SvedbergOpen
DISSEMINATION OF KNOWLEDGE

International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/

Research Paper Open Access

Mitigating Identity-Leaking Features in SDN Traffic Classification through Behavioral Learning

Asem Mohamed Elgharyani[1]*, Sherif Kishk[2], Ahmad Hassan[3]

[1]Department of Electronics and Communications Engineering Department, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt. E-mail: asem02087@gmail.com
[2]Department of Electronics and Communications Engineering Department, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt. E-mail: shkishk@mans.edu.eg
[3]Department of Electronics and Communications Engineering Department, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt. E-mail: aahassan@mans.edu.eg

Abstract
Recent growth in network traffic classification has been associated with near-perfect accuracy, with many studies reporting results above 99%. However, such results may create a false impression of model robustness. These high accuracy levels are often achieved through implicit dependence on identity-leaking attributes, such as port numbers and Layer-7 protocol labels. Such features introduce serious methodological vulnerabilities and make models fragile in modern encrypted, tunneled, and dynamically routed network environments. This paper critiques the existing accuracy-centered paradigm by proposing a domain-oriented behavioral machine learning framework that explicitly removes identity-based shortcuts and prioritizes methodological integrity over nominal performance. A highly restrictive filtering pipeline is presented to isolate protocol-neutral behavioral signatures that describe the inherent spatio-temporal characteristics of application traffic. A compact hybrid feature set of 30 behavioral attributes, derived through One-vs-Rest binary importance analysis, combines domain-protected flow measurements with application-specific local signatures. A large-scale experimental assessment using a comprehensive SDN dataset shows that while eliminating identity-leaking features recalibrates the inflated nominal accuracy of 99% to a robust baseline of 80%, it significantly improves generalization stability and cross-session reliability. To confirm that the observed performance results are due to feature quality rather than algorithmic bias, the proposed feature set is evaluated across several machine learning architectures, including Random Forest, XGBoost, Decision Tree, and LightGBM. These models show a consistent convergence in performance. The findings demonstrate that behavioral robustness, rather than inflated accuracy, is the key requirement for deployable network traffic classification systems. This work establishes a practical behavioral standard for encrypted traffic identification and offers a conceptual framework for building reflective, identity-free machine learning models in modern network infrastructures.

Keywords: Network Traffic Classification; Feature Selection; Behavioral Robustness; Machine Learning Integrity; Encrypted Traffic Identification.

1. Introduction

The rapid development of network communication technologies has fundamentally transformed network traffic identification and analysis [1]. The widespread adoption of encryption protocols, including Transport Layer Security and Quick UDP Internet Connections (QUIC), has made many inspection-based methods increasingly ineffective [2-3]. Traditional techniques, including Deep Packet Inspection (DPI) and signature-based classification, rely on payload data and protocol signatures [4-5], which are actively concealed within modern network infrastructures [2-3], [6].

Consequently, modern traffic classification systems must operate under partial observability, where flow-level metadata and temporal characteristics are often the only available sources of information [7]-[8]. To address these limitations, a significant body of recent literature has used machine-learning-based techniques to classify encrypted network traffic by analyzing statistical flow features [4]. While many of these methods report near-perfect accuracy, often exceeding 99%, these figures frequently foster a false sense of reliability [9-10].

Empirical evidence suggests that such high performance is often artificially inflated by implicit reliance on identity-leaking attributes, such as destination and source ports, Layer-7 protocol labels, or transport-layer handshake parameters [7]. Although these features demonstrate high predictive power in controlled laboratory settings, they rely on brittle assumptions regarding fixed port assignments, protocol transparency, and specific operating-system behaviors. Consequently, models trained on such shortcuts experience

significant generalization failure when confronted with dynamic port allocation, protocol tunneling, VPNs, or traffic transformations induced by middleboxes [10].

This phenomenon represents a fundamental methodological challenge rather than a mere implementation flaw. Due to their inherent architecture, tree-based ensembles prioritize the path of least resistance within the feature space to minimize training loss. When identity-based metadata is present, these models exhibit a bias toward learning surface-level correlations, a phenomenon known as shortcut learning, rather than capturing intrinsic behavioral patterns [11]. This issue is exacerbated by automated feature-selection schemes that rely exclusively on statistical correlation; such methods tend to retain features with high label dependency while discarding those with subtle but behaviorally significant temporal dynamics, such as inter-arrival-time variability [12].

The increasing use of Software-Defined Networking (SDN) has fundamentally changed how network traffic is monitored, managed, and analyzed. Unlike traditional network architectures, SDN completely separates the control and data levels, enabling centralized control, a comprehensive network view, and fine-grained monitoring of network traffic at the flow level. As illustrated in Figure 1, SDN environments delegate all control and decision-making to a central SDN controller, while the data-level network switches are limited to routing packets and collecting data flow statistics. [13-14].

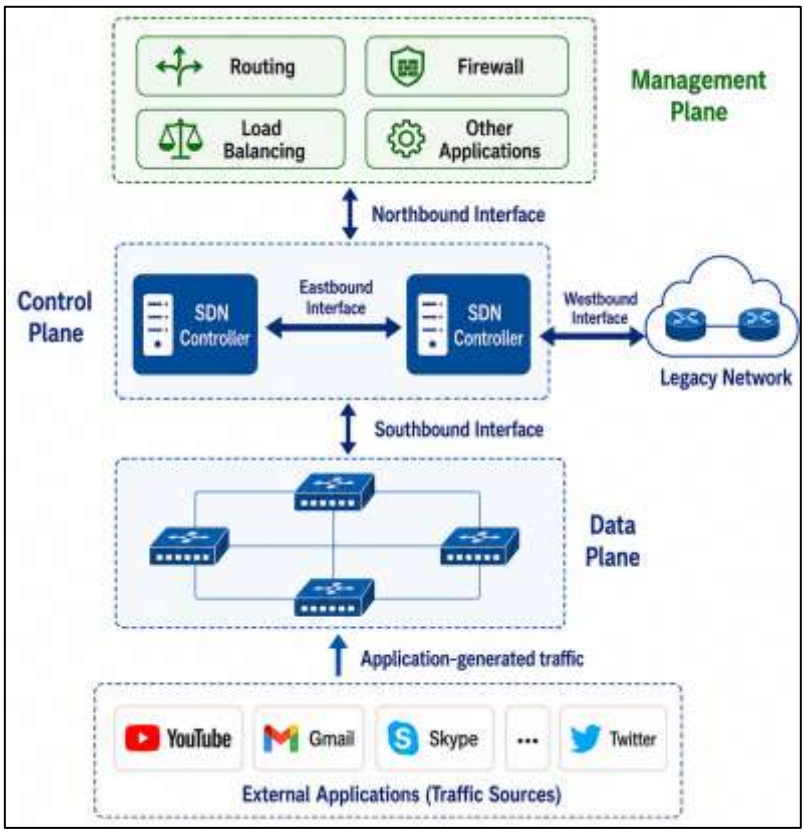


Figure 1. Software Defined Networking (SDN) architecture and its role in behavioral traffic observation Network packets are aggregated into flows and represented through statistical

Temporal characteristics including packet-size distributions, inter-arrival times, and transmission rates. This flow-level abstraction, provided by the SDN data plane, enables the observation of inherent application behavior without depending on explicit identity-related information, including port numbers, protocol identifiers, or payload signatures. Instead, application traffic is captured through flow tables and associated counters within the data plane, while the SDN controller maintains a centralized view of network operations and provides telemetry information to management applications through northbound interfaces [15-17].

The following section provides a critical review of related work in SDN-based and flow-driven network traffic classification. Shamsimukhametov et al. [1] explicitly investigate the impact of Encrypted ClientHello (ECH) on traffic classification and reveal that concealing handshake metadata exposes the brittleness of models that implicitly depend on protocol identifiers. Similarly, Burgetová et al. [3] address the problem of application identification in encrypted environments and conclude that indirect flow-level descriptors

represent the only scalable solution under pervasive encryption. Alwhbi et al. [4] further explore machine-learning-based encrypted traffic classification while acknowledging the methodological risks associated with implicit identity leakage embedded within certain statistical features.

More specialized approaches, such as the work of Chen et al. [7], demonstrate that structural characteristics of encrypted HTTPS traffic, specifically restored Application Data Unit (ADU) lengths, can retain discriminative power even in the absence of payload visibility. At a broader level, comprehensive surveys and analytical studies [8-10] consolidate the state of the art in encrypted traffic analysis and emphasize a clear paradigm shift away from DPI-centric approaches toward flow-based and behavior-driven representations. Collectively, this body of work establishes a strong methodological foundation for abandoning identity-dependent inspection in favor of behavioral flow analysis, directly motivating the design choices adopted in this study.

