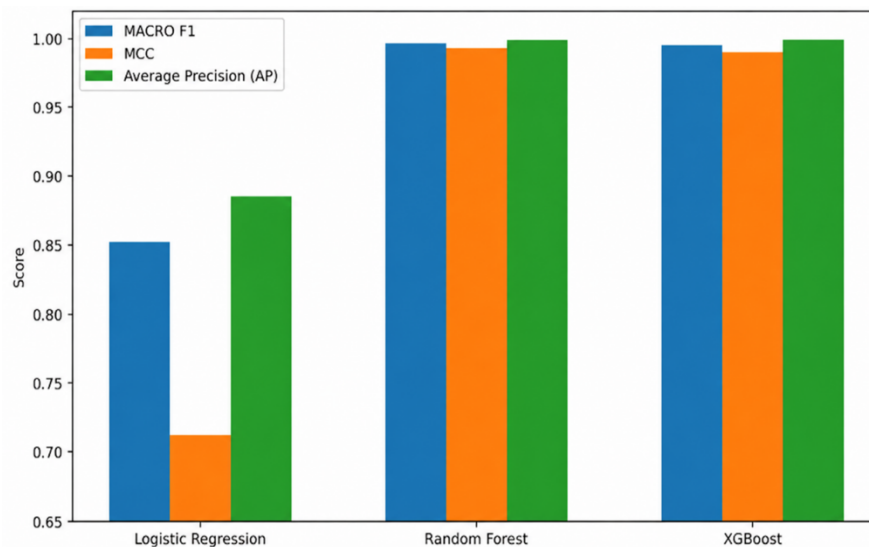


Figure 2. Comparison of Macro F1, MCC and AP across the evaluated models

The error counts reinforce this comparison. Random Forest generated a mean of 161 false positives and 392 false negatives per run. XGBoost produced 325 false positives and 439

false negatives, while Logistic Regression produced 17,970 false positives and 6,845 false negatives. The standard deviations of the principal measures were small for all models, especially for Random Forest and XGBoost, indicating only limited variation across the three repeated splits. Figure 2 compares the principal performance measures used for model selection. Random Forest achieved the highest macro F1 and MCC, while XGBoost produced the highest AP.

4.2. Computational Performance

Table 4 reports the mean false-positive and false-negative counts together with the measured execution times. Logistic Regression was the fastest classifier, with a mean training time of 23.35 seconds and prediction time of 0.15 seconds. XGBoost required 31.40 seconds for training and 2.00 seconds for prediction. Random Forest required substantially greater computation, with a mean training time of 743.42 seconds and prediction time of 4.61 seconds. Although Random Forest produced the strongest threshold-dependent classification performance, XGBoost remained close in predictive performance while requiring substantially less training time. Specifically, XGBoost required approximately 95.8% less mean training time than Random Forest. Logistic Regression remained computationally inexpensive but substantially weaker in predictive quality. The timing results should be interpreted as measurements from the experimental environment rather than as hardware-independent benchmarks.

Table 4. Error and computational summary (mean $\pm$ standard deviation)

Model	DoH precision	DoH recall	FPR	Mean FP	Mean FN	Training time, s	Prediction time, s
Logistic Regression	0.7229 $\pm$ 0.0010	0.8726 $\pm$ 0.0012	0.1197 $\pm$ 0.0006	17,970	6,845	23.35 $\pm$ 1.83	0.15 $\pm$ 0.04
Random Forest	0.9970 $\pm$ 0.0003	0.9927 $\pm$ 0.0002	0.0011 $\pm$ 0.0001	161	392	743.42 $\pm$ 8.35	4.61 $\pm$ 0.53
XGBoost	0.9939 $\pm$ 0.0006	0.9918 $\pm$ 0.0002	0.0022 $\pm$ 0.0002	325	439	31.40 $\pm$ 0.71	2.00 $\pm$ 0.68

Figure 3 shows the mean false-positive and false-negative counts across the three repeated test evaluations. Logistic Regression generated substantially more false

positives than the ensemble models, while Random Forest produced the lowest classification error counts overall.

