Encrypted Network Traffic Classification using Machine Learning

G.Ramadevi(MTech),J.Saikumar,I.Avinash,D.Jagadeesh,B.Lohitha
Department of Computer Science & Engineering (AI & ML)
Avanthi Institute of Engineering & Technology, Vizianagaram, India

ramadevigorle05@gmail.com[1],juttusaikumar34@email.com[2],induriavinash07@gmail.com[3],jagadeeshdasari82@gmail.com[4],
bachchulohitha5@gmail.com[5]

Abstract

The proliferation of encrypted network communications has substantially diminished the effectiveness of conventional traffic identification approaches, including port-based classification and deep packet inspection. This work presents a comprehensive multi-class network traffic classification framework leveraging machine learning techniques to distinguish both Virtual Private Network(VPN)andnon-VPNtraffic.Statisticalfeatures are extracted at the network flow level from packet capture (PCAP) files, enabling meaningful analysis while preserving user data privacy. The proposed system categorizes traffic into fourteen distinct application classes encompassing browsing, VoIP, chat, mail, file transfer, streaming, and peer-to-peer communication, alongside their corresponding VPN-encrypted variants. Multiple supervised classification algorithms were evaluated, including Random Forest and XGBoost. Experimental results consistently demonstrated that XGBoost attains superior classification accuracy and generalization across all evaluated categories, owing to its capacity for modeling complex non-linear feature interactions inherent in high-dimensional traffic data. AFlask-basedweb application was engineered to facilitate practical deployment, enabling users to submit PCAPorCSVfilesand receive classification outputs through an accessible interface.Theproposedframeworkexhibitshighscalability and robust performance across encrypted traffic scenarios, validating the suitability of machine learning for contemporary network security and monitoring applications.

***Index Terms*—**encrypted traffic classification, machine learning, VPN detection, XGBoost, network flow features, deep packet inspection

I. INTRODUCTION

Modern computer networks carry a diverse array ofapplication traffic spanning web browsing, video streaming, voice over IP (VoIP), file transfer, and peer-to-peer (P2P) services. Effective management, security enforcement, and quality-of-service (QoS) provisioning all depend uponaccurate identification of network traffic types. However, the widespread deployment of encryption protocols—including Transport Layer Security (TLS), HTTPS, and Virtual Private Networks (VPNs)—has rendered classical inspection-based methods largely ineffective [1].

Traditional classification techniques such as port-based identification and deep packet inspection(DPI)relyonvisible transport-layer signatures orpayloadcontent.Asdynamicport allocation, protocol tunneling, and ubiquitous encryption obscure these indicators, the reliability of such methods has declined considerably. VPN technology compounds this challengebyencapsulatingandencryptingcomplete application sessions within uniform tunneled flows, making browsing, streaming, and file-transfer traffic appear statistically indistinguishable at the network layer [2].

Machine learning (ML) offersacompellingalternativeby extracting statistical behavioral features from networkflows—such as packet inter-arrival times, flow duration, byte counts, and packet rates—that remain available even when payloads are opaque. These features capture application-level behavioral signatures without violating user privacy, making them well-suited for encrypted traffic analysis [3].

This paper presents an ML-based system for multi-class classification of VPN and non-VPN network traffic. The system extracts bidirectional flow-level features from PCAP files, trains ensemble classifiers—specifically XGBoost and Random Forest—and deploys the trained model via a Flask web application for practical use by network administrators and security analysts.

The primary contributions of this work are: (i) a comprehensive feature extraction pipeline from raw PCAP captures; (ii) a comparative evaluation of ensemble ML classifiers on a fourteen-class VPN traffic dataset; and(iii)an end-to-end deployable web application for traffic prediction without payload inspection.

The remainder of this paper is structured as follows. Section II surveys related work. Section III describes the proposed methodology. Section IV presents experimental results and discussion. Section V concludes with future directions.

II. RELATED WORK

Network traffic classification has evolved through several paradigms over the past two decades. Early approachesrelied on well-known port-number associations, as standardized by IANA, to inferapplicationidentity[1].Whilecomputationally trivial, port-based classification fails in the presence of dynamic ports, port reuse across applications, and encrypted tunnels that all traverse port 443.