SDN introduces a fundamental architectural separation between the control plane and the data plane, enabling centralized control, global visibility, and fine-grained flow-level monitoring. This paradigm transforms the network into a programmable measurement infrastructure capable of exposing rich behavioral descriptors without relying on application-layer semantics. Marshoodulla and Saha [19] demonstrate how SDN architectures can mitigate data heterogeneity in IoT environments by leveraging centralized control and consistent flow monitoring.

More closely aligned with the objectives of this research, Alzu'bi et al. [20] analyze the impact of feature quality on SDN-based traffic learning and classification, showing that model performance and robustness are strongly influenced by the behavioral relevance of selected features rather than by the learning algorithm itself. These findings reinforce the role of SDN not merely as an alternative networking paradigm, but as a behavioral observability layer that naturally supports protocol-agnostic, encryption-resilient traffic analysis. This perspective aligns directly with the proposed framework in this work, where SDN-derived flow statistics form the basis for deep data sanitization and behavior-centric feature engineering.

Beyond intrusion detection and anomaly analysis, a significant subset of the literature focuses on application-level traffic classification, particularly for Over-the-Top (OTT) services such as YouTube, Twitter, and Gmail. These studies aim to distinguish application behaviors based on traffic dynamics rather than explicit identifiers. Zhao et al. [21] provide a large-scale mobile traffic dataset designed specifically for application identification, highlighting the importance of comprehensive flow-level data in achieving reliable classification. This research approach plays a vital role in establishing the practical significance of the current study. By focusing on realistic deployment scenarios, the proposed framework is designed to address real-world traffic classification problems, rather than simply detecting breaches and other vulnerabilities.

Recent research has increasingly recognized poor generalization as a significant limitation of contemporary traffic classification models. This issue largely stems from the reliance on statistically correlated features that do not necessarily capture the underlying behavior of network traffic. As a result, there has been a growing emphasis on behavioral feature engineering approaches that leverage temporal dynamics, structural traffic characteristics, and explainable representations to improve model robustness and adaptability across diverse network environments

Sun et al. and Sasi et al [16]. highlight the importance of multi-feature interactions and explainable learning frameworks in capturing meaningful traffic behavior.

Feature-selection studies such as Kouassi et al. [12] demonstrate that consistent performance across multiple classifiers, such as Random Forest, XGBoost, and LightGBM, is a strong indicator of feature robustness rather than algorithmic bias. Liu-Thorrold [11] further emphasizes that bridging the generalization gap in network learning systems requires abandoning identity shortcuts in favor of behavior-based representations. Complementary contributions by Shuwandy et al. and Zhang et al. [22] reinforce the concept of behavioral fingerprints and spatio-temporal traffic patterns as fundamental discriminators. These insights directly support the central premise of this work, which prioritizes methodological integrity and behavioral robustness over inflated nominal accuracy.

Consequently, many modern traffic classification methods achieve excellent quantitative performance; however, their effectiveness is often based on identity-related indicators rather than a robust understanding of traffic behavior. technical robustness, and operational validity in real environments [9-10]. In this study, we propose a Domain-Driven Behavioral ML Approach and introduce a hybrid-signature model that eliminates identity-based features [11], [20].

Rather than relying on identity-dependent attributes, this study constructs a set of 30 behavioral features designed to characterize the intrinsic spatio-temporal dynamics of network traffic [11-12]. This set contains nine network-theory-protected features and 21 signatures derived using binary-importance analysis. Using a Random Forest model optimized through 45 calibration runs, we prioritize "realistic accuracy" over superficial performance while maintaining resilience against encryption.

The main contributions of this study are as follows. First, to address misleading feature-induced bias, we propose a robust framework that does not require ports or IP addresses, making it applicable to production-level scenarios. Second, through hybrid feature engineering, we provide a selection criterion that balances statistical significance with network-domain knowledge. Third, through model-agnostic comprehensive validation, we evaluate the proposed feature set using three modern machine learning architectures, namely XGBoost, LightGBM, and Decision Tree, to demonstrate that behavioral signatures are broadly effective [12].

Methodology

This paper introduces a systematic, multi-stage generalization framework for the comprehensive classification of network applications. The framework employs a flow-based behavioral model that excludes identity-based data leakage, making it suitable for dynamic network environments.

This study is based on the large SDN-based network traffic dataset "IP Network Traffic Flows Labeled with 75 Apps," which records the communication patterns of seventy-five different applications. This dataset provides a rich, high-dimensional feature space consisting of eighty-seven raw features, enabling detailed analysis of application-specific behavior. The dataset is publicly available on Kaggle and is referenced in [23], [24]. Figure 2 presents the proposed methodology, which is organized as a strict pipeline designed to extract unique digital signatures for each application while carefully removing features that could lead to identity leakage or model bias. This transparency ensures that the behavioral signatures derived in later filtering stages are based on a verifiable and accessible data repository.

The first stage cleanses the raw SDN data through two sequential filtering steps to maintain data integrity [19]. Filter 1 eliminates traditional identifiers, such as IP addresses, Flow IDs, and timestamps, thereby removing non-behavioral attributes from the model. Filter 2 applies a domain-protected strategy by excluding misleading domain features, such as port numbers and Layer-7 protocols, which may be unreliable under dynamic port assignment [25], [26]. As a result, the framework prioritizes protected features that represent the raw statistical fingerprints of flows, including flow duration, packet lengths, and inter-arrival times.

The second stage involves advanced feature engineering aimed at defining unique application signatures. Filter 3 applies a One-vs-Rest binary classification approach using a Random Forest importance-ranking strategy to isolate the most discriminative local signatures of each service, such as the persistent bandwidth usage of YouTube compared with the intermittent transactional behavior of Gmail [27], [28]. Filter 4 then combines these local signatures into a single optimized feature set comprising thirty highly condensed behavioral attributes. This process achieves substantial dimensionality reduction while retaining explanatory and discriminative power.

In the final step, the resulting feature matrix is subjected to stringent multi-classifier validation, where the behavioral signatures are evaluated in terms of resilience and generalizability across different mathematical frameworks. Performance is compared across several algorithms, including tree-based ensembles such as XGBoost, LightGBM, and Decision Trees [29]. This extensive validation demonstrates that the observed classification accuracy is due to the inherent quality of the developed behavioral signatures rather than the specific effectiveness of a particular algorithm. Thus, the proposed system is confirmed to be practically appropriate for real-world applications.

While the exclusion of identity-based features may lead to a marginal decrease in nominal accuracy, it significantly enhances the model's ability to generalize across different network topologies and encrypted sessions.