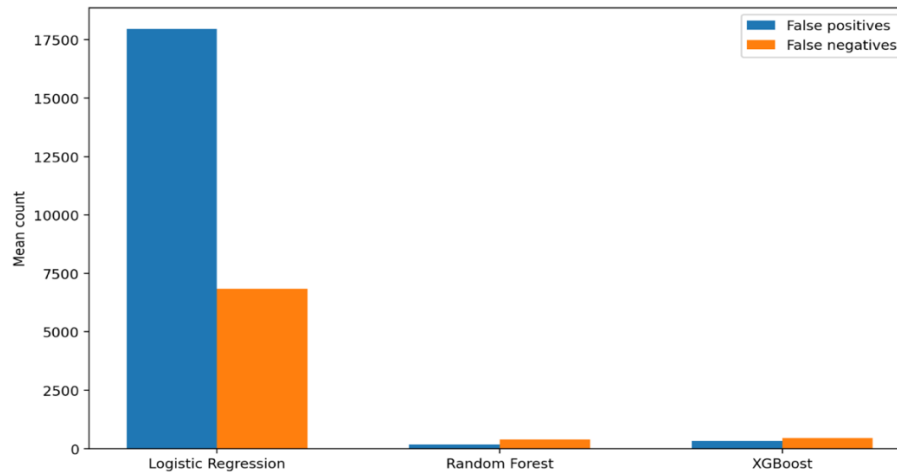


Figure 3. Mean false positives and false negatives by model

Figure 4 compares the computational cost of the three classifiers. Random Forest required the longest training time, whereas XGBoost provided a substantially faster alternative with only a small reduction in threshold-dependent performance.

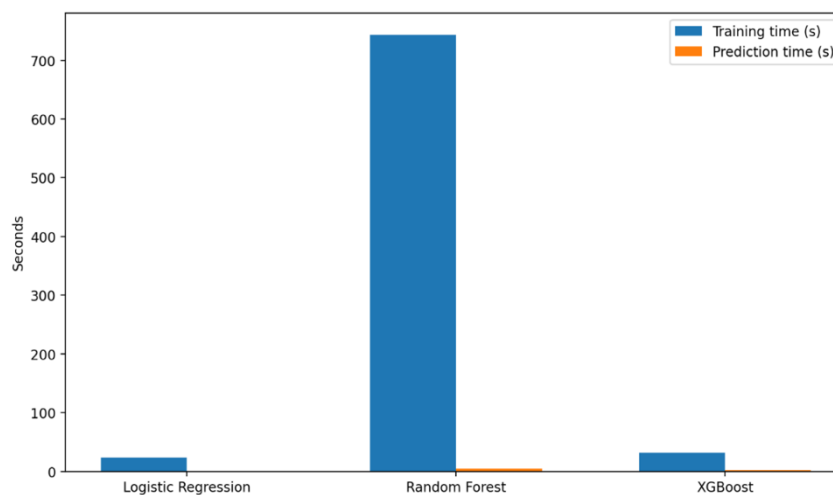


Figure 4. Mean training and prediction time by model

4.3. Permutation Feature Importance

Permutation importance was calculated as the reduction in held-out Average Precision (AP) after an individual predictor was randomly permuted. The procedure was applied to the seed-42 fitted models using a stratified held-out sample of 20,000 observations and

three permutation repetitions. For Logistic Regression, the most influential variables were PacketLengthVariance (0.5058), Duration (0.5052), PacketLengthStandardDeviation (0.4632) and PacketLengthMean (0.3640). FlowBytesSent also contributed meaningfully, with an importance value of 0.1774. This pattern shows that the linear classifier depended on a broader combination of duration, packet-length and flow-volume characteristics. For the tree-based models, PacketLengthMode was the dominant predictor. Its importance was 0.2879 for Random Forest and 0.3157 for XGBoost. Other contributing variables included PacketLengthMean, PacketLengthVariance, PacketLengthStandardDeviation, PacketLengthMedian, FlowBytesSent, FlowBytesReceived and selected response-time characteristics. Overall, the nonlinear models concentrated their predictive dependence more strongly on a smaller set of packet-length characteristics, whereas Logistic Regression showed a more distributed importance profile. These values represent the reduction in AP following feature permutation and therefore describe the dependence of the fitted classifier on individual predictors under the specified interpretation procedure. They do not establish causal relationships between individual flow characteristics and DoH traffic. Figure 5 presents the leading permutation-importance features using the reduction in AP as the scoring criterion. PacketLengthMode dominated the two tree-based models, whereas Logistic Regression relied more broadly on packet-length, duration and flow-volume measures.

