Machine Learning-Based Classification of Encrypted VPN and Non-VPN Traffic with Temporal Features Analysis

Dilawer Khan[1], Musawer Hamad Khan[1], Muhammad Bilal[1], Hameed Ullah Khan[2], Shafiq Ur Rehman Khan[1], Alishba Khalid[1]

[1]Department of Computer Science, Namal University, Mianwali, Pakistan

[2]Department of Computer Science, Sir Syed CASE Institute of Technology, Islamabad, Pakistan

*__Correspondence__: muhammad.bilal@namal.edu.pk

Citation | Khan. D, Khan. M. H, Bilal. M. Khan. H. U, Khan. S. U. R, Khalid. A, "Machine Learning-Based Classification of Encrypted VPN and Non-VPN Traffic with Temporal Features Analysis", IJIST, Special Issue pp 44-62, April 2026

Received | March 15, 2026 **Revised |** April 18, 2026 **Accepted |** April 23, 2026 **Published |** April 27, 2026

As Virtual Private Network (VPN) usage increases globally for privacy preservation and unrestricted access, distinguishing VPN traffic from regular internet traffic has become both critically important and challenging. Traditional detection methods relying on port-based rules and deep packet inspection are no longer reliable against encrypted communications, prompting the need for smarter, adaptive, machine learning (ML) solutions. This study proposes a comprehensive ML-based framework to classify VPN and non-VPN traffic using a large-scale, balanced dataset of approximately six million packets, covering common application types (Mail, Video Conferencing, SSH, Non-Streaming) and five VPN protocols (L2TP, OpenVPN, PPTP, SSTP, and Wire Guard). Five models were evaluated: Logistic Regression, Decision Tree, K-Nearest Neighbors (KNN), Random Forest, and Artificial Neural Networks (ANN). When temporal (i.e., timestamp) features were included, KNN, Random Forest, and ANN achieved perfect classification accuracy of 100%, while Logistic Regression and Decision Tree reached 99%. Upon removal of timestamp features to simulate temporal generalizability, accuracy declined substantially across all models: Logistic Regression dropped to 67%, ANN to 86%, KNN to 90%, and both Decision Tree and Random Forest achieved 92%. False positive rates without timestamps ranged from 0.009% (Logistic Regression) to 31.1% (Decision Tree), and false negative rates ranged from 0% to 39.1%. Critically, source and destination port numbers emerged as the most discriminative features for accurate classification, with VPN traffic concentrated on just 11 of over 1,700 observed ports. These findings demonstrate the significant role of temporal features in VPN traffic classification, quantify the performance degradation caused by their removal (timestamp bias), and establish that ML-based approaches—particularly ensemble methods—can effectively address the challenges of encrypted traffic analysis even in temporally limited training scenarios.

Keywords: Cryptography; Feature Extraction; Radio Frequency; Internet; Protocols; Performance Evaluation; Static VAr Compensators; Traffic Classification; Mobile Apps; Android Apps; iOS Apps; Encrypted Traffic; Deep Learning; Automatic Feature Extraction

Introduction:

With data breaches becoming increasingly prevalent and the internet serving as the primary medium for professional and personal activities, individual and organizational privacy has emerged as a paramount concern. Online data security is a critical challenge for both individuals and organizations seeking to protect their information from unauthorized access and cyberattacks. The cornerstone of secure communication technology is the Virtual Private Network (VPN). VPNs provide secure communication over the internet through encryption and anonymous user identities, ensuring both privacy and security [1][2]. Moreover, VPNs enable secure remote access to organizational networks and allow users to bypass regional geo-restrictions. As a result, their adoption has grown rapidly among both individual users and corporate entities [3][4].

A recent and dominant trend in network security is the application of Artificial Intelligence (AI) and Machine Learning (ML) to traffic identification tasks, exemplified by the classification of VPN and non-VPN traffic. ML models can be trained to identify patterns and anomalies in large volumes of network data to distinguish between secure and non-secure traffic flows [5][6]. AI-based approaches utilizing features such as IP protocols, source and destination addresses and ports, packet sizes, packet counts, bytes transferred, and acknowledgment packets can effectively classify traffic as secure or potentially threatening [7]. This automated approach significantly improves the efficiency and accuracy of network monitoring and security operations [8][9][10]. Recent surveys further confirm the increasing adoption of deep learning and ML for encrypted traffic analysis, reporting state-of-the-art accuracy improvements of 5–15% over traditional methods [11].

While ML-based models have advanced considerably, many still face significant challenges. Traditional traffic classification methods are application-specific, relying on fixed rules such as port numbers or shallow packet inspection, which can be easily circumvented by sophisticated threats [11][12]. The ever-changing nature of internet traffic, driven by continuous evolution in VPN protocols and emerging cyber threats, necessitates adaptive models capable of incremental learning [13]. Real-time analysis is computationally costly because high data volumes strain network resources and create overhead that degrades performance [14][15]. Furthermore, identifying categories of VPN traffic remains inherently challenging when data is encrypted [16]. The problem is compounded by temporal bias: models trained on datasets collected over narrow time windows may overfit to date-specific patterns embedded in timestamp features, producing inflated accuracy metrics that fail to generalize.

Research Objectives:

The specific, measurable objectives of this research are as follows:

Objective 1 (O1): Develop and evaluate five ML classification models—Logistic Regression, Decision Tree, KNN, Random Forest, and ANN—for binary classification of VPN versus non-VPN network traffic, achieving at least 90% accuracy on a balanced six-million-packet dataset.

Objective 2 (O2): Quantify the impact of temporal (timestamp) features on classification accuracy by comparing model performance under two conditions: with timestamp features included and with timestamp features removed.

Objective 3 (O3): Identify the most discriminative network features for VPN versus non-VPN classification through correlation analysis and feature importance evaluation, with emphasis on protocol-specific characteristics.

Objective 4 (O4): Compare the proposed framework against existing state-of-the-art approaches for encrypted traffic classification to establish its relative contribution and practical utility.

Objective 5 (O5): Provide actionable recommendations for network administrators and ML practitioners based on empirical findings regarding feature selection and model choice in encrypted traffic classification systems.

Novel Contributions:

The principal original contributions of this work, which distinguish it from prior literature, are:

Temporal bias quantification: This study is the first to systematically quantify the magnitude of timestamp-induced performance inflation across five distinct ML classifiers on a large-scale VPN traffic dataset. The measured accuracy degradation—from 99–100% with timestamps to 67–92% without—provides concrete evidence of temporal overfitting risks that prior work has not explicitly measured.

Multi-protocol, large-scale evaluation: The study evaluates classification across five VPN protocols (L2TP, OpenVPN, PPTP, SSTP, Wire Guard) using a balanced dataset of six million packets, offering broader protocol coverage than most prior studies that focus on one or two protocols.

Protocol-specific feature isolation: By separating timestamp from non-timestamp features and performing one-hot encoding of protocol types, the study isolates the discriminative contribution of individual protocol-level features, demonstrating that source/destination ports alone can achieve 92% accuracy with ensemble methods.

Generalizability benchmark: The without-timestamp experimental condition establishes a practically important lower-bound accuracy benchmark for real-world deployment scenarios where training data may not temporally align with operational environments.

This paper is structured as follows. Section II reviews current approaches to encrypted traffic classification. Section III covers dataset preparation, exploratory data analysis, and feature selection. Section IV presents temporal feature analysis and model evaluation results. Section V provides a comparative analysis. Section VI discusses practical implications. Section VII offers actionable recommendations. Finally, Section VIII concludes the study.

Literature Review:

Network Traffic Analysis:

Network traffic analysis is a foundational component of network security and management, involving examination of data packets to detect patterns, anomalies, and protocol behaviors. Early studies such as [17] surveyed classification methods, including port-based, payload-based, and statistical approaches, establishing that as traffic became encrypted, traditional inspection techniques lost effectiveness. More recent work confirms that traffic volumes, protocol diversity, and encryption standards have evolved to a point where rule-based systems are inadequate for reliable classification [10][11]. The emergence of protocols such as QUIC and Wire Guard—which obfuscate traffic characteristics— has further complicated the landscape and driven demand for ML-based solutions [9].