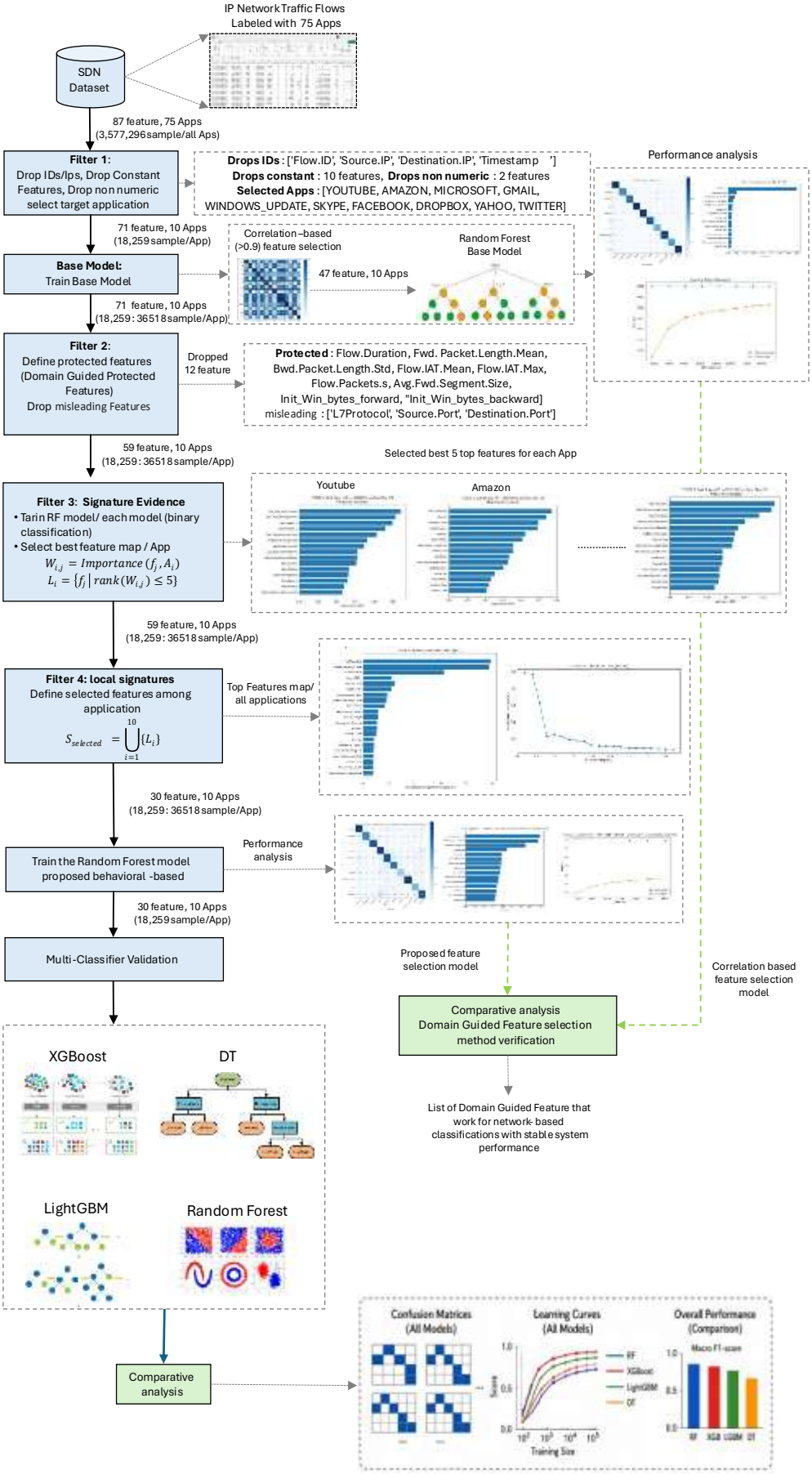


Figure 2. Architecture of the proposed Domain-Driven Behavioral ML Framework

Results

3.1. Phase 1: Performance Analysis of the Base Model

In the initial stage, a Random Forest (RF) classifier was trained as a control model to assess the discriminative capacity of the original feature set. Table 1 shows that the model achieved an accuracy of 99%, with a corresponding weighted F1-score. Figure 3 indicates that the classification performance across all ten application categories is nearly perfect, with minimal cross-class contamination.

However, a deeper examination of this performance suggests that it is indicative of overfitting and data leakage rather than a true understanding of behavior. Although these statistical results are impressive, they do not reflect the model's performance under dynamic or encrypted network conditions [9-10].

Table 1. Detailed Per-Class performance metrics for RF base model

Class	Precision	Recall	F1-score	Support
AMAZON	1.00	1.00	1.00	10,956
DROPBOX	0.99	0.97	0.98	7,531
FACEBOOK	1.00	0.99	0.99	8,710
GMAIL	0.99	1.00	0.99	10,955
MICROSOFT	1.00	1.00	1.00	10,956
SKYPE	0.99	1.00	0.99	9,197
TWITTER	0.95	0.98	0.96	5,478
WINDOWS_UPDATE	1.00	0.99	1.00	10,341
YAHOO	1.00	0.99	1.00	6,380
YOUTUBE	1.00	1.00	1.00	10,955
Total Accuracy			0.99	91459
Total Macro avg	0.99	0.99	0.99	
Total Weighted avg	0.99	0.99	0.99	

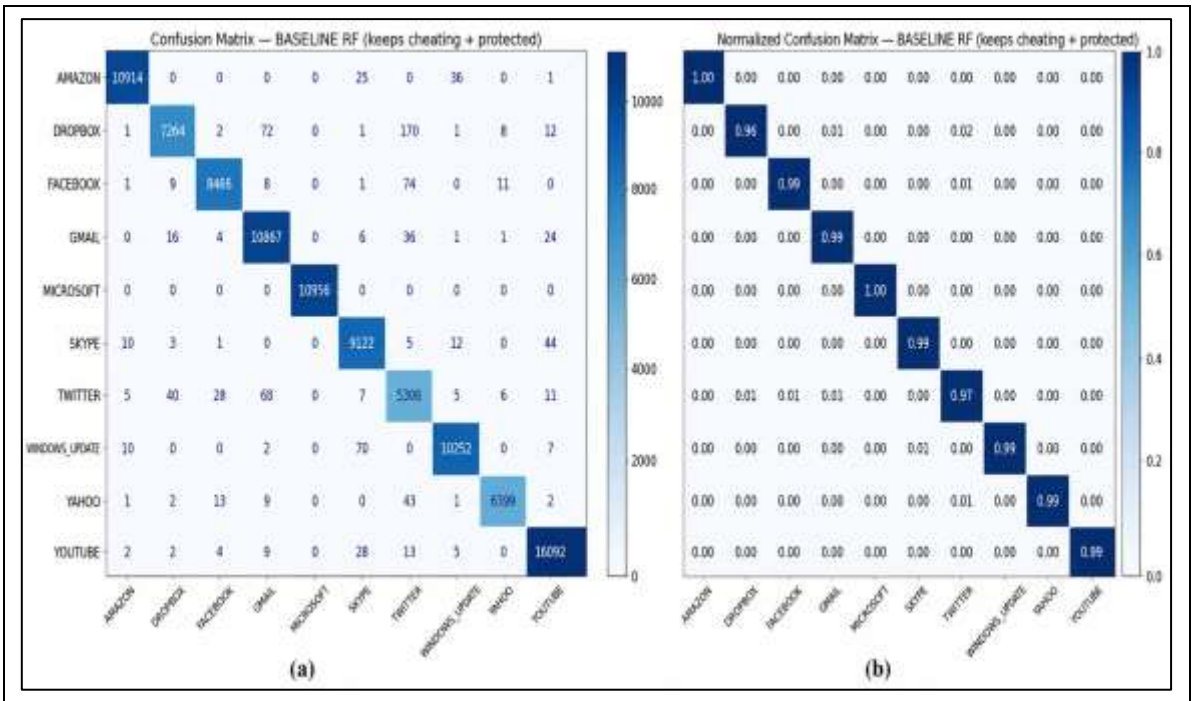


Figure 3. (a) Confusion Matrix, (b) Normalized Confusion Matrix of the RF base line

A critical analysis of the feature-importance distribution shown in Figure 4 highlights a strong reliance on identity-based attributes rather than behavioral dynamics. Table 2 lists the most salient features of the base model and explains their role in cognitive bias and information leakage.

The initial filtering step, which used a "0.9 correlation threshold" and excluded 24 features, proved to be methodologically insufficient. Although mathematically efficient, this method does not consider security-related information gain, as shown in Table 3. It allowed parasitic features, such as ports, to remain because

their linear correlations were weak, despite their tendency to cause overfitting in laboratory-controlled experiments.

This near-perfect accuracy was largely due to two factors: first, artificial data balancing, since the perfectly equal distribution does not reflect the class imbalance typically observed in real-world traffic [30]; and second, shallow discriminative power, since the model does not learn data-flow behavior but instead relies on traffic sources, such as IP addresses and ports [9-11].

Therefore, the baseline model is highly vulnerable to performance degradation when confronted with encrypted traffic, VPN communications, or traffic generated over dynamic ports, where the association between ports and applications becomes unreliable. This methodological limitation motivates the proposed domain-oriented behavioral framework, which deliberately avoids "misleading" features and instead relies on behaviorally meaningful characteristics to achieve more robust and generalizable traffic classification.