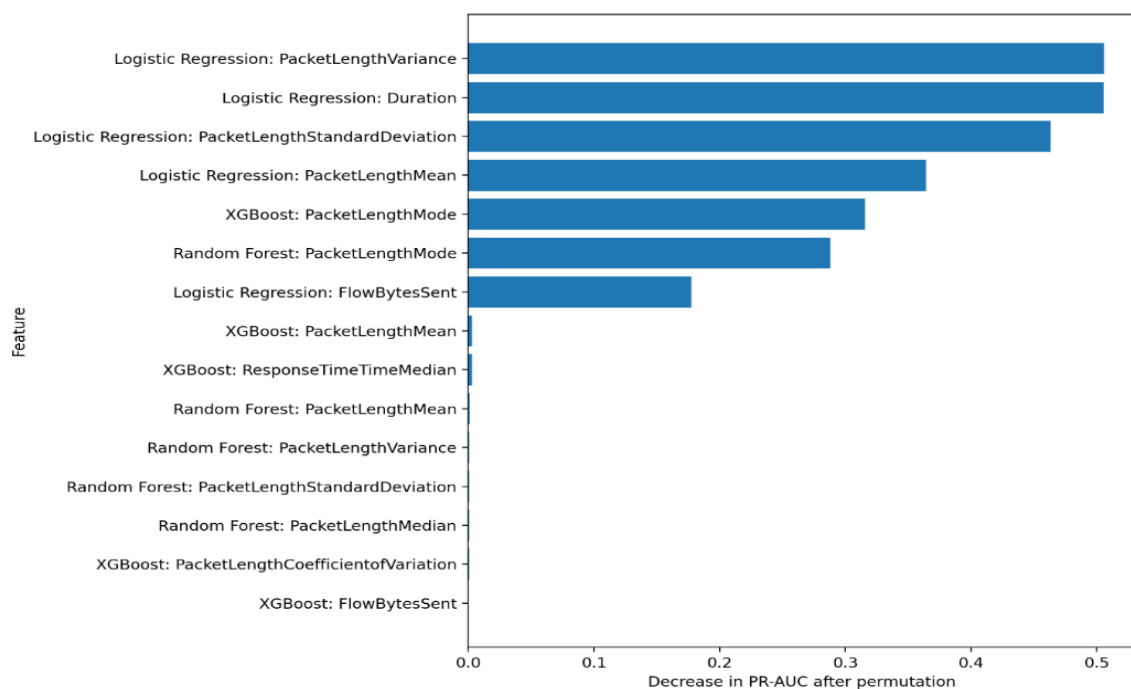


Figure 5. Top permutation-importance features based on reduction in Average Precision

4.4. Discussion

1) Predictive and Computational Trade-offs

The findings address RQ1 by demonstrating that the two tree-based classifiers distinguished DoH from non-DoH HTTPS flows substantially more effectively than the Logistic Regression baseline. Random Forest achieved the strongest threshold-dependent performance, including the highest macro F1 and MCC and the lowest false-positive rate. XGBoost followed closely and achieved the highest Average Precision (AP) and ROC-AUC. Logistic Regression was computationally inexpensive but produced markedly weaker classification performance, particularly in terms of DoH precision and false-positive control.