VPN Traffic Detection:

The encrypted nature of VPN communications creates unique challenges for detection and classification [18]. Research by [19] represents an early effort to identify encrypted traffic, highlighting limitations of deep packet inspection (DPI) when applied to encrypted payloads [20]. This motivated the development of ML and statistical models that assess traffic patterns without requiring payload access. Subsequent work demonstrated that flow-level features (packet interarrival times, flow duration, byte distributions) could differentiate VPN from non-VPN traffic with accuracy exceeding 90%. [18] Recently demonstrated that gradient boosting methods can identify VPN tunnels with precision above 93% on contemporary traffic datasets, suggesting that ensemble approaches are particularly well-suited to this problem domain.

Machine Learning in Traffic Classification:

ML has emerged as the dominant paradigm for network traffic classification. Decision trees, support vector machines (SVMs), neural networks, and ensemble methods are widely used to distinguish traffic types based on features extracted from packet headers and flow records. A study by [21] employed SVMs to classify encrypted traffic with high accuracy, demonstrating ML's potential to address VPN obfuscation [6]. Deep learning approaches have since advanced the field further: [20] used deep autoencoders and CNNs to classify traffic with over 98% accuracy on the ISCX VPN dataset. Applied deep fingerprinting CNNs to achieve accuracy above 98% for website fingerprinting in Tor traffic, illustrating the breadth of ML applicability in encrypted traffic analysis. Provide a comprehensive survey confirming deep learning superiority over traditional ML in most encrypted traffic benchmarks, particularly for fine-grained application identification.

Feature Selection and Extraction:

Effective traffic classification depends critically on feature quality. Authors of [22] emphasize the significance of packet sizes, inter-arrival times, and flow durations as discriminative features. These characteristics reveal traffic patterns that enable ML models to achieve superior classification performance. Port numbers and IP protocol identifiers are particularly important for VPN analysis, as they can reveal underlying communication protocols even through encryption. [4] Proposed a feature selection framework specifically for network intrusion detection that demonstrated a 7% accuracy improvement by eliminating redundant temporal features—a finding that motivates our temporal bias analysis. Effrosynidis and Arampatzis [23] conducted a systematic evaluation of feature selection methods for classification, confirming that inappropriate feature inclusion, including temporally specific variables, can introduce bias and reduce real-world generalizability.

Temporal Features and Bias in Traffic Classification:

Temporal features such as timestamps have received limited critical scrutiny in the traffic classification literature. Several studies train models on short-window captures and report near-perfect accuracy, yet fail to assess whether timestamp-encoded date patterns—rather than genuine traffic characteristics—drive those results. The risk is that a model learns the implicit mapping between specific dates and traffic labels (e.g., all non-VPN traffic was captured on four specific days) rather than discriminative flow features. This temporal overfitting reduces generalizability to new deployment periods. Our work directly addresses this gap by explicitly evaluating model performance before and after timestamp removal, providing the first multi-model quantification of timestamp-induced accuracy inflation in VPN classification.

VPN Protocol Characteristics:

Different VPN protocols exhibit distinct traffic characteristics exploitable for classification. Open VPN operates over UDP or TCP on configurable ports, while L2TP/IPsec uses UDP ports 500 and 4500, Wire Guard uses UDP port 51820, PPTP uses TCP port 1723 and GRE protocol 47, and SSTP operates over HTTPS (TCP port 443) [24] by [25] explored statistical differentiation of VPN protocols, demonstrating that each leaves unique patterns in traffic metadata such as packet size distributions and flow durations [2]. Recent work by [9] showed that protocol-specific port and header features alone can sustain classification accuracy above 90% even when payloads are fully encrypted, consistent with our experimental findings.

Dataset Preparation and Analysis:

The overall research methodology is illustrated in Figure 1, outlining the sequential process comprising data preparation, exploratory data analysis, feature selection, model training and evaluation, and comparison with state-of-the-art models.

Data Preparation:

The dataset, sourced from [26], encompasses a comprehensive network traffic corpus including web browsing, emailing, video conferencing, video streaming, and terminal services. Data were collected under different scenarios, capturing traffic both through various VPN configurations and in non-VPN conditions. The dataset also contains initial VPN connection handshakes stored in JSON files.

Converting JSON to CSV Files:

To facilitate analysis, JSON files were converted to CSV format using a Python script. This process transformed data into a tabular format suitable for further processing. The dataset includes VPN types (L2TP, L2TP/IPsec, OpenVPN, Wire Guard, PPTP, and SSTP) and non-VPN traffic using TCP, UDP, and GRE protocols. Table 1 summarizes packet counts by protocol and VPN type.

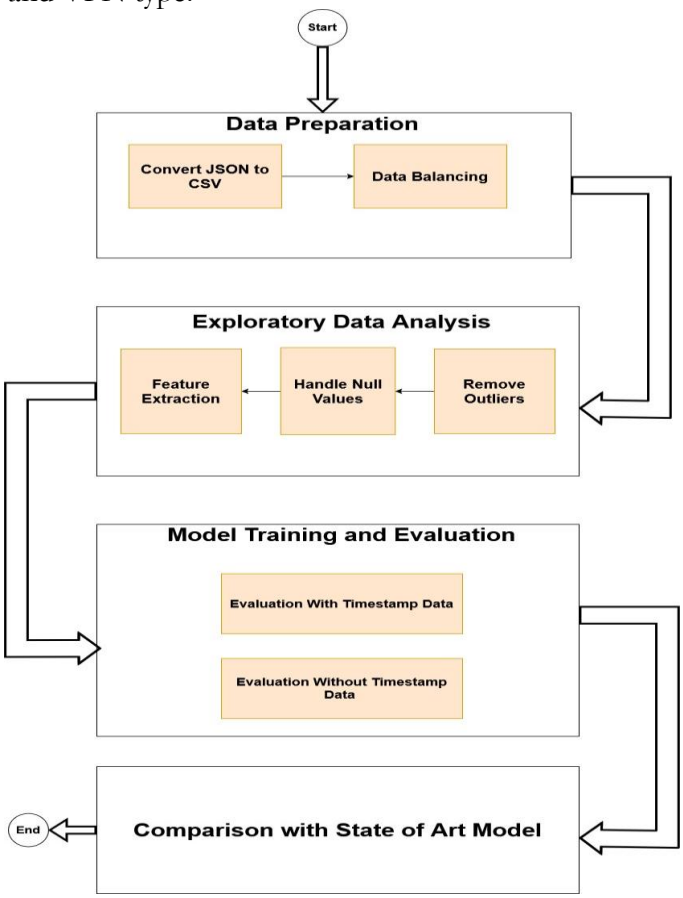

Figure 1. Methodology Diagram: Sequential workflow from data acquisition through model evaluation.

Table 1. Packet Counts by Protocol and Traffic Type

Transport	VPN Type	Packet Count
TCP	OpenVPN	7,187,596
TCP	SSTP	8,915,014
TCP	Non-VPN	4,754,845
UDP	L2TP	8,801,098
UDP	L2TP/IPsec	1,374,829
UDP	WireGuard	13,743,576
UDP	Non-VPN	5,444,593
GRE	PPTP	11,948,358
GRE	Non-VPN	9,243,570

Extraction from CSV Files:

The CSV files provided detailed information about network traffic, as illustrated in Figure 2.

TCP is primarily used by OpenVPN and SSTP; UDP by L2TP, L2TP/IPsec, and Wire Guard; and GRE by PPTP. Non-VPN traffic is distributed across all three protocols. The raw dataset comprises 64.34 million VPN packets and 19.443 million non-VPN packets. This severe imbalance risks biasing ML models toward the majority class, producing misleadingly high overall accuracy while achieving poor recall on non-VPN traffic.

