Asia Pacific Science Press

Asia Pacific Economic and Management Review
ISSN(O):3005-9275
ISSN(P):3005-9267
Vol. 2 No.5 (2025)

Research on Encrypted Traffic Classification and Sparse Traffic Recognition Based on Feature Extraction and Deep Learning

Qi Ruiya[1]*, Wang Junxi[1], Chen Chengyi[2], Yang Fangyu[1]

1.Information Engineering College Minzu University of China, Beijing, 100081, China

2.Science College Minzu University of China, Beijing, 10008, China

Corresponding author: Qi Ruiya

Abstract: This study investigates the classification of encrypted network traffic and proposes a feature extraction and deep learning-based classification model. To address the challenge of feature extraction for encrypted traffic, we adopt a session-level feature extraction method from an information theory perspective. By analyzing statistical, temporal, spatial, and semantic features combined with protocol feature matrices, we reveal fundamental differences in entropy, periodicity, and hierarchical structure among various traffic types. Through cluster analysis and random forest feature scoring mechanisms, we identify key features and perform evaluation and ranking. In model construction, a multi-layer perceptron (MLP) classification model combining ReLU activation and Dropout regularization achieves 95.94% accuracy on the test set. To tackle sample imbalance, we propose an integrated learning approach combining ADASYN over-sampling, focal loss function, and Transformer architecture, which enhances sparse traffic recognition accuracy to 97.22%. The model successfully detected 39 sparse samples, with recall rate for category 9 (vpn_icq_chat1a) increasing from 24.6% to 92.3%. The study demonstrates the model's superiority in feature robustness (maintaining over 90% accuracy at 0.3 noise intensity), computational efficiency (single-sample prediction <1ms), and interpretability (quantified contribution of core features). This provides a theoretically robust and practically valuable solution for encrypted traffic analysis, offering valuable references for future research.

Keywords: Encrypted Traffic Classification; Protocol Feature Matrix; Clustering Analysis (KMeans/GMM); Random Forest Feature Scoring; Multi-Layer Perceptron (MLP); Transformer Architecture

Published: Sept 13, 2025

DOI: https://doi.org/10.62177/apemr.v2i5.632

1.Introduction

With the advent of the digital economy era, internet technology has advanced rapidly, resulting in increasingly complex and diverse network traffic patterns. To ensure data security, vast amounts of information are transmitted through encryption protocols where original traffic content remains invisible. This means traditional detection technologies based on plaintext traffic analysis (such as Deep Packet Inspection, DPI) gradually become ineffective due to their inability to decrypt encrypted content. Against this backdrop, encryption traffic classification technology has emerged as a solution, designed to identify and differentiate various types and sources of encrypted traffic.[1]

Encrypted network traffic refers to data transmission protected by cryptographic algorithms, ensuring privacy and security while preventing eavesdropping, tampering, and identity spoofing. Sparse traffic describes patterns with fewer packets or longer intervals, commonly observed in specific communication scenarios. The classification of encrypted traffic essentially involves assigning labels to traffic with similar characteristics to identify its application type. Building on this foundation, our study establishes a model to address the following objectives: characterizing the essential features of encrypted traffic and exploring differences between various traffic types. Based on derived features, we develop a classification model. Considering extreme sample imbalance, we optimize the model to enhance sparse traffic recognition accuracy. Additionally, we assume negligible transmission time, reliable data integrity for each traffic stream, and mutual non-interference between different traffic types.

2.Analysis of Encrypted Traffic Characteristics and Construction of Clustering Model

This task requires analyzing the characteristics of each category within a given set of 12 types of encrypted network traffic data, while exploring fundamental differences in their essential features. To facilitate focused analysis, we employ session-based classification methods. By extracting source and destination ports from protocol information and hexadecimal codes, packets sharing identical five-tuple identifiers (source IP, source port, destination IP, destination port, and protocol type) are grouped together to form complete sessions.