The magnitude of the performance difference is consistent with nonlinear structure in the evaluated statistical feature space. Logistic Regression represents the relationship between the predictors and class outcome through an additive linear decision function after feature standardisation. Consequently, complex class boundaries must be represented through a single weighted combination of duration, packet-length, timing and flow-volume characteristics. Random Forest and XGBoost, by contrast, recursively partition the predictor space and can represent conditional relationships in which the discriminative value of one flow characteristic depends on the value of another. For example, a packet-length characteristic may become particularly informative only within specific ranges of flow duration, transmitted volume or timing behaviour. Tree-based classifiers can model such conditional interactions directly through successive decision splits, whereas they are not represented explicitly by the linear baseline. The superior performance of the ensemble models is therefore compatible with nonlinear and interaction-dependent separation between DoH and non-DoH flows within the evaluated dataset.

This interpretation should nevertheless remain cautious. The experiment did not directly quantify individual feature interactions or compare fitted decision-boundary geometry. The results therefore demonstrate that the two ensemble classifiers extracted substantially more discriminative information from the available predictors, rather than proving that any specific nonlinear interaction caused the observed performance advantage. The finding is broadly consistent with previous investigations using CIRA-CIC-

DoHBrw-2020, which showed that statistical and time-series characteristics can support strong DoH classification [5].

The findings also answer RQ2 by showing that model ranking depended on the evaluation criterion. Random Forest was strongest after predicted probabilities were converted into class labels using the fixed threshold of 0.5, whereas XGBoost achieved the highest AP and ROC-AUC. This distinction is operationally relevant because threshold-dependent measures describe performance under one selected decision rule, while AP evaluates precision-recall ranking performance across operating points. Random Forest would therefore be favoured within the present experiment where minimising classification errors at the selected threshold is the principal objective, whereas XGBoost may offer greater flexibility where decision thresholds are subsequently adjusted according to alert volume, error costs or risk tolerance.

The false-positive differences have direct implications for security-monitoring workload. Logistic Regression produced a mean false-positive rate of 11.97%, corresponding to approximately 17,970 non-DoH observations incorrectly classified as DoH in each held-out test evaluation, compared with 161 for Random Forest and 325 for XGBoost. In a high-volume Security Operations Centre (SOC), a detector generating false alerts at this scale could substantially increase the number of events requiring triage and investigation. Repeated false alarms may increase analyst workload, reduce confidence in automated detection and contribute to alert fatigue, potentially diverting attention from higher-priority events. False-positive control is therefore an important operational consideration in addition to overall classification accuracy.

Computational cost further distinguishes the two ensemble models. Random Forest required a mean training time of 743.42 seconds, compared with 31.40 seconds for XGBoost, meaning that XGBoost required approximately 95.8% less mean training time while retaining closely comparable threshold-dependent performance and achieving the highest AP. Random Forest nevertheless produced the lower false-positive rate and fewer classification errors at the 0.5 threshold. The results therefore indicate a trade-off between fixed-threshold error control and computational cost rather than a universally superior classifier. These timing measurements remain specific to the

executed environment and model configurations and should not be interpreted as hardware-independent benchmarks.

2) Interpretation of Influential Flow Characteristics

The permutation-importance findings address RQ3 by identifying packet-length characteristics as the principal contributors to the nonlinear classifiers. PacketLengthMode was the dominant predictor for both Random Forest and XGBoost, producing AP reductions of 0.2879 and 0.3157 respectively when permuted. PacketLengthMean, PacketLengthVariance, PacketLengthStandardDeviation, PacketLengthMedian, flow-volume characteristics and selected response-time measures provided additional predictive information. Logistic Regression showed a different pattern, with greater dependence distributed across PacketLengthVariance, Duration, PacketLengthStandardDeviation, PacketLengthMean and FlowBytesSent.

The prominence of PacketLengthMode in the tree-based models is plausible because it represents the most frequently occurring packet length within an observed flow. Repeated packet-size patterns may provide concentrated splitting points that distinguish recurring characteristics of DoH and conventional HTTPS flows within the benchmark. Random Forest and XGBoost can construct successive conditional splits around such values and combine them with other characteristics such as duration, flow volume and timing. A predictor that provides an effective early or repeated partition of the feature space can consequently become highly influential in a fitted tree ensemble.