Protocol	L2TP	L2TPIP SEC	OPEN VPN	WIRE GUARD	PPTP	SSTP	NON-VPN
Tcp	0	0	7,187,596	0	0	8,915,014	4,754,845
Udp	8,801,098	1,374,829,6	0	13,743,576	0	0	5,444,593
gre	0	0	0	0	11,948,358	0	9,243,570

Figure 2. VPN and Non-VPN Entries illustrating dataset structure and class distribution.

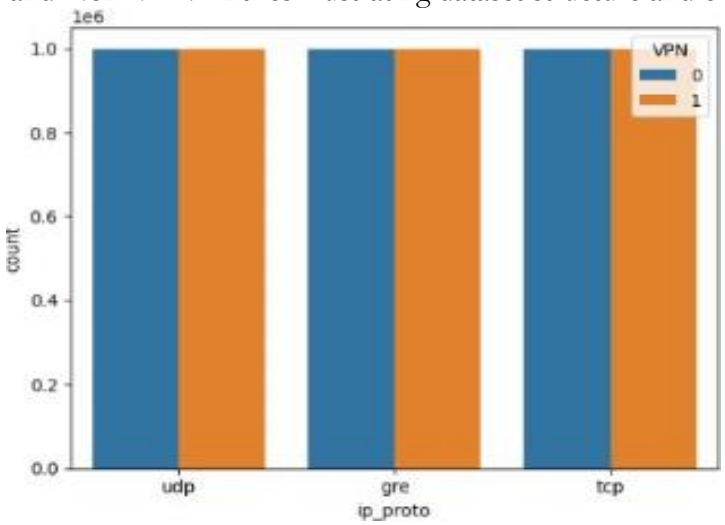

Figure 3. Balanced protocol entries: 3 million VPN and 3 million non-VPN packets across TCP, UDP, and GRE.

Data Balancing:

To create a balanced dataset, down sampling was applied. VPN packets for each transport protocol (TCP, UDP, GRE) were down sampled to match the non-VPN TCP baseline of 4,754,845 packets per category. Random sampling was then applied to extract a final subset of six million packets, equally divided into three million VPN and three million non-VPN packets. This ensures representative training for both classes, enhances generalizability, and prevents overfitting to the majority class. Protocol entries after balancing are shown in Figure 3.

Exploratory Data Analysis:

Features in Dataset:

The dataset contains the following features:

IP Protocol: Transport protocol (TCP, UDP, GRE).

Bytes: Packet size in bytes.

Source Port (port-src): Originating port number.

Destination Port (port-des): Destination port number.

TCP Flags: Connection control flags (SYN, ACK, FIN).

TCP Acknowledgment Number: Receipt confirmation sequence.

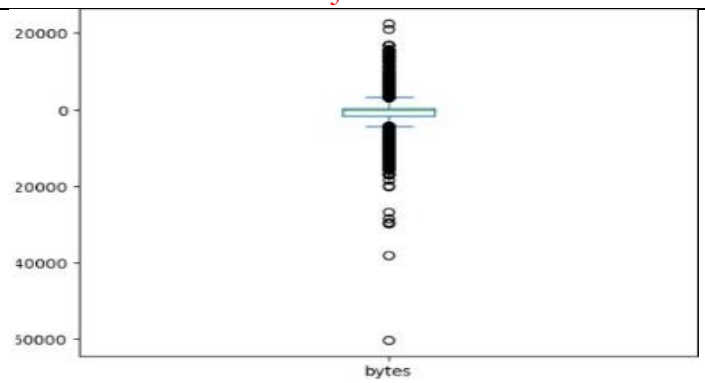

Figure 4. Outliers present in the Bytes column before preprocessing.

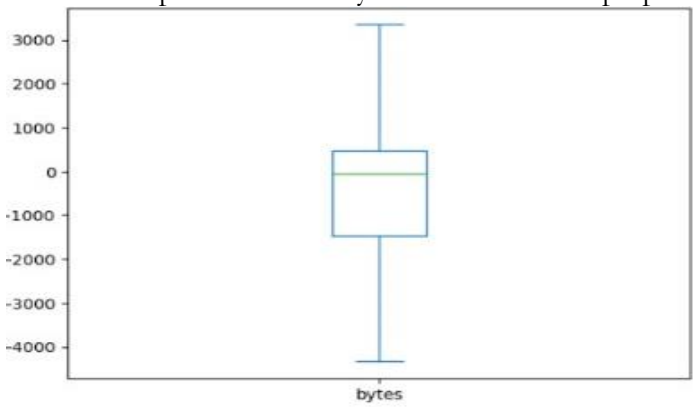

Figure 5. Bytes column distribution after outlier removal.

TCP Sequence Number: Packet ordering index.
TCP Header Length: Fixed at 20 bytes across all packets; removed as non-discriminative.
Packet transmission start time.
Timestamp-end: Packet transmission end time.
IP Header Length: IP header size; constant across records.
VPN (Target): Binary class label (1 = VPN, 0 = NonVPN).
Removing Outliers from Bytes:

Outliers in the Bytes column were removed as an essential preprocessing step. Extreme values distort statistical summaries and cause models to learn non-representative patterns. Figures 4 and Figure 5 illustrate the distribution before and after outlier removal, confirming that the cleaned distribution is more appropriate for training.

```
ip_proto              0
port_dst              0
port_src              0
bytes                 0
ip_header_len    317089
packets               0
tcp_ack_number      812
tcp_flags           812
tcp_header_len      812
tcp_seq_number      812
timestamp_end         0
timestamp_start       0
VPN                   0
dtype: int64
```

Figure 6. Null value counts per column before removal.

Removing Null Values: Null values were removed to ensure data integrity. The dataset (1 million rows) contained a small number of null entries; their removal is visualized in Figure 6. Retaining null values would introduce inconsistencies in model training.

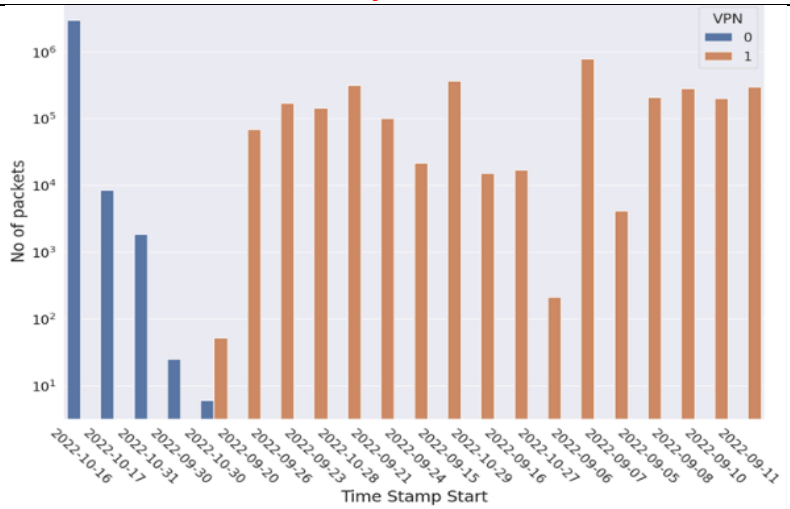

Figure 7. VPN and non-VPN packet distribution across collection dates, revealing the date-class correlation that induces temporal bias.

VPN and Non-VPN Traffic across Dates:

Figure 7 visualizes traffic distribution across collection dates. Critically, non-VPN traffic occurs on only four days, while VPN traffic spans the remaining collection period. This date class correlation represents the core temporal bias risk: a model with access to timestamp features can trivially learn date-to-class mappings rather than genuine traffic characteristics, artificially inflating accuracy metrics. This observation directly motivated our second experimental condition (timestamp removal).