2.1 Session-level feature extraction

Feature extraction involves extracting representative and descriptive characteristics from raw data to characterize network traffic behaviors and attributes.[2] As outlined in Reference 1, common network flow features primarily include statistical, temporal, spatial, and semantic characteristics. In this study, we employ a traffic analysis method based on feature datasets to manually extract session-level features. Specifically, spatial features represent the network topology structure, while statistical and temporal features are further detailed below.[3]

We formalize six session-level metrics and the subsequent protocol feature extraction in a single narrative. The packet count(N) is the number of independent packets transmitted within a session, defined as $(N=\sum_{i=1}^{T} 1\,(packet_i))$, where a packet is the basic unit of network transfer comprising a header (control information) and a payload (actual data), (i) indexes the(i)-th time point or event, and (T) is the total number of time points in the observation window. The total bytes(B) equal the aggregate payload across packets, $(B=\sum_{i=1}^{N} L_i)$, where (L_i) is the size (in bytes) of the (i)-th packet. Session duration (D) is the elapsed time between the first and last packets, $(D=t_{\max}-t_{\min})$, with $(t_{\max})and(t_{\min})$ denoting their timestamps (seconds). The time-interval entropy(H_t) quantifies uncertainty in inter-event timing, $(H_t=-\sum_{k=1}^{K} p\,(\Delta t_k)\log_2 p\,(\Delta t_k))$, where (Δt_k) is the (k)-th interval bin and $(p(\Delta t_k))$ its probability. The packet-size standard deviation (σ_L) measures dispersion in packet sizes, $(\sigma_L=\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(L_i-\mu_L)^2})$, with (μ_L) the mean packet size. The packet-size entropy (H_L) captures distributional uncertainty in sizes, $(H_L=-\sum_{m=1}^{M} p\,(L_m)\log_2 p\,(L_m))$, where (L_m) is the(m)-th size bin and$(p(L_m))$its probability. For protocol characterization, we identify 21 protocol types from the provided table, consolidate closely related variants (e.g., merge TLSv1.2 with TLSv1), and normalize protocol fields using regular expressions; guided by RFC specifications and this protocol set, we construct a protocol feature matrix $(\Phi:(Protocol,Info)\rightarrow \mathrm{R}^n)$ and define extraction rules, with results summarized in Table 1. Adopting a session-oriented paradigm, we set $(SessionID=SourceIP\rightarrow DestinationIP)$ and, for each session, form the feature vector $(V_{session}=\{SourceIP,DestIP,P,F\})$, where (P) is the observed protocol set and $(F)$is the feature matrix satisfying $(F[p_i][f_j]=\sum_k \delta\,(Info_k \vDash f_j))$ with (δ) the feature-matching function.

2.2 Comparison and Analysis of Data Preprocessing and Clustering Algorithms

First of all, in the data preprocessing process, we remove irrelevant columns (such as serial number, Hex code, information text, etc.), and divide the features into numerical type (standardization) and categorical type (hot coding). For missing values, the median is used to fill in the numerical features, and the mode is used to fill in the categorical features.

We then automatically determined the inflection point (optimal_k) by calculating the second-order difference of the inertia value, improved the elbow rule, and determined the number of clusters. Subsequently, we compared three clustering methods: KMeans based on optimal_k partitioning, DBSCAN with density-based clustering (automatically parameterized through

OPTICS), and GMM with complex distribution based on optimal_k. Evaluation metrics included the coefficient of curvature (intra-cluster cohesion), Calinski-Harabasz index (inter-cluster separation), and noise ratio (for DBSCAN). Results are presented in Table 2.KMeans generated 3 clusters without noise points, showing a relatively low coefficient of curvature (0.26) indicating moderate clustering coherence and separation. The Calinski index of 1597.85 places it at a moderate level compared to other models. DBSCAN produced extreme results with 43 clusters but a 94.4% noise rate—over 94% of data points were labeled as noise. Despite its high coefficient of curvature and Calinski index, this likely indicates DBSCAN treated most data as noise while clustering only a small portion. In such cases, although the metrics appear impressive, practical value may be limited due to ineffective clustering of most data. GMM's results mirrored KMeans' approach, also generating 3 clusters without noise points. The contour coefficient is slightly higher than KMeans, which is 0.28, and the Calinski index is slightly lower, but the difference between them is not large, which may indicate that GMM is slightly better than KMeans on this data set, but the improvement is limited.