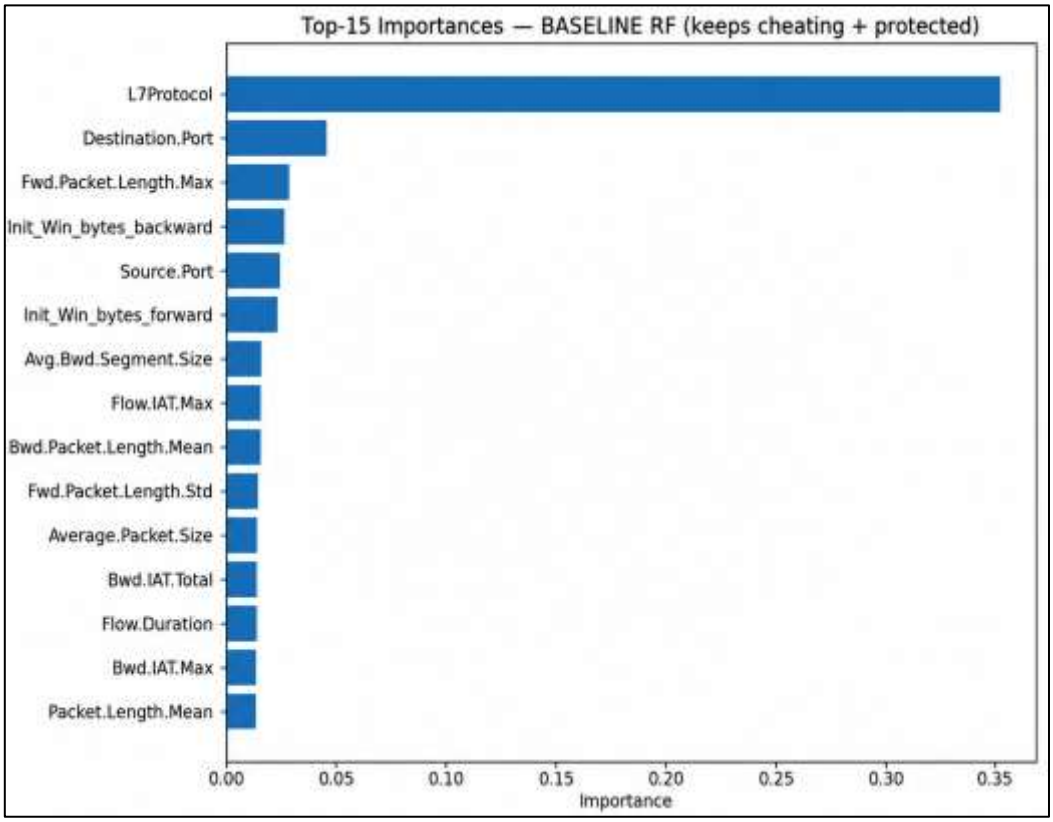


Figure 4. Top-15 important features of the RF baseline

Table 2. Critique of Top Features in the Base Model

Feature Name	Relative Importance	Technical Critique & Leakage Risk
L7Protocol	0.38	Circular Logic: This feature is an output of prior DPI. Relying on it makes the model a "mirror" of previous classifications rather than an independent behavioral analyzer [31].
Destination.Port	0.05	Port Obfuscation Vulnerability: The model treats ports as static indicators. It will fail against modern "Port Hopping" techniques or traffic hidden behind standard ports [32].
Source.Port	0.03	Identity Bias: Source ports are transient and session-specific; they do not characterize the inherent data-flow pattern of an application [33].

3.2. Phase 2: Deep Data Sanitization and Protected Feature Engineering

Phase 2 represents the methodological core of the proposed methodology. By applying Filter 2, the framework moves beyond superficial identifiers and captures the digital fingerprints of network applications.

The main question is: why should the "misleading set" be excluded? The base model, which achieved 99% accuracy, was found to produce artificially inflated performance due to shortcut learning. To address this issue, we explicitly excluded L7Protocol, Source.Port, and Destination.Port [20]. Although Init_Win_bytes

reflect transport initialization, it is constant across encryption layers and remains relatively stable within OS-level stacks, making it fundamentally different from explicit identity tokens such as ports or Layer-7 labels.

This exclusion is based on two key research perspectives. From a networking perspective, particularly regarding port hiding and tunneling, port numbers are no longer reliable sources of information in modern networking environments. Techniques such as port hopping and dynamic port allocation allow applications to change ports dynamically to bypass firewalls. Dependence on such features makes a model vulnerable to these changes; a simple port change can lead to catastrophic generalization failure. Moreover, the L7Protocol feature is likely to be the result of a prior DPI operation. Building a model around this feature represents circular logic and provides no independent research value.

From a machine learning perspective, algorithms, especially tree-based models such as Random Forest and XGBoost, are designed to find the path of least resistance to minimize loss. When port numbers in laboratory data are artificially associated with application labels, the model falls into the trap of shortcut learning tendency. It assigns excessive weight to these leaky features and largely ignores deeper behavioral attributes, such as inter-arrival time. This produces a brittle model that performs well in a controlled laboratory environment but fails in real-world deployments where port-to-application mappings are dynamic.

The protected feature set was therefore designed to extract the "rhythm of the flow" rather than identity-based shortcuts. Instead of relying on identity crutches, we identified a collection of protected features that are protocol-invariant. These characteristics remain observable even when the payload is encrypted or the port is obfuscated. These metrics were selected because they represent the biometric fingerprint of an application, as described in Table 3.

First, temporal dynamics, such as Flow.IAT, capture the pulse of the application [7-8]. For example, a transactional application such as Gmail often exhibits a distinct "Burst-Wait-Burst" pattern, whereas a streaming application such as YouTube typically shows a more regular and rhythmic flow. These temporal patterns are difficult to alter because they are part of the application logic and remain visible even under encryption.

Second, packet-length statistics capture the nature of the communication between the client and the server. Packet-size distributions differ fundamentally between text-based communication, video streaming, and gaming traffic. Therefore, they can serve as behavioral signatures of the underlying application protocol [7-8].

Third, flow dynamics, such as Flow.Packets/s, measure the throughput density required for the application to function effectively. This is a behavioral characteristic that helps distinguish interactive services from background system updates [11-12].

Unlike correlation-driven feature selection, this categorization enforces semantic separation between temporal, volumetric, and structural traffic properties, ensuring that no feature encodes implicit identity information or deployment-specific artifacts.

Table 3. The Protected Behavioral Feature Matrix, all listed features satisfy the protocol-invariance, encryption-resilience, and application-logic dependency criteria.

Feature Category	Feature Name	Description
Volume Metrics	Flow.Duration	Reflects the session life cycle; distinguish between long-lived updates and short-lived queries.
Packet Length Stats	Fwd.Packet.Length.Mean	Represents the "Typical Payload Size"; a behavioral constant reflecting the type of data transferred.
Packet Length Stats	Bwd.Packet.Length.Std	Measures response variability; a powerful metric for distinguishing multi-media responses from simple text.
Temporal (IAT)	Flow.IAT.Mean	The "Temporal Rhythm" of the app; the hardest feature for an obfuscator to alter without degrading Quality of Service (QoS).
	Flow.IAT.Max	Captures peak inter-arrival delays, reflecting burst-boundary behavior and idle gaps inherent to application logic.
Flow Dynamics	Flow.Packets.s	Reflects traffic density; a core feature for separating Streaming from Transactional behaviors.
Segment Metrics	Avg.Fwd.Segment.Size	Provides insight into physical layer segmentation, linked directly to the underlying protocol's congestion control.
	Init_Win_bytes_forward	Initial TCP receive window size in the forward direction; reflects OS-level TCP stack configuration.

Transport Initialization Metrics	Init_Win_bytes_backward	Initial TCP receive window size in the backward direction; influenced by kernel parameters and network infrastructure characteristics.

3.3. Feature Fusion Strategy

To validate our behavioral framework, we selected ten popular network applications for this study: Amazon, Dropbox, Facebook, Gmail, Microsoft, Skype, Twitter, Windows Update, Yahoo, and YouTube. The training mechanism was structured so that the model would learn the distinct behavioral signature of each service rather than rely on temporary network identifiers.

To obtain a granular and fair evaluation of feature importance, we divided the multi-class problem into ten binary sub-problems using a One-vs-Rest (OvR) strategy. A separate Random Forest classifier was trained to distinguish each target application from the remaining classes. For example, each binary model was trained to classify one application, such as Amazon, against all other applications.

The main objective of this stage is to compute the feature-importance score, $W_{i,j}$, for every candidate feature j with respect to each application i. This score is mathematically calculated using the Mean Decrease in Gini Impurity (MDI) [34]:

$$W_{i,j} = \frac{1}{M}\sum_{k=1}^{M} \Delta G(i,k) \quad (1)$$

Where, M is the number of trees in the forest and $\Delta G(i,k)$ is the reduction in Gini impurity that was obtained by splitting on feature j in tree k. We obtain a Local Signature Set, denoted as L_i, of the top-K features (K = 5) with the largest discriminative power to that particular application (binary model) [12]:

$$L_i = f_j \mid rank(W_{i,j}) \leq 5 \quad (2)$$

The space of final features (S_{final}) is built using a composite logic of weight and repetition. This is to make sure that the model both reflects features that are of universal importance in many applications or highly discriminative to one service [11]:

$$S_{final} = P \cup \bigcup_{i=1}^{10} \{L_i\} \quad (3)$$

where L_i is the set of local signatures that have been algorithmically identified through binary training and P is the Protected Behavioural Set, a set of fundamental flow measures that are maintained manually to provide domain integrity and anti-encryption resistance.