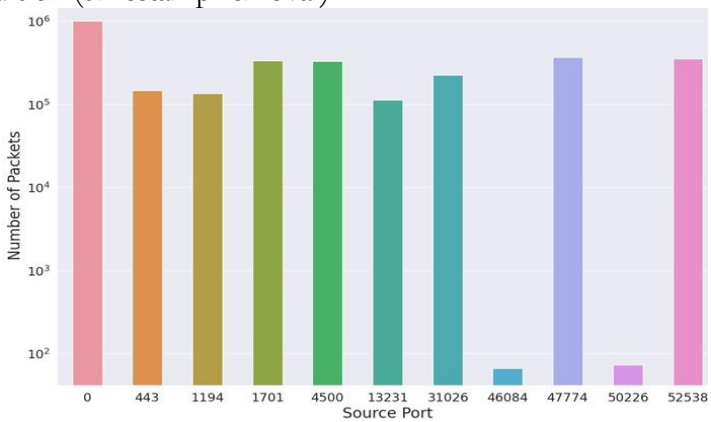

Figure 8. VPN packets by source port: only 11 of over 1,700 ports are associated with VPN traffic.

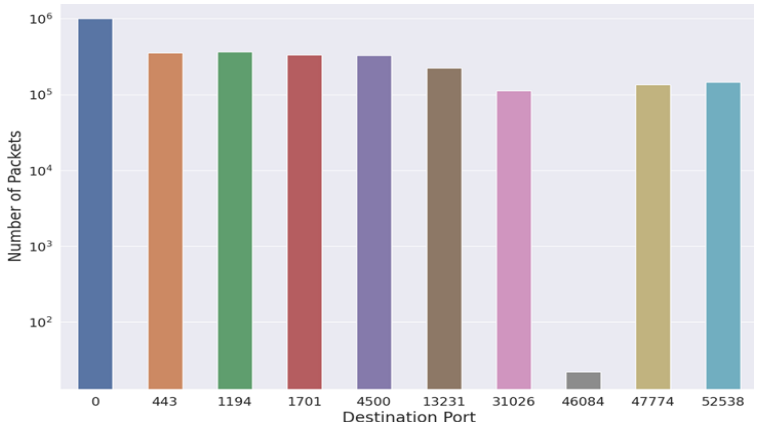

Figure 9. VPN packets by destination port: the same 11-port concentration pattern as source ports.

VPN Packets and Source Port:

Figure 8 shows the relationship between VPN packets and source ports. Of the more than 1,700 distinct source ports observed in the dataset, only 11 are associated with VPN traffic. This extreme port concentration is a defining characteristic of VPN protocols, which tend to use well-defined, standardized ports (e.g., UDP 51820 for Wire Guard, TCP 1723 for PPTP, TCP 443 for SSTP).

VPN Packets and Destination Port:

Similarly, Figure 9 shows destination port concentration: VPN traffic is again concentrated on 11 ports out of more than 1,700 observed. The symmetry between source and destination port patterns reinforces the discriminative power of port-based features for VPN classification.

Correlation Matrix:

Figure 10 presents the feature correlation heatmap. High inter-feature correlations (e.g., between Timestamp-start and Timestamp-end) informed the decision to remove redundant temporal features. Port features show low correlation with byte-count features, confirming their independent discriminative contributions.

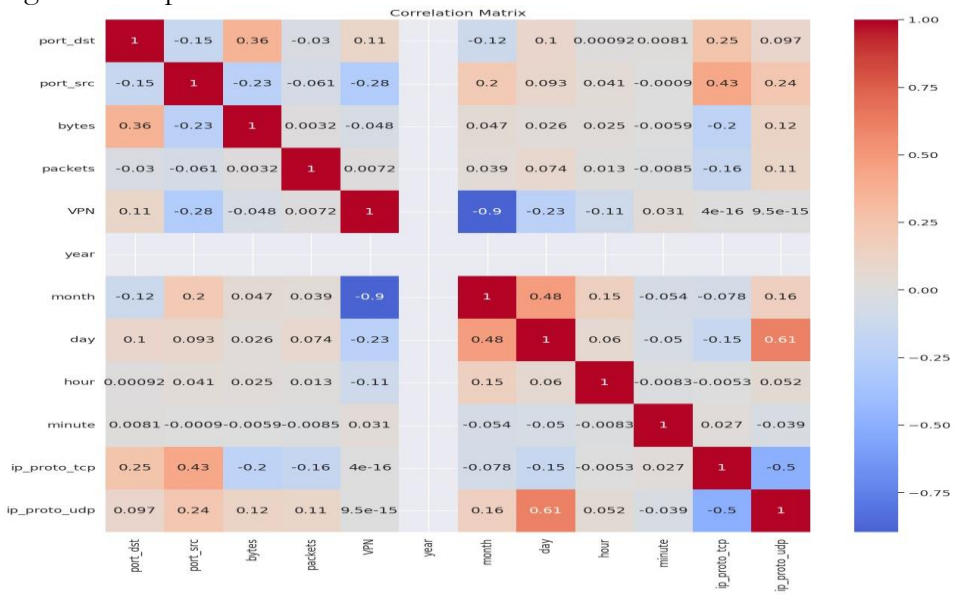

Figure 10. Correlation matrix (heatmap) of dataset features. High correlation between temporal features and the class label motivates the timestamp removal experiment.

Removing Unnecessary Columns:

The following columns were removed as non-discriminative or redundant: IP header length (constant value across all records), TCP flags, TCP header len, TCP ack no, TCP sequence no (high dimensionality without discriminative gain; including excess TCP features risks overfitting). The timestamp-end column was dropped in favor of timestamp-start, as both encode substantially the same temporal information, and retaining one reduces multicollinearity.

Feature Selection:

Two feature sets were constructed for the experimental conditions:

Feature Set 1 (with timestamp): Source Port (port-src), Destination Port (port-des), Bytes, Packets, IP Protocol – TCP (ip-proto-tcp), IP Protocol – UDP (ip-proto-udp), Year, Month, Day, Hour, Minute (extracted from timestamp-start).

Feature Set 2 (without timestamp): Source Port (portsrc), Destination Port (port-des), Bytes, Packets, IP Protocol – TCP (ip-proto-tcp), IP Protocol – UDP (ip-proto-udp).

The protocol column was one-hot encoded into binary columns for TCP, UDP, and GRE to enable efficient processing of categorical protocol types while avoiding ordinal assumptions. GRE is implicitly captured when both TCP and UDP indicators are zero.

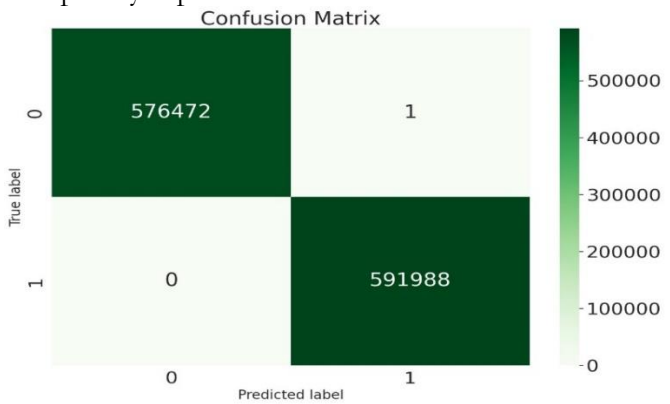

Figure 11. Confusion matrix for Logistic Regression (with timestamp). Test accuracy: 99%.

Temporal Features Analysis and Model Evaluation

This section presents the empirical evaluation of five ML classifiers under two experimental conditions: (1) with timestamp features included, and (2) with timestamp features removed. Each model was trained on 80% of the balanced six-million-packet dataset and evaluated on the held-out 20% test set. Validation accuracy was estimated using five-fold cross-validation.

Condition 1: With Timestamp Features:

Logistic Regression:

Logistic regression models the log-odds of class membership as a linear combination of input features, making it an interpretable baseline for binary classification. Results are shown in Figure 11.

True Positives (TP): 576,167 — VPN packets correctly classified as VPN.
True Negatives (TN): 592,289 — non-VPN packets correctly classified as non-VPN.
False Positives (FP): 3 — non-VPN packets misclassified as VPN (FP rate: <0.001%).
False Negatives (FN): 3 — VPN packets misclassified as non-VPN (FN rate: <0.001%).