Logistic Regression cannot construct comparable conditional partitions. Its prediction is based on the weighted additive contribution of the scaled predictors. Consequently, information represented by related packet-length characteristics may be distributed across PacketLengthVariance, PacketLengthStandardDeviation, PacketLengthMean and other variables rather than being concentrated predominantly in one predictor. This provides a plausible explanation for the broader permutation-importance profile observed for Logistic Regression.

These interpretations should not be treated as evidence that PacketLengthMode is intrinsically or causally responsible for DoH behaviour. Permutation importance measures the dependence of a fitted model on a predictor under the specified evaluation

procedure. Correlated packet-length characteristics may contain overlapping information, allowing one variable to substitute partly for another. Importance may therefore be concentrated in one predictor for one classifier and distributed across several correlated predictors for another.

The broader observation that packet size and timing characteristics remain informative despite encryption is consistent with previous encrypted-DNS research [7], [12], [13], [15]. Encryption protects payload contents but does not necessarily eliminate observable traffic metadata such as packet sizes, timing patterns and transferred volume. The present results extend that observation within the specific controlled Layer 1 benchmark rather than establishing a universally stable feature signature.

3) Implications for DoH Monitoring

The findings demonstrate that the evaluated models distinguished DoH from non-DoH HTTPS flows within the CIRA-CIC-DoHBrw-2020 Layer 1 benchmark using statistical flow characteristics without relying on payload contents, domain names, SNI values, JA3 fingerprints, IP addresses or resolver identities. This supports further investigation of statistical-flow classification in environments where TLS payload inspection is unavailable, restricted or undesirable.

From an enterprise-monitoring perspective, statistical classification could potentially operate as a complementary analytical layer alongside conventional flow-telemetry infrastructure. Characteristics such as flow duration, transferred byte volume, packet-size distributions and timing behaviour are conceptually compatible with telemetry-oriented monitoring in which flow metadata is analysed without decrypting application payloads. NetFlow/IPFIX-style pipelines may therefore provide a practical integration point where equivalent statistical characteristics can be collected or derived. However, the present experiment did not implement a production NetFlow/IPFIX pipeline, and operational deployment would require verification that the required features can be reproduced consistently from the organisation's telemetry infrastructure. Although flow-based monitoring avoids decrypting TLS payloads, the resulting metadata can still reveal behavioural patterns. Its collection, retention and analysis should therefore be governed by proportionate privacy, access-control and data-retention policies.

The scope of such monitoring must also be distinguished from malicious-activity detection. Layer 1 identifies whether the observed flow resembles DoH or non-DoH HTTPS traffic. It does not establish whether detected DoH traffic is benign, malicious or contrary to organisational policy. A production workflow should therefore treat DoH identification as an initial classification or prioritisation stage rather than automatically interpret every detected DoH flow as a security incident. Additional policy, contextual or Layer 2 analysis would be required before escalation [5]. Organisational security policy should therefore define whether identified DoH activity is merely observed, subjected to further investigation, blocked or escalated according to the relevant network context and risk level.

Within the present benchmark, Random Forest offered the strongest fixed-threshold error control, whereas XGBoost offered a more favourable relationship between predictive performance and retraining cost. XGBoost may therefore be an attractive candidate for further evaluation in environments where model updating or repeated retraining is operationally important. However, the current timing measurements were obtained through batch evaluation of the held-out test partition and do not constitute a dedicated measurement of single-flow latency on an edge router or production network appliance. Claims of sub-millisecond per-flow inference or suitability for high-throughput edge deployment would therefore require separate throughput, latency, processor utilisation and memory profiling under the intended production architecture.

Deployment decisions should consequently consider false-positive tolerance, threshold selection, retraining frequency, available telemetry, processing capacity and privacy requirements rather than predictive accuracy alone. External validation, probability calibration, memory profiling, latency testing and evaluation using contemporary traffic from the intended organisational environment would be required before either Random Forest or XGBoost could be recommended for production deployment.