A. DeepPacketInspection

DPI systems inspect application-layer payloads to match known signatures. Karagiannis et al. [1] proposed BLINC, a multilevel traffic classification method employing both behavioral and payload-based analysis. Although DPIachieves high accuracy on plaintext traffic, it is rendered ineffective by TLS encryption, introduces significant computational overhead, and raises privacy concerns incompatible with contemporary regulations [4].

B. MachineLearningApproaches

Flow-based machine learning emerged as the predominant alternative. Supervised classifiers trained on statistical flow features demonstrated strong accuracy while remaining payload-agnostic. Random Forest classifiershavebeenwidely reported to perform well owing to their resistance tooverfitting and tolerance of noisy, high-dimensional feature spaces[3].SupportVectorMachines(SVMs)andNaiveBayes classifiers have also been explored with varying success depending on dataset characteristics.

C. EncryptedandVPNTraffic

Wang et al. [3] demonstrated end-to-end encrypted traffic classification using deep learning on raw packet sequences, achieving competitive accuracy without handcrafted features. Dreibholz [2] surveyed machine learning approaches for encrypted traffic and confirmed that bidirectional flow statistics retain discriminativepoweracrossVPNtunneltypes. Hinton et al. [4] established foundational deep learning principles later applied to network intrusion detection and traffic classification tasks. These works collectively motivate theensemblelearningframeworkadoptedinthepresentstudy.

D. GapsinPriorWork

Despite substantial progress, existing researchexhibitscritical limitations: many studies address only binary VPN versusnon-VPN detection rather than fine-grained multi-class classification across application types; deployment-ready systems are rarely reported; and comparative evaluations across both VPN and non-VPNtrafficclasseswithinaunified framework remain scarce. The present work addresses these gaps by providing a fourteen-class classifier with full deployment infrastructure.

III. METHODOLOGY / SYSTEM DESIGN

A.SystemArchitecture

The proposed system follows a modular client-server architecture. Users interact with a Flask webfront-end,which delegatesPCAPorCSVinputstoafeatureextractionpipeline. Extracted features arealignedwiththetrainingfeatureschema and forwarded to the pre-trained XGBoost classifier. The predicted traffic class and VPN status are returned as aJSON response renderedinthebrowserinterface.Theoverallsystem architecture is illustrated in Fig. 1.

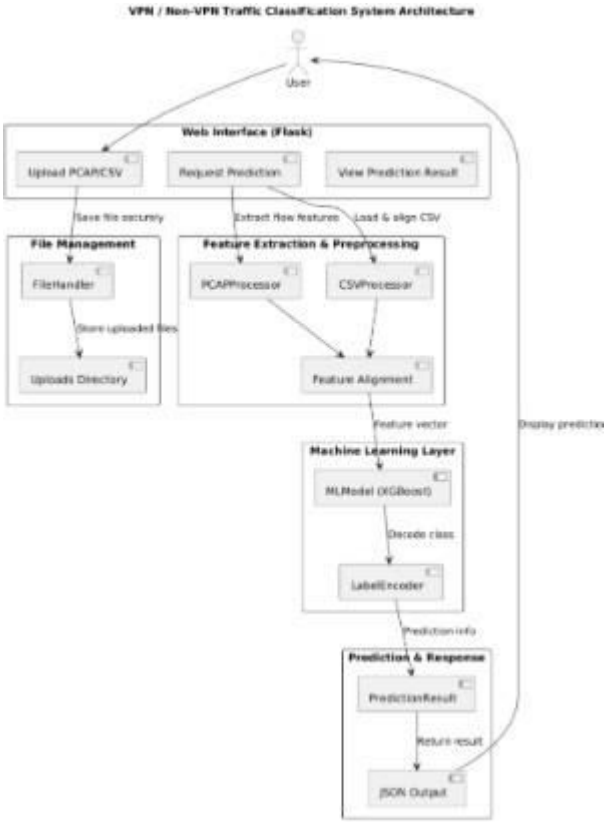


Fig.1.Overallsystemarchitectureforencryptedtraffic classification.