A detailed examination of the binary classification models reveals an important characteristic of the proposed framework: different application classes tend to rely on distinct sets of discriminative features. The binary feature-importance analysis shows limited dominance of a common feature across classes, suggesting that the model leverages diverse behavioral characteristics when distinguishing between applications. For instance, YouTube is primarily characterized by Bwd.Packets.s, which achieves the highest importance score of 0.0583. This suggests that the size of packets received from the server is an important discriminative feature for YouTube traffic. The result may be attributed to the content-delivery patterns commonly observed in video streaming services, where substantial amounts of data are transferred from servers to end use .On the other hand, Twitter is characterized mainly by Fwd.Packets.s and Bwd.Packets.s and Flow.Bytes.s, which achieved importance scores of 0.0580, 0.0440, and 0.0390, respectively. These results indicate that the classification model primarily captures the intensity of packet exchanges and the volume of transmitted data, highlighting the dynamic and interactive communication patterns associated with Twitter traffic. This variation, summarized in Table 4, provides evidence that the selected features capture different aspects of application behavior rather than depending on a single identity-related attribute. As a result, the classification process appears to be driven by application-specific behavioral characteristics instead of potentially misleading indicators, such as port numbers, that could artificially influence multiple classes.

Furthermore, unlike the baseline model, which exhibited a strong dependence on a single feature, namely L7Protocol, with an importance score of 0.38, the proposed model demonstrates a more balanced distribution of feature importance, as illustrated in Figure 5. This distribution suggests a reduced reliance on any single dominant attribute and reflects a broader utilization of the behavioral information contained within the traffic flows.

Table 4. Top-5 Local Signatures for binary classification model

Application	Top 5 features, importance
AMAZON	Fwd.Packets.s: 0.0543, Bwd.Packets.s: 0.0462, wd.Packet.Length.Max: 0.0420, Bwd.Packet.Length.Mean: 0.0417, Fwd.Packet.Length.Max: 0.0353

DROPBOX	Bwd.Packet.Length.Min: 0.0937, Idle.Mean: 0.0441, Bwd.IAT.Max: 0.0433, Idle.Min: 0.0422, Fwd.Packet.Length.Max: 0.0390
FACEBOOK	Fwd.Packet.Length.Max: 0.0700, Flow.IAT.Min: 0.0660, Fwd.IAT.Min: 0.0426, Bwd.IAT.Total: 0.0406, Max.Packet.Length: 0.0390
GMAIL	Fwd.Packet.Length.Max: 0.0490, Fwd.Packets.s: 0.0440, Bwd.Packets.s: 0.0391, Bwd.Packet.Length.Max: 0.0349, Average.Packet.Size: 0.0306
MICROSOFT	Fwd.Packets.s: 0.0597, Bwd.Packets.s: 0.0495, Fwd.Packet.Length.Max: 0.0471, Fwd.IAT.Max: 0.0386, Flow.IAT.Min: 0.0372
SKYPE	Fwd.Packet.Length.Max: 0.0488, Fwd.Packets.s: 0.0465, Max.Packet.Length: 0.0379, Bwd.Packets.s: 0.0373, Flow.Bytes.s: 0.0351
TWITTER	Fwd.Packets.s: 0.0580, Bwd.Packets.s: 0.0440, Flow.Bytes.s: 0.0390, Bwd.IAT.Max: 0.0387, Flow.IAT.Std: 0.0381
WINDOWS_UP-DATE	Fwd.Packet.Length.Max: 0.1080, Subflow.Fwd.Bytes: 0.0573, To-tal.Length.of.Fwd.Packets: 0.0560, Bwd.Packet.Length.Mean: 0.0536, Packet.Length.Mean: 0.0505
YAHOO	Fwd.Packets.s: 0.0681, Bwd.Packets.s: 0.0533, Fwd.Packet.Length.Max: 0.0501, Flow.IAT.Min: 0.0460, Fwd.Packet.Length.Std: 0.0427
YOUTUBE	Bwd.Packets.s: 0.0*583*, Fwd.Packet.Length.Max: 0.0534m, , Fwd.Packets.s: 0.0488, Bwd.Packet.Length.Max: 0.0389, Flow.IAT.Min: 0.0386

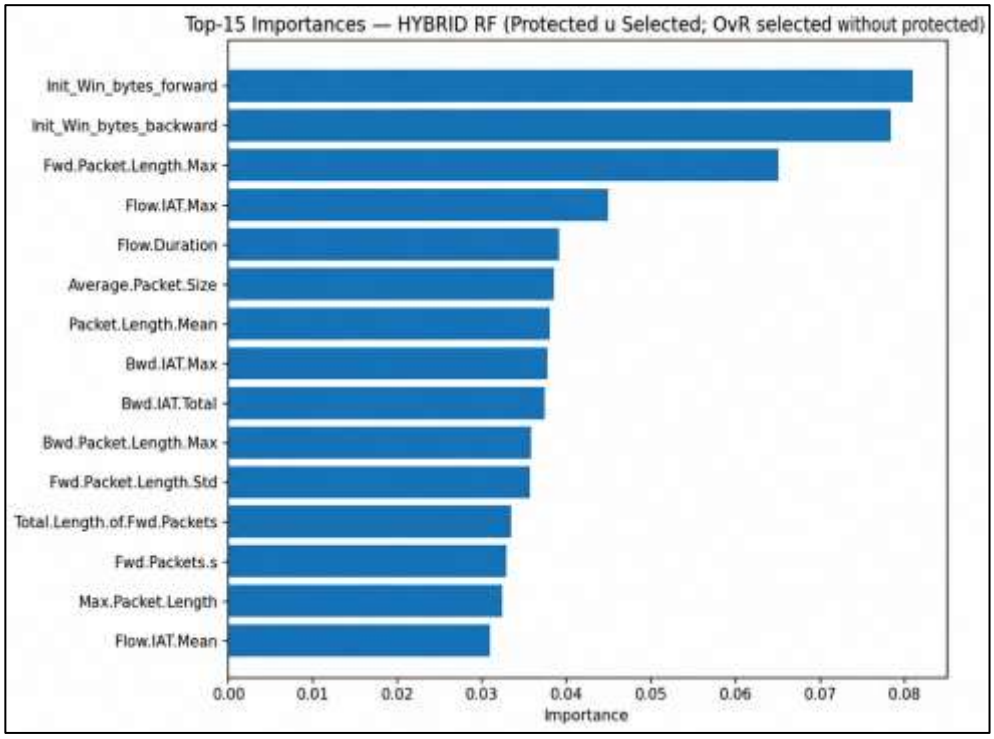


Figure 5. Global Importance of the Proposed Model

The importance gradient remains smooth across the remaining 21 features. This balance is an important indicator of model stability. Since decision-making weight is distributed across several behavioral characteristics, the loss of a single feature is unlikely to cause a complete breakdown in classification performance. The change from 99% accuracy in the base model to 80% accuracy in the proposed Random Forest model, as shown in Table 5, represents a shift from "nominal accuracy" to operational robustness. First, the reduction in accuracy is mainly due to the removal of identity-based shortcuts. As reflected in the performance metrics, the behavioral features shown in Table 2 demonstrate that the model learns the rhythm of the flow using only behavioral attributes.

Second, although the overall accuracy is reduced, the confusion matrix in Figure 6 and the per-class reports show balanced performance across all ten applications. This confirms that the model does not over-optimize in favor of a particular class. Instead, it provides a stable and consistent baseline of domain-specific intelligence.