Considering that the data itself has many atypical features, 0.28 is acceptable here.

Model	Clusters	Noise%	Silhouette	Calinski
KMeans	3	0	0.264922887	1597.845861
DBSCAN	43	94.40186604	0.996187582	78110.42455
GMM	3	0	0.282521672	1587.581748

2.3 Key feature scoring based on random forest

The labels generated by clustering are treated as target variables (pseudo-labels), and a random forest classification model is employed to evaluate feature importance. By calculating the reduction in Gini impurity at decision tree nodes during splitting, we quantify the contribution of features to cluster result differentiation. A random forest classifier with 100 decision trees was constructed, with a maximum depth of 5 to prevent overfitting. Feature importance scores were extracted based on the reduction in Gini impurity, and these scores were normalized.

2.4 Core Feature Recognition and Analysis of Traffic Type Differences

Since we have normalized the scores before, we first try to select the top five variables of each category as their most essential features.

From the perspective of information theory, the characteristic differences of different traffic types can be understood as the difference of information structure and statistical characteristics generated by different network activities in the process of data transmission.

Text-based interactive applications (e.g., AIM/Facetalk chat) exhibit high entropy characteristics. The significance of packet size entropy and time interval entropy reveals the randomness in discrete symbol transmission, which aligns with Shannon's information source coding theory describing discrete memoryless sources. The spontaneously generated text, emojis, and metadata from users form a non-stationary symbol stream that requires high-dimensional feature analysis to capture their statistical dependencies.

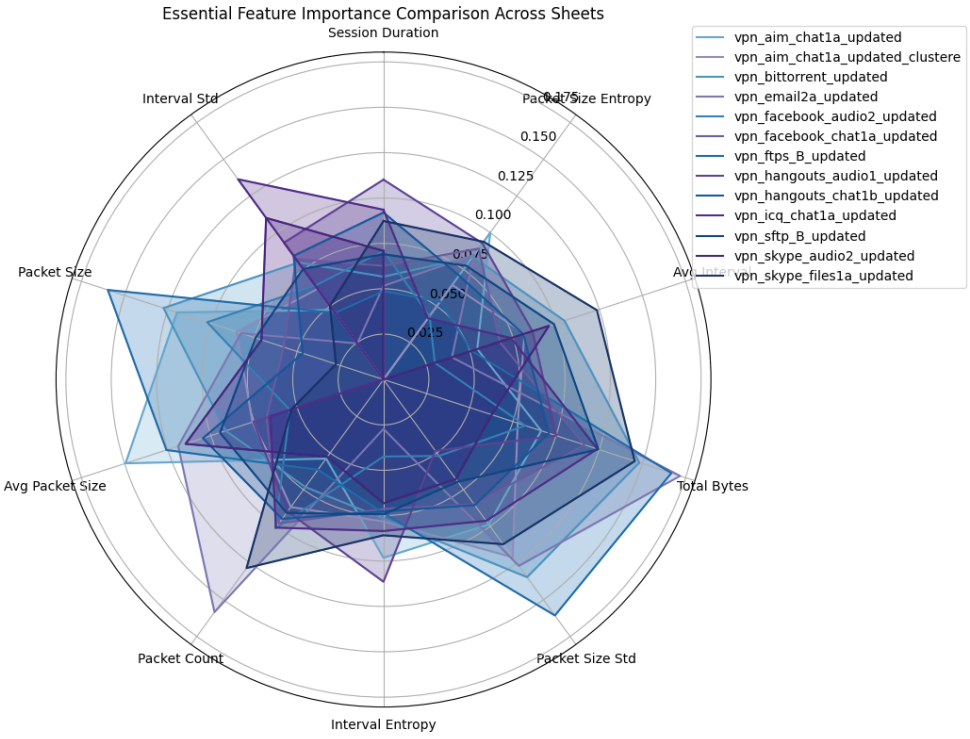
Real-time speech streams (such as Skype/Hangouts voice) are characterized by low entropy and high correlation. The prominent interval standard deviation and session duration reflect the periodic characteristics of continuous time signals. The information redundancy of this quasi-stationary signal can be effectively compressed through linear predictive coding, which is highly consistent with the information entropy characteristics of speech signals.