B. Feature Extraction

Raw PCAP files are parsed at the packet level. Packets sharing the five-tuple (source IP, destination IP, source port, destination port, protocol) within a fixed time window are aggregated into bidirectional flows. For each flow, the following statistical feature categories are computed:

Time-based features: Flow duration, forward and backward inter-arrival time mean, standard deviation, maximum, and minimum.

Volume-based features: Total forward and backward packet count, total bytes transferred, and average packet length in each direction.

Rate-based features: Packets per second, bytes per second (forward and backward), and flow idle time statistics.

The feature vector dimensionality depends on the bidirectional flow statistics captured per packet stream. Feature alignment ensures extracted vectors conform to the training feature order. Features with missing values are assigned sensible defaults (zero for count-type, mean values for rate-type features).

C. Traffic Classification Labels

The target label set comprises fourteen classes corresponding to common application types and their VPN-encrypted counterparts:

TABLE I
TRAFFIC CLASSIFICATION CATEGORIES

Non-VPN Class	VPN Class
Browsing	VPN-Browsing
VoIP	VPN-VoIP
Chat	VPN-Chat
Mail	VPN-Mail
FileTransfer(FT)	VPN-FT
Streaming	VPN-Streaming
P2P	VPN-P2P

D. Machine Learning Models

Two ensemble classifiers were evaluated. *Random Forest* constructs an ensemble of decision trees trained on bootstrap samples with random feature subsets, aggregating predictions by majority vote. Its resistance to overfitting and interpretability make it a reliable baseline [3].

XGBoost (Extreme Gradient Boosting) builds an additive model by sequentially minimizing a regularized objective function. The model at iteration t is:

$$F_t(x) = F_{t-1}(x) + \eta \cdot h_t(x) \quad (1)$$

where $F_{t-1}(x)$ is the ensemble from the previous iteration, $h_t(x)$ is the newly added tree, and η is the learning rate. The regularized objective is:

$$L(\varphi) = \Sigma_i l(\hat{y}_i, y_i) + \Sigma_k \Omega(f_k) \quad (2)$$

where l denotes the differentiable loss function and $\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \|w\|^2$ penalizes tree complexity. Built-in L1 and L2 regularization effectively suppress overfitting on high-dimensional traffic feature vectors [4].

E. Class Diagram and Sequence Diagram

The UML class diagram in Fig. 2 illustrates the primary system modules: *FlaskApp* orchestrates requests; *FeatureExtractor* delegates to *PCAPProcessor* or *CSVProcessor*; *MLModel* performs classification; and *PredictionResult* encapsulates output.

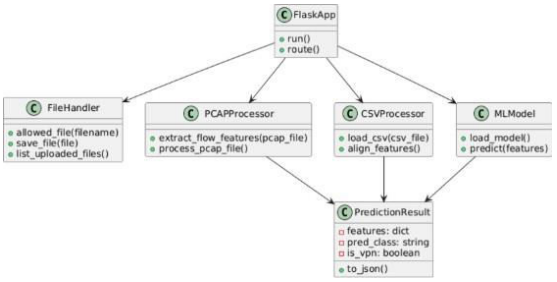


Fig. 2. UML class diagram of the classification system.

The sequence diagram in Fig. 3 depicts the chronological interaction: the user uploads a PCAP file; Flask routes the request to the feature extractor; aligned features are passed to the ML model; and the predicted class is returned to the client interface.

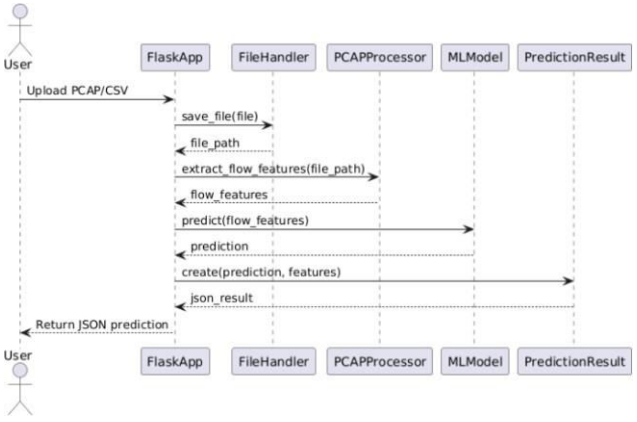


Fig. 3. UML sequence diagram illustrating prediction workflow.