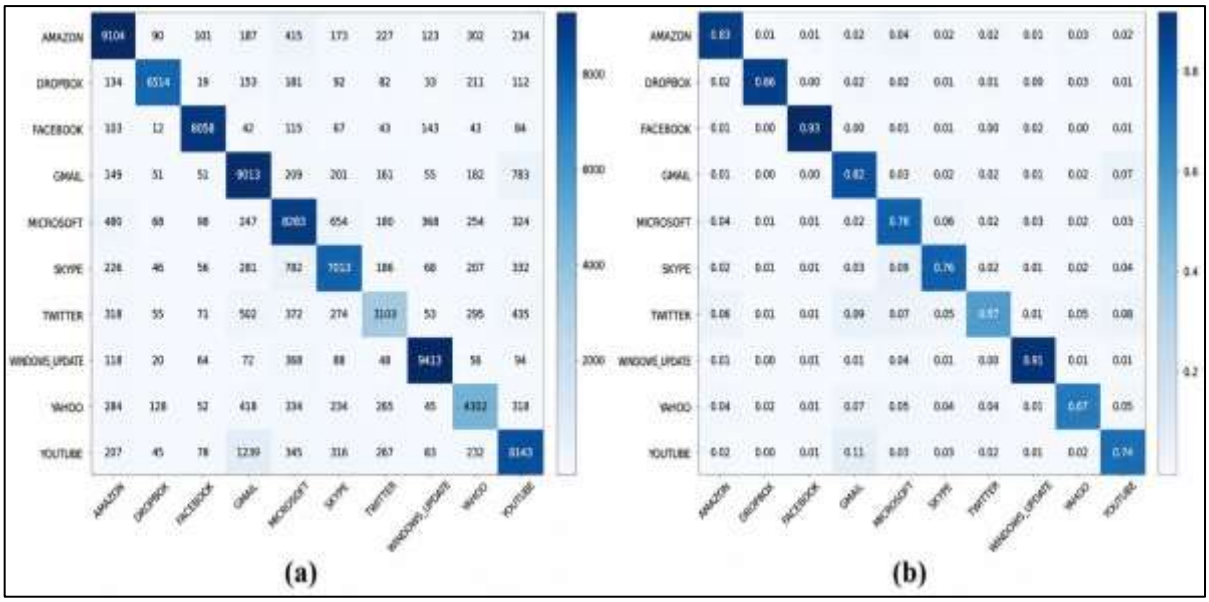


Figure 6. (a) Confusion Matrix, (b) Normalized Confusion Matrix the Random Forest proposed model

Table 5. Detailed performance metrics for RF proposed model

Class	Precision	Recall	F1-Score	Support
AMAZON	0.82	0.83	0.82	10,956
DROPBOX	0.93	0.86	0.89	7,531
FACEBOOK	0.93	0.93	0.93	8,710
GMAIL	0.74	0.82	0.78	10,955
MICROSOFT	0.72	0.76	0.74	10,956
SKYPE	0.77	0.76	0.77	9,197
TWITTER	0.68	0.57	0.62	5,478
WINDOWS_UPDATE	0.91	0.91	0.91	10,341
YAHOO	0.71	0.67	0.69	6,380
YOUTUBE	0.75	0.74	0.75	10,955
Accuracy			**0.80**	91,459
Macro Avg	0.80	0.79	0.79	91,459
Weighted Avg	0.80	0.80	0.80	91,459

3.4. Model validation: multi modal classification

To confirm that the discriminative ability of the proposed behavioral framework results from the selected features rather than from a specific algorithmic behavior, multi-classifier validation was conducted. The final hybrid feature set was evaluated using several machines learning architectures, including XGBoost, LightGBM, and Decision Tree, to determine whether the identified behavioral signatures are robust and generalizable.

This cross-validation confirms the reliability of the framework and its applicability to different mathematical modeling approaches. It also demonstrates that classification performance is grounded in intrinsic traffic dynamics rather than in model-specific advantages.

The summary of the multi-model evaluation results presented in Table 6 shows a robust and consistent performance baseline across all tested classifiers. The strong consistency between XGBoost (83%) and LightGBM (82%) indicates a high degree of algorithmic consensus. This suggests that the proposed hybrid feature-selection methodology successfully isolates a stable behavioral essence of network traffic, which can be recognized with high precision by modern gradient-boosting learners.

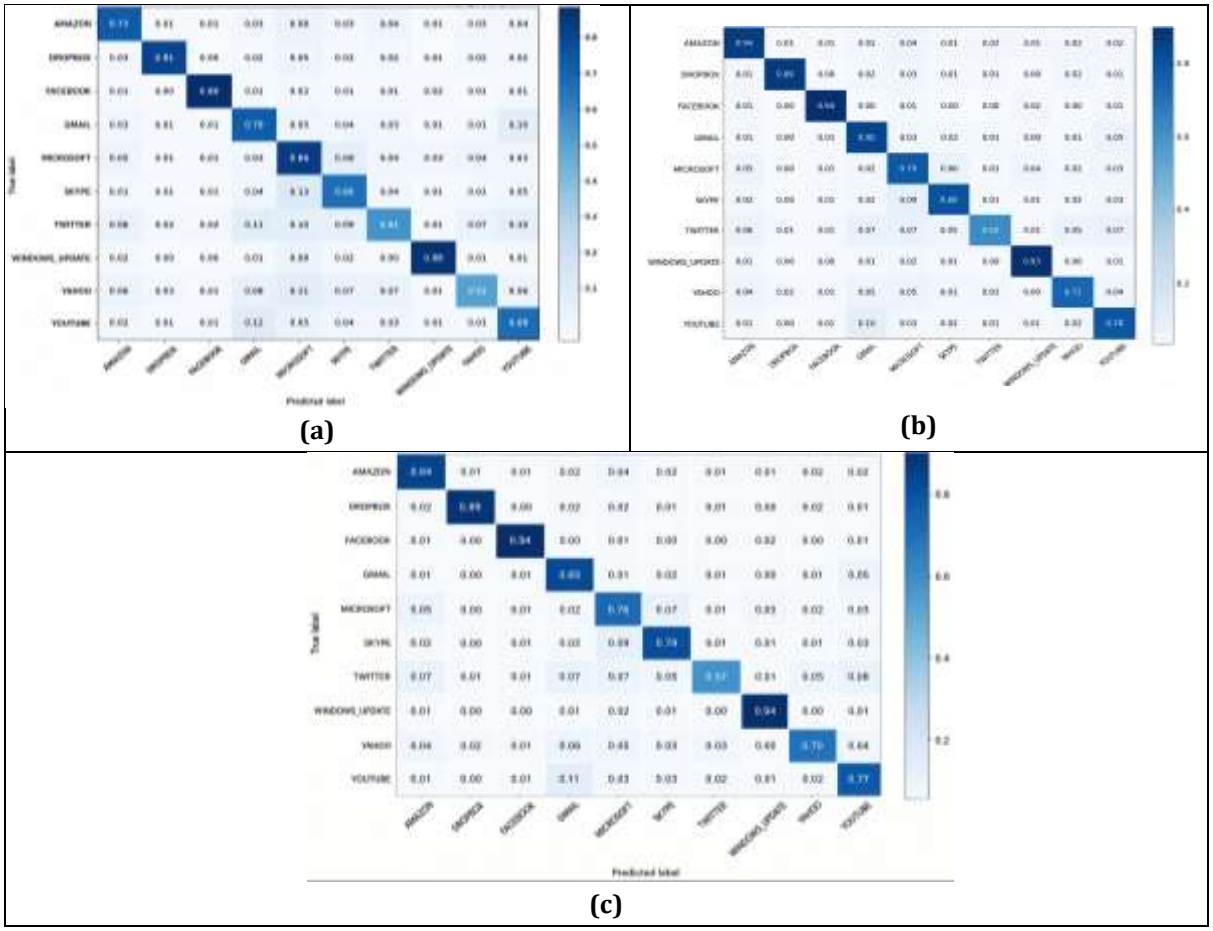
The lower accuracy of the Decision Tree model (71%) indicates that the relationships among purely behavioral features, including IAT distributions and window sizes, are complex and non-linear. Therefore,

ensemble-based methods are important for capturing the high-dimensional complexity of modern encrypted traffic.

The validity of the model is further assessed through the analysis of the confusion matrices for the individual classifiers. Unlike the base model, which relied on identity-leaking features, the observed errors in the proposed framework are behaviorally rational. Figure 7, Figure 8, and Figure 9 present the performance analysis of XGBoost, LightGBM, and Decision Tree.