4) Threats to Validity and Technical Limitations

The principal threat to internal validity arises from the use of stratified record-level partitioning. The cleaned parquet representation does not contain capture identifiers, browser-session identifiers, user identifiers, resolver identifiers or timestamps. Related flows could therefore not be assigned to the same development or test partition using

group-aware or temporal splitting. Statistically similar flows originating from a common underlying browsing activity or capture context may consequently have occurred in different partitions, potentially producing optimistic estimates if they share highly similar statistical characteristics. The small variation observed across seeds 42, 52 and 62 demonstrates consistency under repeated record-level partitioning only and does not establish robustness across users, sessions, capture periods or network environments.

A second limitation concerns dataset provenance and external validity. The experiment used the cleaned parquet representation of CIRA-CIC-DoHBrw-2020 rather than independently reconstructing Layer 1 directly from the original packet captures. The exact file used in the experiment was verified through its dimensions, class counts, predictor types, integrity checks and SHA-256 hash, but the study did not independently reproduce every transformation used to create the cleaned representation. Evaluation was also restricted to one benchmark dataset. Strong within-dataset discrimination should therefore not be interpreted as evidence that equivalent performance would occur in other organisations, resolver environments, browser populations, geographic locations or contemporary live-network traffic [16]. The age and controlled nature of the benchmark also limit its contemporary representativeness. CIRA-CIC-DoHBrw-2020 may not capture changes in modern browser and resolver implementations, HTTP/3 and QUIC-based encrypted traffic, contemporary traffic-shaping practices or current enterprise network conditions.

The experimental design also used three fixed random seeds and fixed classifier configurations rather than exhaustive hyperparameter optimisation. No paired statistical significance test was performed between Random Forest and XGBoost, and probability calibration, threshold optimisation and feature ablation were outside the executed protocol. Permutation importance was calculated using the seed-42 fitted models and a 20,000-observation held-out sample rather than repeated across all seeds. Consequently, the reported ranking characterises feature dependence for the seed-42 models and selected held-out sample; it does not establish that the ranking is stable across random seeds, alternative samples or external datasets. Its results therefore describe feature dependence under that specific interpretation procedure rather than feature-ranking stability across all fitted models.

Computational measurements present an additional limitation. Training and prediction times depend on hardware, software versions, parallelism, batch size and implementation details and should not be interpreted as universal algorithmic benchmarks. The experiment did not directly measure memory consumption, per-flow inference latency, telemetry-export overhead or sustained throughput in a production router, collector or SOC pipeline. Consequently, the strong ensemble results demonstrate benchmark separability rather than operational deployment readiness [24].

5) Threats to Validity and Technical Limitations

The Future research should evaluate the classifiers using datasets that preserve capture, browsing-session and temporal identifiers so that group-aware and time-separated partitioning can be implemented. Cross-dataset experiments should determine whether the identified packet-length and timing characteristics transfer to different browser and resolver combinations, geographic locations, organisational networks and contemporary DoH implementations. Further work should also examine probability calibration, decision-threshold optimisation under unequal error costs, paired model comparisons, feature-importance stability across repeated fits and reduced-feature configurations. From an operational perspective, future evaluation should measure memory consumption, sustained throughput and single-flow inference latency within NetFlow/IPFIX-style monitoring pipelines and assess resilience to padding, traffic shaping and other mechanisms that may alter observable packet-length or timing characteristics. These extensions would help determine whether the strong within-benchmark results reflect transferable DoH behaviour or characteristics specific to the evaluated dataset and experimental protocol.