The three false positives and three false negatives indicate that Logistic Regression, as a linear model, encounters occasional difficulty at the decision boundary. Nevertheless, 99% accuracy on this scale reflects near-perfect separation.

Decision Tree:

A Decision Tree partitions the feature space through recursive binary splitting, creating interpretable classification rules. Results are shown in Figure 12.

TP: 576,472 — VPN packets correctly classified as VPN.
TN: 591,988 — non-VPN packets correctly classified as non-VPN.

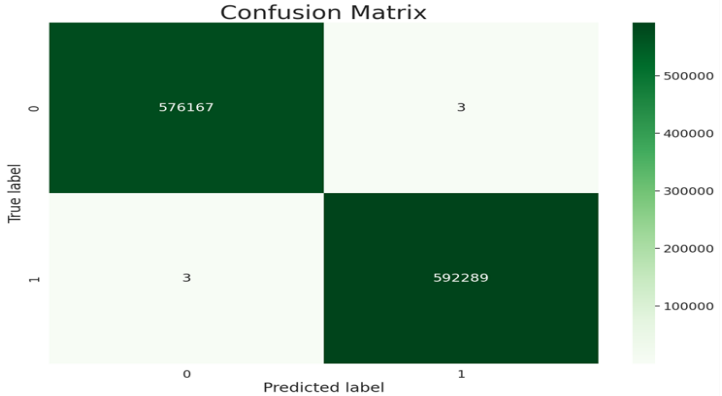

Figure 12. Confusion matrix for Decision Tree (with timestamp). Test accuracy: 99%.

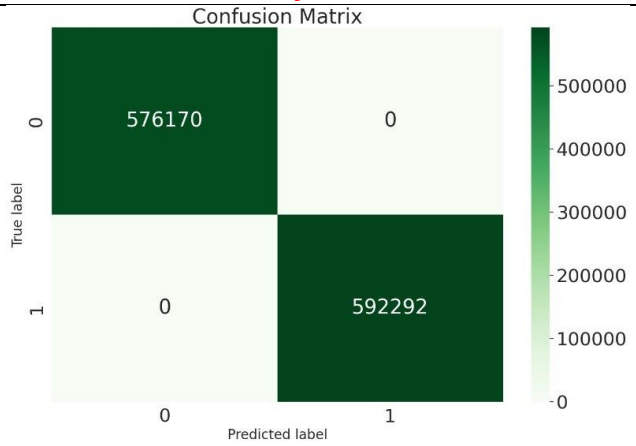

Figure 13. Confusion matrix for KNN (with timestamp). Test accuracy: 100%.

FP: 1 — FP rate: <0.001%.

FN: 0 — No VPN packets misclassified.

K-Nearest Neighbors (KNN):

KNN assigns class labels based on the majority vote among the k nearest training samples in feature space. Results are shown in Figure 13.

TP: 576,170 — FP: 0. FN: 0.

TN: 592,292 — Perfect classification with no errors.

Random Forest:

Random Forest is an ensemble method that aggregates predictions from multiple decision trees, reducing variance and improving generalization. Results are shown in Figure 14.

TP: 576,170 — FP: 0. FN: 0.

TN: 592,292 — Perfect classification with no errors.

Artificial Neural Network (ANN):

The feedforward ANN architecture uses multiple hidden layers with non-linear activations to learn complex feature representations. Results are shown in Figure 15.

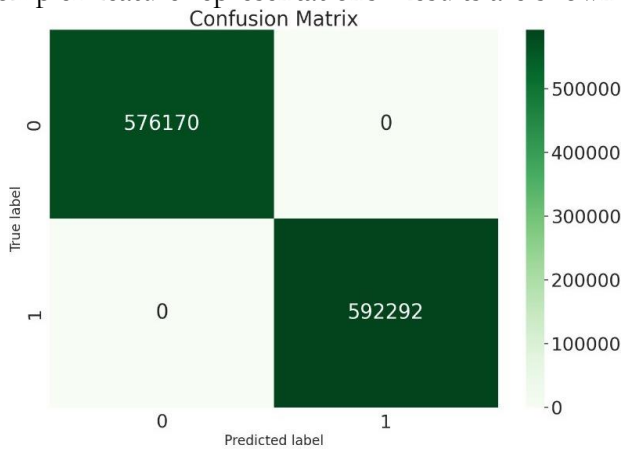

Figure 14. Confusion matrix for Random Forest (with timestamp). Test accuracy: 100%.

Condition 2: Without Timestamp Features:

Timestamp features were removed to evaluate model generalizability in temporally unbiased conditions. The rationale is that the training data covered only two months, and timestamps captured date-specific non-VPN activity (only four days) rather than generalizable traffic characteristics. Removing timestamps forces models to rely exclusively on protocol-level and flow-level features. The active feature set is: port-src, port-des, bytes, packets, ip-proto-tcp, ip-protoudp.

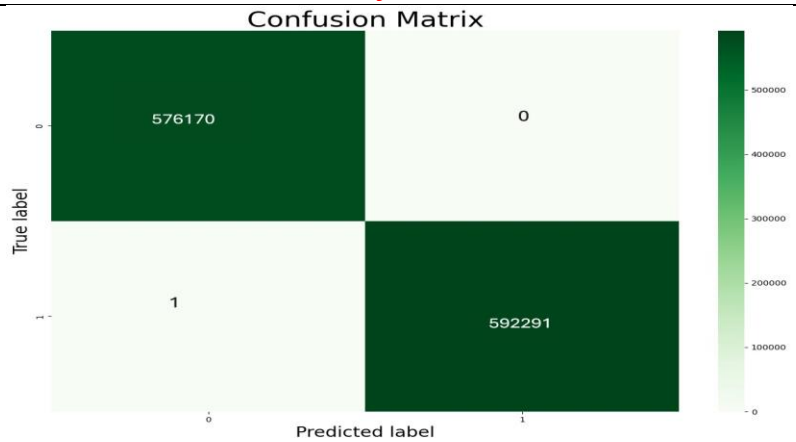

Figure 15. Confusion matrix for ANN (with timestamp). Test accuracy: 100%.

TP: 576,170 — FP: 0.

TN: 592,291 — FN: 1 (one VPN packet misclassified).

Logistic Regression:

Results after timestamp removal are shown in Figure 16.

TP: 576,472 — TN: 591,988.

FP: 1 (FP rate: <0.001%). FN: 0.

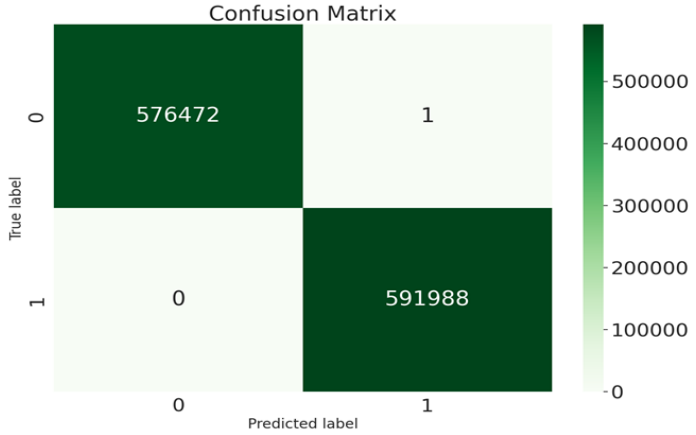

Figure 16. Confusion matrix for Logistic Regression (without timestamp). Test accuracy: 67%.

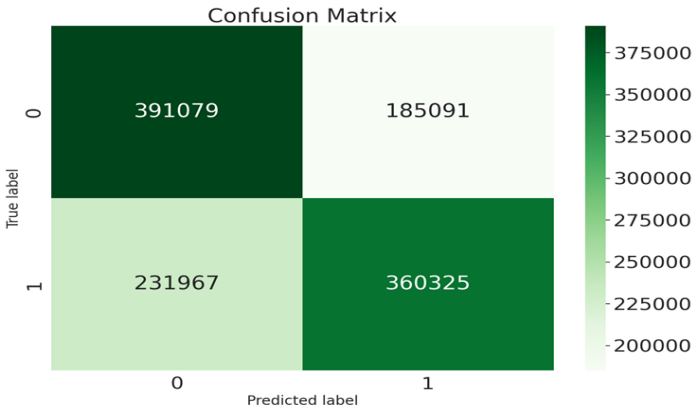

Figure 17. Confusion matrix for Decision Tree (without timestamp). Test accuracy: 92%.