Table 6. Performance Metrics for the Validated Behavioral Frameworks

Class/metrics	Decision Tree (%)			XGBoost (%)			LightGBM (%)			
	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Support
AMAZON	0.75	0.73	0.74	0.83	0.84	0.84	0.82	0.84	0.83	10,956
DROPBOX	0.88	0.81	0.84	0.94	0.89	0.92	0.94	0.89	0.91	7,531
FACEBOOK	0.92	0.89	0.91	0.94	0.94	0.94	0.94	0.94	0.94	8,710
GMAIL	0.66	0.70	0.68	0.78	0.85	0.82	0.77	0.85	0.81	10,955
MICROSOFT	0.57	0.66	0.61	0.73	0.78	0.75	0.72	0.77	0.74	10,956
SKYPE	0.63	0.66	0.64	0.80	0.80	0.80	0.78	0.79	0.79	9,197
TWITTER	0.45	0.41	0.42	0.77	0.60	0.67	0.75	0.58	0.66	5,478
WINDOWS_UPDATE	0.90	0.86	0.88	0.91	0.93	0.92	0.92	0.94	0.93	10,341
YAHOO	0.58	0.51	0.54	0.76	0.72	0.74	0.76	0.70	0.73	6,380
YOUTUBE	0.67	0.68	0.67	0.78	0.78	0.78	0.78	0.77	0.77	10,955
Accuracy			0.71			0.83			0.82	91,459
Macro Avg	0.70	0.69	0.70	0.83	0.81	0.82	0.82	0.81	0.81	
Weighted Avg	0.71	0.71	0.71	0.83	0.83	0.82	0.82	0.82	0.82	



(a) **(b)** **(c)**

Figure 7. Normalized Confusion Matrix of the validation models (a) Decision Tree, (b) XGBoost, and () LightGBM.

(a)	(b)
(c)	

Figure 8. Top 15 important features selected by of the validation models (a) Decision Tree, (b) XGBoost, and (c) LightGBM.

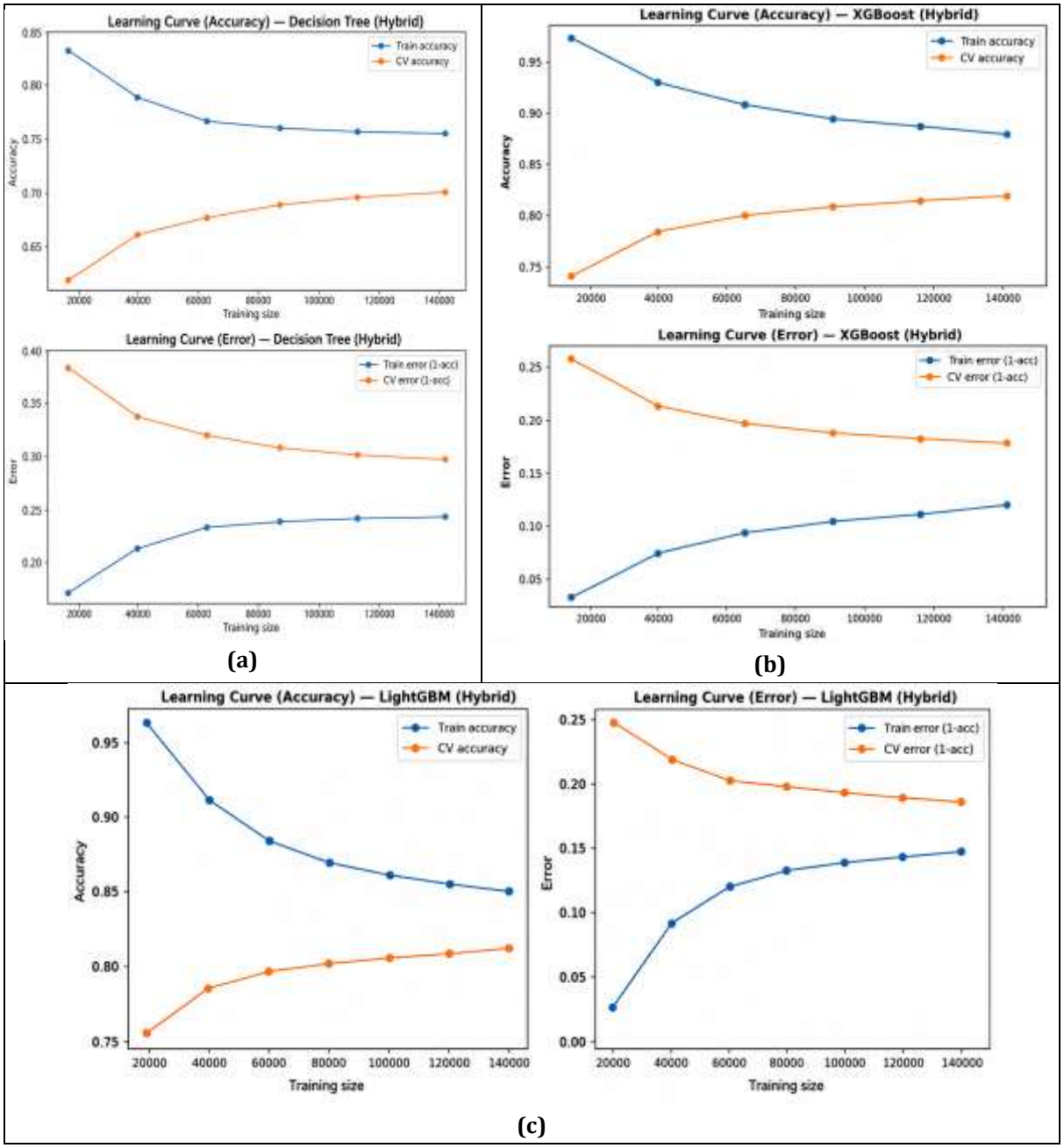


Figure 9. Accuracy and error learning curves of the validation models (a) Decision Tree, (b) XGBoost, and (c) LightGBM.

4. Discussion

The main goal of this research was to address the critical issue of identity leakage in current network traffic classification systems, where models often rely on "Features whose apparent predictive power depends on dataset-specific artifacts as opposed to durable, robust behavioral signal", such as port numbers and Layer-7 labels, rather than genuine behavioral patterns. This dependency introduces a risk of misleadingly high classification accuracy, as the learned patterns may not generalize to encrypted environments or scenarios involving protocol evolution and dynamic network behavior

To overcome this problem, we developed a behavioral framework that recognizes applications through their unique flow rhythm. The proposed approach validates our hypothesis through a multi-stage filtering process based on data integrity and domain-driven feature selection. Using a binary One-vs-Rest (OvR) strategy, we extracted 30 application-specific signatures, including inter-arrival times (IATs) and packet-length distributions. This shifted the model's focus away from identity-leaking shortcuts and toward underlying traffic dynamics.

The results show that although nominal accuracy decreased from a biased 99% to a more realistic 80%, the proposed model achieved much higher methodological integrity and stability. To confirm that the results were not dependent on a specific algorithm, we conducted multi-classifier validation. The strong and consistent performance of XGBoost (83%), LightGBM (82%), and Decision Tree (71%) provides clear algorithmic consensus regarding the quality of the proposed feature set.

The confusion matrices further demonstrate that the framework can distinguish between similar applications based on their actual data-flow properties rather than apparent metadata. Therefore, this study concludes that behavioral robustness is more important than nominal accuracy for real-world deployment.

Future work will test this framework in Internet of Things (IoT) simulations using tools such as NS-3, Cooja, and Mininet-WiFi to analyze its performance under latent and ad hoc traffic conditions. We will also explore model-efficiency optimization through Bayesian hyperparameter optimization and the adoption of deep learning architectures, including Temporal Convolutional Networks (TCNs), to better capture temporal information without significantly increasing model size. This direction may support practical implementation in edge-computing environments.