5. CONCLUSION

This study compared Logistic Regression, Random Forest and XGBoost for distinguishing DNS-over-HTTPS traffic from non-DoH HTTPS traffic using the cleaned Layer 1 representation of the CIRA-CIC-DoHBrw-2020 dataset containing 1,019,117 flow records and 29 statistical predictors. The results showed that the tree-based classifiers substantially outperformed the Logistic Regression baseline, with Random Forest achieving the strongest threshold-dependent performance through a macro F1 of 0.9965,

an MCC of 0.9930 and the lowest false-positive rate of 0.0011, while XGBoost achieved the highest Average Precision of 0.9994 and ROC-AUC of 0.9997 while requiring substantially less training time than Random Forest. Logistic Regression remained computationally efficient but produced lower DoH precision and substantially more false positives. The permutation-importance analysis further showed that packet-length characteristics were central to the nonlinear classifiers, with PacketLengthMode contributing most strongly to Random Forest and XGBoost, whereas Logistic Regression depended on a broader combination of packet-length variability, duration and flow-volume characteristics. These findings demonstrate that statistical flow characteristics can support strong separation of DoH and non-DoH HTTPS traffic within the evaluated benchmark without relying on payload inspection, domain names, IP addresses, resolver identities or JA3 fingerprints. From an operational perspective, Random Forest provided the strongest error control at the selected threshold, whereas XGBoost offered a more favourable balance between predictive performance and training cost, indicating the potential value of statistical-flow classification as a complementary monitoring approach where encrypted payload inspection is unavailable or undesirable. However, the findings should not be interpreted as evidence of deployment readiness or cross-network generalisation because the absence of capture, session, user and temporal identifiers prevented group-aware or temporal partitioning, and evaluation was restricted to a single benchmark dataset. Future research should therefore prioritise group-aware and temporal evaluation, cross-dataset testing, probability calibration, threshold optimisation, feature-stability analysis and validation using contemporary organisational or live-network traffic.

REFERENCES

- [1] P. Hoffman and P. McManus, 'DNS Queries over HTTPS (DoH)', RFC Editor, RFC8484, Oct. 2018. doi: 10.17487/RFC8484.
- [2] K. Hynek, D. Vekshin, J. Luxemburk, T. Cejka, and A. Wasicek, 'Summary of DNS Over HTTPS Abuse', *IEEE Access*, vol. 10, pp. 54668–54680, 2022, doi: 10.1109/ACCESS.2022.3175497.

- [3] Y. Li *et al.*, 'Temporal Association-Aware Dual-Sequence Modeling for Robust DoH Tunnel Detection', in *2026 29th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, Fuzhou, China: IEEE, May 2026, pp. 2375–2380. doi: 10.1109/CSCWD68734.2026.11582286.
- [4] D. Vekshin, K. Hynek, and T. Cejka, 'DoH Insight: detecting DNS over HTTPS by machine learning', in *Proceedings of the 15th International Conference on Availability, Reliability and Security*, Virtual Event Ireland: ACM, Aug. 2020, pp. 1–8. doi: 10.1145/3407023.3409192.
- [5] M. MontazeriShatoori, L. Davidson, G. Kaur, and A. Habibi Lashkari, 'Detection of DoH Tunnels using Time-series Classification of Encrypted Traffic', in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCoM/CyberSciTech)*, Calgary, AB, Canada: IEEE, Aug. 2020, pp. 63–70. doi: 10.1109/DASC-PICom-CBDCoM-CyberSciTech49142.2020.00026.
- [6] A. R. Alzighaibi, 'Detection of DoH Traffic Tunnels Using Deep Learning for Encrypted Traffic Classification', *Computers*, vol. 12, no. 3, p. 47, Feb. 2023, doi: 10.3390/computers12030047.
- [7] F. Bannat Wala, S. Campbell, and M. Kiran, 'Insights into DoH: Traffic Classification for DNS over HTTPS in an Encrypted Network', in *Proceedings of the 2023 on Systems and Network Telemetry and Analytics*, Orlando FL USA: ACM, Jul. 2023, pp. 9–17. doi: 10.1145/3589012.3594895.
- [8] K. Jerabek, K. Hynek, O. Rysavy, and I. Burgetova, 'DNS Over HTTPS Detection Using Standard Flow Telemetry', *IEEE Access*, vol. 11, pp. 50000–50012, 2023, doi: 10.1109/ACCESS.2023.3275744.
- [9] K. Jerabek, K. Hynek, and O. Rysavy, 'Comparative analysis of DNS over HTTPS detectors', *Comput. Netw.*, vol. 247, p. 110452, Jun. 2024, doi: 10.1016/j.comnet.2024.110452.
- [10] M. Lyu, H. H. Gharakheili, and V. Sivaraman, 'A Survey on DNS Encryption: Current Development, Malware Misuse, and Inference Techniques', *ACM Comput. Surv.*, vol. 55, no. 8, pp. 1–28, Aug. 2023, doi: 10.1145/3547331.