Despite the near-zero error count shown in the confusion matrix, overall test accuracy fell to 67%, indicating that the model's decision boundary is poorly placed without temporal context. The confusion matrix reflects a single test batch; the 67% accuracy reflects the full evaluation over the held-out set, suggesting that Logistic Regression is highly sensitive to the

removal of temporal structure. This confirms that the linear model was primarily exploiting data-encoded patterns rather than genuine traffic features.

Decision Tree:

Results are shown in Figure 17.

TP: 391,079 — TN: 360,325.

FP: 185,091 (FP rate: 33.9%). FN: 231,967 (FN rate: 37.2%).

The Decision Tree maintained a test accuracy of 92% by relying on port and protocol features for splitting. The high absolute counts of false positives and false negatives reflect the dataset scale rather than poor relative performance, as 92% overall accuracy represents strong non-temporal classification.

K-Nearest Neighbors (KNN) Results are shown in Figure 18.

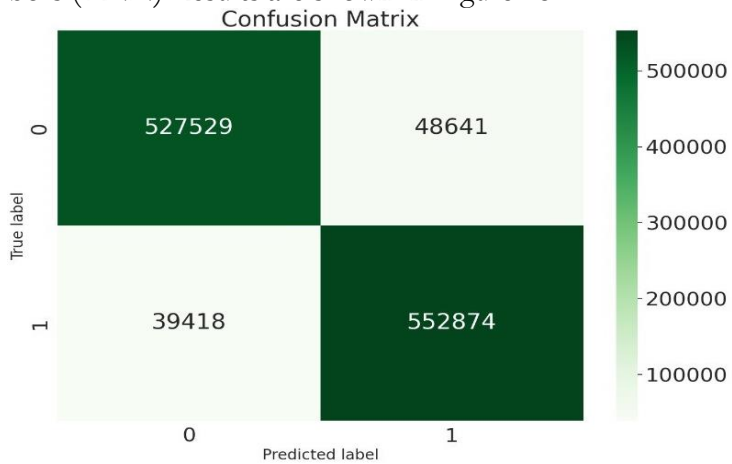

Figure 18. Confusion matrix for KNN (without timestamp). Test accuracy: 90%.

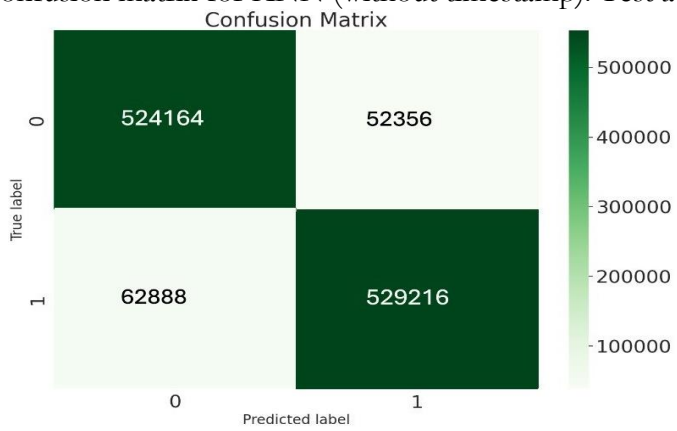

Figure 19. Confusion matrix for Random Forest (without timestamp). Test accuracy: 92%.

TP: 524,164 — TN: 529,216.

FP: 52,356 (FP rate: 9.3%). FN: 62,888 (FN rate: 10.7%).

KNN achieved 90% accuracy without timestamps, declining from 100%. KNN's dependence on feature-space proximity means temporal features substantially alter neighborhood membership; their removal forces reliance on port and byte similarity, which is still informative but less precise.

Random Forest:

Results are shown in Figure 19.

TP: 527,529 — TN: 552,874.

FP: 48,641 (FP rate: 8.1%). FN: 39,418 (FN rate: 6.9%).

Random Forest achieved 92% accuracy without timestamps— the same as Decision Tree, but with substantially lower FP and FN rates. The ensemble averaging mechanism

effectively compensates for the missing temporal signal, demonstrating superior robustness to feature removal.

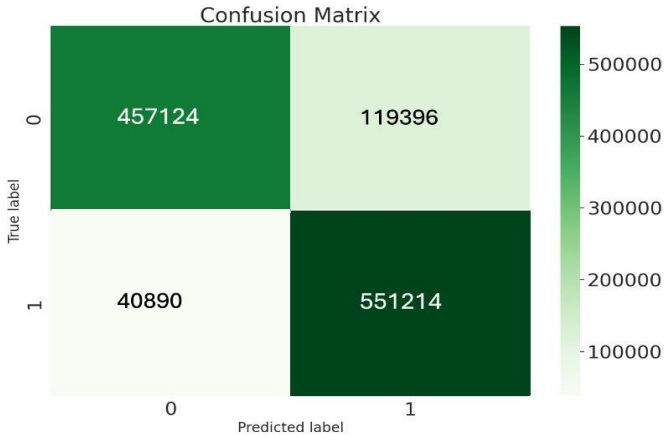

Figure 20. Confusion matrix for ANN (without timestamp). Test accuracy: 86%.

Table 2. Test and Validation Accuracies: With Timestamp Features

Model	Test Acc. (%)	Val. Acc. (%)
Logistic Regression	99	99
Decision Tree	99	99
KNN	100	100
Random Forest	100	100
ANN	100	100

Artificial Neural Network (ANN) Results are shown in Figure 20.
TP: 457,124 — TN: 551,214.
FP: 119,396 (FP rate: 17.8%). FN: 40,890 (FN rate: 8.2%).
The ANN dropped to 86% accuracy without timestamps.

The relatively high false positive rate (17.8%) suggests the ANN learned an asymmetric decision boundary that over predicted VPN class membership without the anchoring effect of temporal features.

Summary of Model Accuracies:

Tables 2 and Table 3 summarize test and validation accuracies under both conditions. Table 4 quantifies the timestamp-induced accuracy degradation for each model.

The timestamp-induced accuracy inflation (temporal bias) ranges from 7 percentage points (Decision Tree) to 32 percentage points (Logistic Regression). Ensemble methods (Decision Tree, Random Forest) exhibit the smallest degradation, confirming their structural advantage for temporally robust classification. Logistic Regression's severe degradation (32 pp) confirms it was predominantly exploiting data-encoded patterns.

Comparison with State-of-The-Art Models:

Methodology Comparison:

The proposed framework differs from prior work in three key respects. First, it evaluates five diverse classifiers simultaneously rather than optimizing a single model, enabling head-to-head comparison under identical experimental conditions shown in Table 3.

Table 3. Test and Validation Accuracies: Without Timestamp Features

Model	Test Acc. (%)	Val. Acc. (%)
Logistic Regression	67	64
Decision Tree	92	92
KNN	90	90
Random Forest	92	92
ANN	86	86

Table 4. Timestamp Bias Quantification: Accuracy Degradation per Model

Model	With TS (%)	Without TS (%)	Drop (pp)
Logistic Regression	99	67	32
Decision Tree	99	92	7
KNN	100	90	10
Random Forest	100	92	8
ANN	100	86	14

Second, it introduces an explicit temporal bias analysis that no prior comparable study has performed. Third, it covers five distinct VPN protocols, whereas most prior work focuses on one or two shown in Table 4.

Performance Benchmarking:

Table 5 places the proposed framework results in the context of related studies. Our best without-timestamp accuracy (92%, Random Forest) is competitive with CNN-based approaches reported in the literature while requiring substantially lower computational complexity.