References

1. D. Shamsimukhametov, A. Kurapov, M. Liubogoshchev, and E. Khorov, "Is Encrypted ClientHello a Challenge for Traffic Classification?," *IEEE Access*, vol. 10, pp. 77883–77897, 2022, doi: 10.1109/ACCESS.2022.3191431.
2. J. Zhou *et al.*, "Challenges and Advances in Analyzing TLS 1.3-Encrypted Traffic: A Comprehensive Survey," *Electronics 2024, Vol. 13,* vol. 13, no. 20, Oct. 2024, doi: 10.3390/ELECTRONICS13204000.
3. I. Burgetová, P. Matoušek, and O. Ryšavý, "Towards identification of network applications in encrypted traffic," *Annals of Telecommunications 2025*, pp. 1–18, Sep. 2025, doi: 10.1007/S12243-025-01114-Z.
4. I. A. Alwhbi, C. C. Zou, R. N. Alharbi, I. A. Alwhbi, C. C. Zou, and R. N. Alharbi, "Encrypted Network Traffic Analysis and Classification Utilizing Machine Learning," *Sensors 2024, Vol. 24,* vol. 24, no. 11, May 2024, doi: 10.3390/S24113509.
5. J. Ahn, R. Hussain, K. Kang, and J. Son, "Exploring Encryption Algorithms and Network Protocols: A Comprehensive Survey of Threats and Vulnerabilities," *IEEE Communications Surveys and Tutorials*, 2025, doi: 10.1109/COMST.2025.3526605.
6. R. Picot, F. Gohring de Magalhaes, A. Shahnejat Bushehri, M. Ben Atti, G. Nicolescu, and A. Quintero, "Protocol-Agnostic and Packet-Based Intrusion Detection Using a Multi-Layer Deep-Learning Architecture at the Network Edge," *IEEE Access*, vol. 13, pp. 57867–57877, 2025, doi: 10.1109/ACCESS.2025.3555201.
7. Z. Chen, G. Cheng, D. Niu, Y. Zhao, Y. Zhou, and S. Jiang, "Ultimate Encrypted Traffic Feature Engineering: HTTPS Encrypted Traffic Classification Using Restored Application Data Unit Length," *IEEE Trans. Dependable Secure Comput.*, 2025, doi: 10.1109/TDSC.2025.3615592.
8. M. Shen *et al.*, "Machine Learning-Powered Encrypted Network Traffic Analysis: A Comprehensive Survey," *IEEE Communications Surveys and Tutorials*, vol. 25, no. 1, pp. 791–824, 2023, doi: 10.1109/COMST.2022.3208196.
9. S. Jorgensen *et al.*, "Extensible Machine Learning for Encrypted Network Traffic Application Labeling via Uncertainty Quantification," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 1, pp. 420–433, Jan. 2024, doi: 10.1109/TAI.2023.3244168.
10. F. Alserhani, "Analysis of Encrypted Network Traffic for Enhancing Cyber-security in Dynamic Environments," *Applied Artificial Intelligence*, vol. 38, no. 1, p. 2381882, Dec. 2024, doi: 10.1080/08839514.2024.2381882;WGROUP:STRING:PUBLICATION.
11. L. Liu-Thorrold, "Neural Network Models for Network Intrusion Detection: Closing the Generalisation Gap," 2025, doi: 10.26190/UNSWORKS/31548.
12. B. M. Kouassi *et al.*, "Top-K Feature Selection for IoT Intrusion Detection: Contributions of XGBoost, LightGBM, and Random Forest," *Future Internet 2025, Vol. 17,* vol. 17, no. 11, Nov. 2025, doi: 10.3390/FI17110529.

13. M. Hussain, N. Shah, R. Amin, S. S. Alshamrani, A. Alotaibi, and S. M. Raza, "Software-Defined Networking: Categories, Analysis, and Future Directions," *Sensors 2022, Vol. 22,* vol. 22, no. 15, Jul. 2022, doi: 10.3390/S22155551.
14. J. Liu and Q. Xu, "Machine learning in software defined network," *Proceedings of 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2019*, pp. 1114–1120, Mar. 2019, doi: 10.1109/ITNEC.2019.8729331.
15. D. Nuñez-Agurto, W. Fuertes, L. Marrone, and M. Macas, "Machine Learning-Based Traffic Classification in Software-Defined Networking: A Systematic Literature Review, Challenges, and Future Research Directions.," *IAENG Int. J. Comput. Sci.*, vol. 49, no. 4, p. 1002, Dec. 2022, Accessed: Jan. 26, 2026. [Online].
16. Y. Zhao, Y. Li, X. Zhang, G. Geng, W. Zhang, and Y. Sun, "A Survey of Networking Applications Applying the Software Defined Networking Concept Based on Machine Learning," *IEEE Access*, vol. 7, pp. 95397–95417, 2019, doi: 10.1109/ACCESS.2019.2928564.
17. S. V. Margariti, I. G. Tsoulos, E. Kiousi, and E. Stergiou, "Traffic Classification in Software-Defined Networking Using Genetic Programming Tools," *Future Internet 2024, Vol. 16,* vol. 16, no. 9, Sep. 2024, doi: 10.3390/FI16090338.
18. A. O. Salau and M. M. Beyene, "Software defined networking based network traffic classification using machine learning techniques," *Scientific Reports 2024 14:1*, vol. 14, no. 1, pp. 20060-, Aug. 2024, doi: 10.1038/s41598-024-70983-6.
19. S. Z. Marshoodulla and G. Saha, "An approach towards removal of data heterogeneity in SDN-based IoT framework," *Internet of Things*, vol. 22, p. 100763, Jul. 2023, doi: 10.1016/J.IOT.2023.100763.
20. A. Alzu'bi, J. Khrais, N. Ennab, and A. Abuarqoub, "The impact of feature quality on SDN traffic learning and classification," *IET Conference Proceedings*, vol. 2022, no. 14, pp. 192–197, Nov. 2022, doi: 10.1049/ICP.2022.2446.
21. S. Zhao, S. Chen, F. Wang, Z. Wei, J. Zhong, and J. Liang, "A Large-Scale Mobile Traffic Dataset For Mobile Application Identification," *Comput. J.*, vol. 67, no. 4, pp. 1501–1513, Apr. 2024, doi: 10.1093/COMJNL/BXAD076.
22. D. Zhang *et al.*, "A Dynamic Website Fingerprinting Defense by Emulating Spatio-Temporal Traffic Features," *Electronics 2025, Vol. 14,* vol. 14, no. 22, Nov. 2025, doi: 10.3390/ELECTRONICS14224441.
23. J. S. Rojas, Á. R. Gallón, and J. C. Corrales, "Personalized service degradation policies on OTT applications based on the consumption behavior of users," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10962 LNCS, pp. 543–557, 2018, doi: 10.1007/978-3-319-95168-3_37.
24. "IP Network Traffic Flows Labeled with 75 Apps." Accessed: Feb. 25, 2026. [Online]. Available: https://www.kaggle.com/datasets/jsrojas/ip-network-traffic-flows-labeled-with-87-apps
25. A. Agrawal *et al.*, "A survey on analyzing encrypted network traffic of mobile devices," *International Journal of Information Security 2022 21:4*, vol. 21, no. 4, pp. 873–915, Feb. 2022, doi: 10.1007/S10207-022-00581-Y.
26. J. Holland, P. Schmitt, P. Mittal, and N. Feamster, "Towards Reproducible Network Traffic Analysis," Mar. 2022, Accessed: Jan. 18, 2026. [Online]. Available: https://arxiv.org/pdf/2203.12410
27. M. Awad, S. Fraihat, M. Awad, and S. Fraihat, "Recursive Feature Elimination with Cross-Validation with Decision Tree: Feature Selection Method for Machine Learning-Based Intrusion Detection Systems," *Journal of Sensor and Actuator Networks 2023, Vol. 12,* vol. 12, no. 5, Sep. 2023, doi: 10.3390/JSAN12050067.
28. J. Arafat, F. Tasmin, and S. Poudel, "Feature Selection and Regularization in Multi-Class Classification: An Empirical Study of One-vs-Rest Logistic Regression with Gradient Descent Optimization and L1 Sparsity Constraints," *Proceedings of*, vol. 1, Oct. 2025, Accessed: Jan. 18, 2026. [Online]. Available: https://arxiv.org/pdf/2510.14449
29. A. Choudhury, A. Mondal, and S. Sarkar, "Searches for the BSM scenarios at the LHC using decision tree-based machine learning algorithms: a comparative study and review of random forest, AdaBoost, XGBoost and LightGBM frameworks," *The European Physical Journal Special Topics 2024 233:15*, vol. 233, no. 15, pp. 2425–2463, Sep. 2024, doi: 10.1140/EPJS/S11734-024-01308-X.
30. M. Abbasi, S. Lopez Florez, A. Shahraki, A. Taherkordi, J. Prieto, and J. M. Corchado, "Class Imbalance in Network Traffic Classification: An Adaptive Weight Ensemble-of-Ensemble Learning Method," *IEEE Access*, vol. 13, pp. 26171–26192, 2025, doi: 10.1109/ACCESS.2025.3538170.

31. W. Luo *et al.*, "Doxing via the Lens: Revealing Location-related Privacy Leakage on Multi-modal Large Reasoning Models," Apr. 2025, Accessed: Jan. 18, 2026. [Online]. Available: https://arxiv.org/pdf/2504.19373
32. P. Beltrán-López, M. G. Pérez, and P. Nespoli, "Cyber Deception: Taxonomy, State of the Art, Frameworks, Trends, and Open Challenges," *IEEE Communications Surveys and Tutorials*, 2025, doi: 10.1109/COMST.2025.3594788.
33. S. Mulazzi, "Design and Validation of a Synchronized Heterogeneous Wireless Sensor Network: From Acquisition to Real-Time Inference," Dec. 2025.
34. R. A. Disha and S. Waheed, "Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique," *Cybersecurity 2022 5:1*, vol. 5, no. 1, pp. 1-, Jan. 2022, doi: 10.1186/S42400-021-00103-8.