- [11] S. Siby, M. Juarez, C. Diaz, N. Vallina-Rodriguez, and C. Troncoso, 'Encrypted DNS Privacy? A Traffic Analysis Perspective', in *Proceedings 2020 Network and Distributed System Security Symposium*, San Diego, CA: Internet Society, 2020. doi: 10.14722/ndss.2020.24301.
- [12] J. Bushart and C. Rossow, 'Padding Ain't Enough: Assessing the Privacy Guarantees of Encrypted DNS – Subpage-Agnostic Domain Classification Tor Browser'. Zenodo, Nov. 15, 2022. doi: 10.5281/ZENODO.7319379.
- [13] M. Behnke *et al*, 'Feature Engineering and Machine Learning Model Comparison for Malicious Activity Detection in the DNS-Over-HTTPS Protocol', *IEEE Access*, vol. 9, pp. 129902–129916, 2021, doi: 10.1109/ACCESS.2021.3113294.
- [14] M. Moure-Garrido, S. K. Das, C. Campo, and C. Garcia-Rubio, 'Real-Time Analysis of Encrypted DNS Traffic for Threat Detection', in *ICC 2024 - IEEE International Conference on Communications*, Denver, CO, USA: IEEE, Jun. 2024, pp. 3292–3297. doi: 10.1109/ICC51166.2024.10622347.
- [15] M. Irénée, Y. Wang, X. Hei, X. Song, J. C. Turiho, and E. M. Nyesheja, 'XTS: A Hybrid Framework to Detect DNS-Over-HTTPS Tunnels Based on XGBoost and Cooperative Game Theory', *Mathematics*, vol. 11, no. 10, p. 2372, May 2023, doi: 10.3390/math11102372.
- [16] L. D'hooge, T. Wauters, B. Volckaert, and F. De Turck, 'Inter-dataset generalization strength of supervised machine learning methods for intrusion detection', *J. Inf. Secur. Appl.*, vol. 54, p. 102564, Oct. 2020, doi: 10.1016/j.jisa.2020.102564.
- [17] R. Xie *et al*, 'Rosetta: Enabling Robust TLS Encrypted Traffic Classification in Diverse Network Environments with TCP-Aware Traffic Augmentation', in *Proceedings of the ACM Turing Award Celebration Conference - China 2023*, Wuhan China: ACM, Jul. 2023, pp. 131–132. doi: 10.1145/3603165.3607437.
- [18] Canadian Institute for Cybersecurity, 'University of New Brunswick, "CIRA-CIC-DoHBrw-2020'. 2026. Accessed: Aug. 01, 2026. [Online]. Available: <https://www.unb.ca/cic/datasets/dohbrw-2020.html>
- [19] L. Breiman, 'Random Forests', *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.
- [20] T. Chen and C. Guestrin, 'XGBoost: A Scalable Tree Boosting System', in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco California USA: ACM, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

- [21] D. Chicco and G. Jurman, 'The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation', *BMC Genomics*, vol. 21, no. 1, p. 6, Dec. 2020, doi: 10.1186/s12864-019-6413-7.
- [22] T. Saito and M. Rehmsmeier, 'The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets', *PLOS ONE*, vol. 10, no. 3, p. e0118432, Mar. 2015, doi: 10.1371/journal.pone.0118432.
- [23] F. Pedregosa *et al.*, 'Scikit-learn: Machine Learning in Python', 2012, doi: 10.48550/ARXIV.1201.0490.
- [24] F. Ceschin *et al.*, 'Machine Learning (In) Security: A Stream of Problems', *Digit. Threats Res. Pract.*, vol. 5, no. 1, pp. 1–32, Mar. 2024, doi: 10.1145/3617897.