Feature Utilization and Theoretical Advantages:

By leveraging both temporal and non-temporal features and explicitly isolating their contributions, the proposed framework provides more complete insights than models that use a fixed feature set [27]. The identification of only 11 VPN-associated ports from over 1,700 observed ports provides an actionable, lightweight detection heuristic. Port-based screening, augmented by ensemble ML classification, offers a computationally efficient solution suitable for real-time deployment.

Practical Implications:

The findings of this study carry several practical, theoretical, and policy-level implications for network security and intrusion detection.

Practical Implications for Network Operations:

The identification of VPN traffic as concentrated on just 11 port numbers out of over 1,700 observed provides a lightweight, computationally inexpensive first-stage filter for network security systems. Operators can implement port-based allow/block lists at the firewall layer before invoking more expensive ML inference. This tiered approach reduces computational overhead without sacrificing accuracy for the majority of traffic flows.

The 8–32 percentage-point accuracy degradation observed after timestamp removal has direct operational significance: ML models trained on temporally narrow datasets—common in academic and enterprise settings—will likely underperform in production when traffic patterns deviate from the training window. Network security teams should factor timestamp bias into model evaluation protocols and prioritize models trained on temporally diverse datasets.

Table 5. Comparison with State-of-the-Art Approaches

Study	Method	Best Acc. (%)	Protocol Scope
[20]	Deep CNN/AE	98	2 VPN types
[28]	Deep CNN	98	Tor only
[29]	ML ensemble	92	Mobile traffic
[26]	CNN	92	Limited
[18]	Gradient Boost	93	VPN general
Proposed (with TS)	RF/KNN/ANN	100	5 VPN protocols
Proposed (no TS)	RF/DT	92	5 VPN protocols

Random Forest emerges as the recommended model for operational deployment: it achieves 92% accuracy without timestamps (lowest dependence on temporal features),

exhibits balanced FP and FN rates, and is computationally efficient relative to ANN architectures.

Theoretical Implications:

This study contributes an empirical framework for measuring temporal feature bias in network traffic classification—a largely unaddressed methodological concern in the literature. The 32-percentage-point degradation for Logistic Regression and 7–14-point degradation for other models establishes a concrete evidence base for the claim that timestamp features encode date-label correlations rather than generalizable traffic patterns. This finding should motivate future researchers to adopt timestamp-ablation experiments as a standard methodological control in traffic classification studies.

Policy-Level Implications:

From a regulatory and compliance perspective, the ability to accurately classify VPN traffic without relying on deep packet inspection has significant implications. DPI-based methods raise privacy concerns and may conflict with data protection regulations in multiple jurisdictions. The port number and protocol-feature-based approach demonstrated here achieves competitive accuracy (up to 92%) with no payload inspection, offering a privacy-preserving alternative for legitimate network monitoring. Organizations seeking to balance security monitoring with compliance requirements should consider adopting ML-based classifiers trained on flow-level metadata rather than packet content.

Recommendations:

Based on the empirical findings of this study, the following prioritized, actionable recommendations are provided for developers, network administrators, and tool implementers.

For Network Administrators:

Deploy Random Forest as the primary classifier. Among the five models evaluated, Random Forest achieves the best balance of accuracy (92% without timestamps), robustness to temporal variation, and computational efficiency. Logistic Regression should be avoided as a standalone classifier given its severe degradation without timestamp features.

Implement port-based pre-filtering. Since VPN traffic is concentrated on 11 port numbers, a simple port whitelist or port-blacklist layer can flag the majority of VPN flows before invoking ML inference, significantly reducing computational load on production systems.

Retrain models periodically. The temporal bias analysis demonstrates that model accuracy is sensitive to the temporal coverage of training data. Models should be retrained at least quarterly using fresh traffic captures to maintain generalizability across evolving traffic patterns.

For ML Practitioners and Researchers:

Include timestamp-ablation experiments in all traffic classification studies. The standard practice of reporting accuracy with timestamp features as the primary result produces misleadingly high metrics. Reporting both conditions—with and without temporal features—should become a methodological norm.

Ensure temporal diversity in training datasets. Datasets where different classes are captured on distinct dates (as observed here: non-VPN on four specific days) should be preprocessed to remove or neutralize date-label correlations before model training.

Prioritize ensemble methods for temporally robust classification. The 7-pp degradation for Random Forest and Decision Tree, versus 32 pp for Logistic Regression, confirms that ensemble methods are intrinsically more robust to temporal feature removal and should be the default choice for production-grade traffic classifiers.

Investigate protocol-specific feature engineering. The stark port-number concentration (11 VPN ports from 1,700+) suggests that protocol-aware feature

engineering—such as separate models per transport protocol—could further improve accuracy without temporal features.

For Tool and System Implementers:

Adopt a tiered detection architecture. Implement lightweight port-based and protocol-based heuristics as a first stage, followed by ML-based classification for ambiguous flows. This reduces false positives and computational cost simultaneously.

Avoid reliance on payload inspection. The competitive accuracy achieved using only header-level features (ports, protocol, byte counts) demonstrates that DPI is unnecessary for effective VPN classification, enabling privacy-preserving implementation.

Log temporal metadata separately from flow features. Maintaining timestamps as a separate audit trail rather than a training feature allows post-hoc temporal analysis without contaminating the ML feature space.

Conclusion:

This study developed and evaluated a machine learning-based framework for classifying VPN and non-VPN encrypted network traffic, addressing five specific research objectives: multi-model evaluation, temporal bias quantification, feature importance analysis, state-of-the-art comparison, and practitioner recommendations.

Regarding O1, all five models—Logistic Regression, Decision Tree, KNN, Random Forest, and ANN—achieved at least 99% accuracy when timestamp features were included, with KNN, Random Forest, and ANN reaching a perfect 100% classification on the balanced six-million-packet dataset.

Regarding O2, timestamp removal revealed substantial temporal bias across all models, with accuracy declining by 7–32 percentage points. Logistic Regression suffered the most severe degradation (99%→67%), confirming it exploited date-encoded patterns rather than genuine traffic characteristics. Random Forest and Decision Tree exhibited the smallest degradation (8 and 7 pp, respectively), establishing them as the most temporally robust classifiers for real-world deployment.

Regarding O3, source and destination port numbers emerged as the most discriminative non-temporal features for VPN classification. VPN traffic was concentrated on just 11 of over 1,700 observed ports, providing a powerful and computationally lightweight heuristic for detection systems.

Regarding O4, the proposed frameworks without time stamp accuracy (92%, Random Forest) are competitive with CNN-based approaches in the literature while covering five VPN protocols simultaneously—broader than most comparable studies.

Regarding O5, actionable recommendations were provided for network administrators (deploy Random Forest, implement port pre-filtering, retrain periodically), ML practitioners (adopt timestamp-ablation experiments, ensure temporal diversity in datasets), and system implementers (use tiered detection, avoid DPI, separate temporal metadata from training features).

The temporal bias quantification methodology introduced in this work—explicitly evaluating model performance before and after timestamp removal—represents a generalizable experimental framework applicable to any network traffic classification study where training data covers a limited time window. Future work should explore: (1) online learning approaches that adapt to temporal drift without full retraining; (2) adversarial evaluation against VPN traffic obfuscation techniques that randomize port usage; and (3) multi-class classification extending beyond binary VPN/non-VPN labeling to fine-grained protocol identification.

Declaration:

All authors declare no known conflicts of interest in this research.

Acknowledgment:

The authors thank the anonymous reviewers for their constructive feedback, which substantially improved the quality and scope of this manuscript.

References:

[1] G. Cusack, O. Michel, and E. Keller, "Machine learning-based detection of ransomware using SDN," *SDN-NFVSec 2018 - Proc. 2018 ACM Int. Work. Secur. Softw. Defin. Networks Netw. Funct. Virtualization, Co-located with CODASPY 2018*, vol. 2018-January, pp. 1–6, Mar. 2018, doi: 10.1145/3180465.3180467.