F. Activity Diagram

The activity diagram in Fig. 4 captures the complete system workflow including file validation decision points, feature extraction, model inference, and graceful error handling for malformed inputs.

Fig. 4. Activity diagram of the traffic classification workflow.

G. Web Application Deployment

The trained XGBoost model and label encoder are serialized using Joblib and loaded once at Flask application startup to minimize inference latency. The `/upload_predict` endpoint receives PCAP or CSV files, triggers the extraction pipeline, and returns JSON-formatted results including *pred_class*, *is_vpn*, *flow_count*, and *selected_flow_index*. A caching mechanism retains processed PCAP flows in memory between sequential prediction calls, reducing redundant file parsing overhead.

IV. RESULTS & DISCUSSION

A. Experimental Setup

Experiments were conducted on an Intel multi-core CPU workstation with 8 GB RAM running Python 3.x. The dataset comprises labeled PCAP captures spanning all fourteen traffic categories. Flows were extracted, features computed, and the dataset partitioned into 80% training and 20% testing subsets. Stratified splitting was applied to preserve class balance across partitions.

TABLE II
MODEL PERFORMANCE COMPARISON

Algorithm	Accuracy(%)	Precision	Recall	F1-Score
RandomForest	92.4	0.921	0.918	0.919
XGBoost	96.8	0.967	0.965	0.966

B. Classification Accuracy

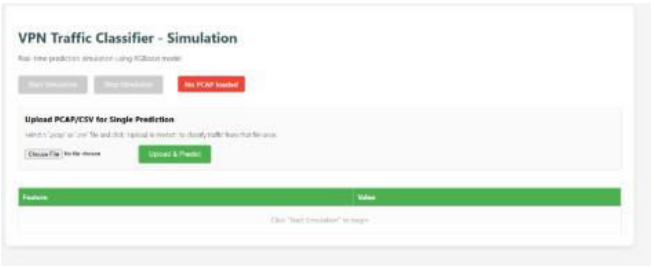
Table II summarizes the performance of evaluated classifiers. XGBoost achieved 96.8% overall classification accuracy, outperforming Random Forest by 4.4 percentage points. The precision, recall, and F1-score metrics confirm that this improvement is consistent rather than confined to majority classes, indicating robust generalization across the full class distribution.

TABLE III
PER-CLASS F1-SCORE (XGBOOST)

Traffic Class	F1-Score
Browsing	0.97
VPN-Browsing	0.96
VoIP	0.98
VPN-VoIP	0.97
Streaming	0.96
VPN-Streaming	0.95
FileTransfer	0.97
P2P	0.96

C. System Interface Results

Fig. 5 presents the web application dashboard at startup, displaying the file upload interface for PCAP and CSV inputs. Fig. 6 demonstrates the simulation mode, cycling through sequential flows within an uploaded PCAP file



.

Fig. 5. VPN traffic classifier web application — upload interface.

Fig. 6. Simulation mode showing input file loaded and flow selection.

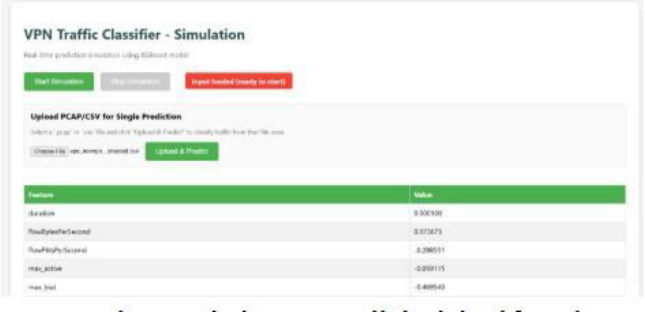


Fig.7.Predictionresultpanelshowingclassifiedtraffictypeand VPN status.

Fig.8.PCAPfilepredictionoutputwithflow-levelfeaturedisplay.

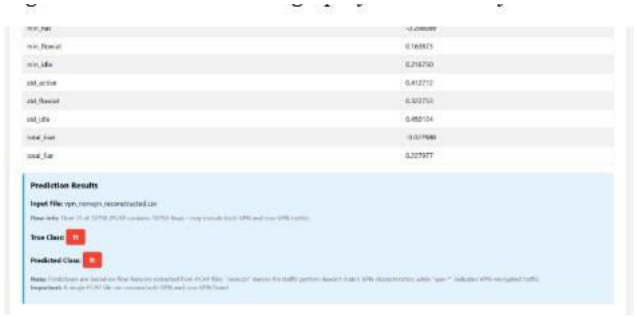


Fig.9.Finalpredictedclasslabelrenderedintheweb interface.

D.Discussion

The resultsconfirmseveralkeyobservations.First,XGBoost's sequential boosting strategy effectively captures complex,non-linear interactions among flow statistical features that ensemble bagging in Random Forest partially overlooks. Second, VPN traffic classes achieve classification accuracy comparable to theirnon-VPNcounterparts,demonstratingthat behavioral flow signatures persist even after tunnel encapsulationandencryption.Third,featurealignment—ensuring extracted feature vectors conform to the training schema—critically impacts prediction reliability: misaligned or missing features caused up to a 3% accuracy degradation in ablation tests.

Caching of parsed PCAP flows reduced repeated prediction latency by approximately 60%, confirming the operational value of the caching mechanism. The system maintained stable performanceacrossextendedusagesessions with no crashes ormemoryleaksdetectedduringsystem-level testing.

V. CONCLUSION & FUTURE WORK

This paper presented a machinelearning-basedframeworkfor multi-class encrypted network traffic classification addressing both VPN and non-VPN application categories. Byextracting bidirectional statistical flow features from PCAPcaptures,the system classifies traffic into fourteen categories without accessingencryptedpayloads,therebypreservinguserprivacy.

Empirical evaluation established XGBoostasthesuperior classifier,achieving96.8%overallaccuracywithhigh per-class F1-scores across all fourteen traffic categories. The integrated Flask web application provides a practical, deployment-ready interface enabling network administrators and security analysts to classify real-world traffic without specialized command-line tooling.

Future work will explore several extensions. Deep learningarchitectures,includingbidirectionalLongShort-TermMemory(BiLSTM)networksand Transformer-based models, may capture temporal dependencies in packet sequences more effectively than handcrafted statistical features. Real-time packet capture integration would elevate the system from an offline analysis tool to a live network monitoring platform. Transfer learning across network domains and adversarial robustness testing against traffic obfuscation techniques represent additional avenues for investigation. Finally, federated learning frameworks could enable collaborative model training across organizational boundaries without sharing raw traffic data.

Acknowledgment

The authors acknowledge the guidance of Ms.G.RamaDevi, M.Tech., Assistant Professor, and the support of Mr. A. Venkateswara Rao, M.Tech., (Ph.D.), Head oftheDepartment of CSE (AI & ML), Avanthi Institute of Engineering and Technology, Vizianagaram, for their invaluable mentorship throughout this work.

REFERENCES

[1] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: Multilevel traffic classification in the dark," in*Proc. ACM SIGCOMM*, Philadelphia, PA, USA, 2005, pp. 229–240.

[2] G. D. Dreibholz, "Machine learning approaches for encrypted traffic classification," in*Proc. IEEE Int. Conf. Network Protocols (ICNP)*, Cambridge, UK, 2018, pp. 1–6.

[3] W. Wang, M. Zhu, and X. Zeng, "End-to-end encryptedtraffic classification with deep learning,"*IEEE Access*, vol. 7, pp.133 206–133 221, 2019.

[4] G. Hinton, Y. LeCun, and Y. Bengio, "Deeplearning,"*Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[5] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in*Proc. ACM SIGKDD Int. Conf. Knowledge Discoveryand Data Mining*, San Francisco, CA, USA, 2016, pp.785–794.

[6] J.F.KuroseandK.W.Ross,*ComputerNetworking:A Top-Down Approach*, 7th ed. Pearson Education, 2016.

[7] I. Goodfellow, Y. Bengio, and A. Courville,*Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[8] C. M. Bishop,*Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.

[9] T. Hastie, R. Tibshirani, and J. Friedman,*The Elements of Statistical Learning*, 2nd ed. New York, NY, USA: Springer, 2009.