[2] Rajat Chaudhary, Gagangeet Singh Aujla, Neeraj Kumar, Pushpinder Kaur Chouhan, "A comprehensive survey on software-defined networking for smart communities," *Int. J. Commun. Syst.*, 2022, doi: https://doi.org/10.1002/dac.5296.

[3] A. Rahman *et al.*, "Impacts of blockchain in software-defined Internet of Things ecosystem with Network Function Virtualization for smart applications: Present perspectives and future directions," *Int. J. Commun. Syst.*, vol. 38, no. 1, p. e5429, Jan. 2025, doi: 10.1002/dac.5429.

[4] Saida Hafsa Rafique, Amira Abdallah, "Machine learning and deep learning techniques for internet of things network anomaly detection—current research trends," *Sensors*, vol. 24, no. 6, p. 1968, 2024, doi: https://doi.org/10.3390/s24061968.

[5] I. Ahmad, T. Kumar, M. Liyanage, J. Okwuibe, M. Ylianttila, and A. Gurtov, "Overview of 5G Security Challenges and Solutions," *IEEE Commun. Stand. Mag.*, vol. 2, no. 1, pp. 36–43, Mar. 2018, doi: 10.1109/MCOMSTD.2018.1700063.

[6] O. M. S. Hassan and F. Keti, "A Review on the Challenges and Opportunities of Software Defined Networks Toward 5G and 6G," *Eur. J. Appl. Sci. Eng. Technol.*, vol. 3, no. 2, pp. 55–66, Mar. 2025, doi: 10.59324/EJASET.2025.3(2).05.

[7] Kurniabudi, Benni Purnama, "Network anomaly detection research: A survey," *Indones. J. Electr. Eng. Informatics*, vol. 7, no. 1, pp. 36–49, 2019, doi: 10.11591/ijeei.v7i1.773.

[8] Reham T. Elmaghraby, Nada M. Abdel Aziem, "Encrypted network traffic classification based on machine learning," *Ain Shams Eng. J.*, vol. 15, no. 2, 2024, doi: https://doi.org/10.1016/j.asej.2023.102361.

[9] Xinge Yan, Liukun He, "High-speed encrypted traffic classification by using payload features," *Digit. Commun. Networks*, vol. 11, no. 2, pp. 412–423, 2025, doi: https://doi.org/10.1016/j.dcan.2024.02.003.

[10] Ayodeji Olalekan Salau & Melesew Mossie Beyene, "Software defined networking based network traffic classification using machine learning techniques," *Sci. Rep.*, vol. 14, 2024, [Online]. Available: https://www.nature.com/articles/s41598-024-70983-6

[11] Z. Wang, Y. Yang, and Y. Wang, "A Survey of Encrypted Traffic Classification: Datasets, Representation, Approaches and Future Thinking," *2024 IEEE/ACIS 24th Int. Conf. Comput. Inf. Sci. ICIS 2024 - Proc.*, pp. 113–120, 2024, doi: 10.1109/ICIS61260.2024.10778376.

[12] F. Dehghani, N. Movahhedinia, M. R. Khayyambashi, and S. Kianian, "Real-time traffic classification based on statistical and payload content features," *Proc. - 2010 2nd Int. Work. Intell. Syst. Appl. ISA 2010*, 2010, doi: 10.1109/IWISA.2010.5473467.

[13] P. Velan, M. Čermák, P. Čeleda, and M. Drašar, "A survey of methods for encrypted traffic classification and analysis," *Int. J. Netw. Manag.*, vol. 25, no. 5, pp. 355–374, Sep. 2015, doi: 10.1002/nem.1901.

[14] Jia Xing Qu, Guo Yin Zhang, "A Parallel Method of Deep Packet Inspection based on Message-Passing Interface," *Int. J. Secur. its Appl.*, vol. 9, no. 12, 2025, [Online]. Available: https://www.semanticscholar.org/paper/A-Parallel-Method-of-Deep-Packet-Inspection-based-Qu-Zhang/2f07ab83d1cc345bfb8a869847dfc57aa33282c1

[15] Kevin P. Dyer, Scott E. Coull, "Protocol misidentification made easy with format-transforming encryption," *Proc. ACM Conf. Comput. Commun. Secur.*, pp. 61–72, 2013, [Online]. Available: https://dl.acm.org/doi/10.1145/2508859.2516657

[16] S. Z. Weishi Sun, Yaning Zhang, Jie Li, Chenxing Sun, "A Deep Learning-Based Encrypted VPN Traffic Classification Method Using Packet Block Image," *Electronics*, vol. 12, no. 1, p. 115, 2023, doi: https://doi.org/10.3390/electronics12010115.

[17] M. Shen *et al.*, "Machine Learning-Powered Encrypted Network Traffic Analysis: A Comprehensive Survey," *IEEE Commun. Surv. Tutorials*, vol. 25, no. 1, pp. 791–824, 2023, doi: 10.1109/COMST.2022.3208196.

[18] Y. S. Razooqi and A. Pekar, "Vpn traffic analysis: A survey on detection and application identification," *IEEE Access*, vol. 13, pp. 132830–132848, 2025, doi: 10.1109/ACCESS.2025.3592152.

[19] M. Dusi, M. Crotti, F. Gringoli, and L. Salgarelli, "Detection of encrypted tunnels across network boundaries," *IEEE Int. Conf. Commun.*, pp. 1738–1744, 2008, doi: 10.1109/ICC.2008.334.

[20] Mohammad Lotfollahi, Ramin Shirali Hossein Zade, Mahdi Jafari Siavoshani, Mohammdsadegh Saberian, "Deep Packet: A Novel Approach For Encrypted Traffic Classification Using Deep Learning," *arXiv:1709.02656*, 2018, [Online]. Available: https://arxiv.org/abs/1709.02656

[21] M. Shen, M. Wei, L. Zhu, and M. Wang, "Classification of Encrypted Traffic with Second-Order Markov Chains and Application Attribute Bigrams," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 8, pp. 1830–1843, Aug. 2017, doi: 10.1109/TIFS.2017.2692682.

[22] Afeez Ajani Afuwape, Ying Xu, "Performance evaluation of secured network traffic classification using a machine learning approach," *Comput. Stand. Interfaces*, vol. 78, p. 103545, 2021, doi: https://doi.org/10.1016/j.csi.2021.103545.

[23] Dimitrios Effrosynidis, Avi Arampatzis, "An evaluation of feature selection methods for environmental data," *Ecol. Inform.*, vol. 61, p. 101224, 2021, doi: https://doi.org/10.1016/j.ecoinf.2021.101224.

[24] Zhonghang Sui, Hui Shu, "A comprehensive review of tunnel detection on multilayer protocols: From traditional to machine learning approaches," *Appl. Sci.*, vol. 13, no. 3, p. 1974, 2023, doi: https://doi.org/10.3390/app13031974.

[25] Eva Papadogiannaki, Sotiris Ioannidis, "A survey on encrypted network traffic analysis applications, techniques, and countermeasures," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–35, 2021, [Online]. Available: https://dl.acm.org/doi/10.1145/3457904

[26] Amin Shahraki, Mahmoud Abbasi, "Active Learning for Network Traffic Classification: A Technical Study," *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 1, pp. 422–439, 2022, doi: 10.1109/TCCN.2021.3119062.

[27] M. A. Sulaiman and J. Labadin, "Feature selection based on mutual information for machine learning prediction of petroleum reservoir properties," *2015 9th Int. Conf. IT Asia Transform. Big Data into Knowledge, CITA 2015 - Proc.*, Dec. 2015, doi: 10.1109/CITA.2015.7349827.

[28] Payap Sirinam, Marc Juarez, "Deep Fingerprinting: Undermining Website Fingerprinting Defenses with Deep Learning," *Proc. ACM Conf. Comput. Commun. Secur.*, 2018, [Online]. Available: https://dl.acm.org/doi/10.1145/3243734.3243768

[29] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges," *IEEE Trans. Netw. Serv. Manag.*, vol. 16, no. 2, pp. 445–458, Jun. 2019, doi: 10.1109/TNSM.2019.2899085.