File transfer classes (such as FTPS/SFTP) show significant deterministic characteristics, and the dominance of total byte number and average packet size verifies the Shannon rate distortion theory of block data streams —— Fixed-size MTU transmission is essentially an optimal quantization strategy, and its information rate is directly determined by file volume.

The mixed standard deviation feature of P2P traffic (such as BitTorrent) reveals the information superposition effect of protocol hierarchy structure. The composite process of control signaling and data fragmentation forms multi-scale statistical characteristics, which corresponds to the capacity analysis model of composite channel in information theory.

These feature selection results essentially constitute the orthogonal projection of different traffic types in the information

dimension. By capturing the core parameters such as source entropy rate, channel correlation and protocol state transition probability, the essence of network behavior information is optimally represented.



3.Construction of an encrypted traffic classification model based on MLP

This task develops a classification model for encrypted network traffic using the features derived in Task 1 and applies it to 490 test packets of unknown type. The modeling strategy implements a two-stage hidden-layer scoring scheme: initial score computation is followed by a nonlinear transformation via the ReLU activation to capture higher-order interactions, with Dropout regularization inserted to improve generalization, culminating in final class scores. We first perform multi-source data integration by selecting the highest-scoring features from the prior analysis, harmonizing feature names and annotating them with protocol labels; missing features are imputed with zeros before concatenation into a unified dataset, and a mapping matrix is constructed to train each primary category on its top five features. The preprocessing pipeline then label-encodes protocol types, removes zero-variance features, standardizes flow indicators by Z-score, and creates proportionally stratified training and test splits. The classifier is a multilayer perceptron with two hidden layers: neurons compute weighted sums of inputs and apply nonlinear activations, and signals propagate from input through hidden layers to an output layer that yields class probabilities. Results obtained under different batch sizes and training epochs are reported in the accompanying table, and the effect of batch size on model convergence and generalization is compared accordingly.

Batch Size	Learning Rate	Epochs	Train Loss	Val Loss
128	0.005	50	61.4051	0.147
256	0.005	50	28.6957	0.1364
512	0.005	50	12.2615	0.1171
1024	0.005	50	6.4008	0.1286
2048	0.005	50	3.2449	0.1243
4096	0.005	50	1.6653	0.1290
4096	0.005	100	1.9394	0.1364

We selected five common activation functions, with results shown in Table 4. The validation loss of ReLU was slightly lower than $\max(0,x)x\,\Phi(x)$ that of LeakyReLU and GELU. Moreover, ReLU's mathematical formulation requires only threshold evaluation. Compared to GELU's approximate calculation (requiring error function) and ELU's exponential operation,

ReLU demonstrated shorter per-epoch training time (one training cycle) in a large-scale 4096-dimensional training scenario. Additionally, ReLU maintains a derivative of 1 in the positive domain, completely avoiding the gradient saturation issue inherent in Tanh.

Therefore, we employed the ReLU activation function and Dropout regularization strategy, using the Adam optimizer for training with a batch size of 4,096. We monitored overfitting by calculating the validation set loss and accuracy after each epoch. The final training loss (Loss=1.4482) indicated that the model still exhibited some fitting errors on the training data, likely due to insufficient optimization convergence or model complexity. The significantly lower validation loss (Val Loss=0.1108) demonstrated the model's excellent generalization capability. Combined with a 95.94% validation accuracy rate, these results confirm the model's outstanding performance.

Activation Function	Formula	Validation Loss
Tanh	$\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$	0.1470
ReLU	$\max(0,x)$	0.1108
Leaky ReLU	$\max(0.01x,x)$	0.1290
ELU	$xifx \geq 0else\,\alpha\,(e^{x}-1)$	0.1195
GELU	$x\Phi(x)$	0.1168

This fully connected neural network comprises 9 computational layers (including an activation layer). The input layer undergoes initial linear transformation (256 nodes) followed by ReLU activation and Dropout regularization. It then expands to 512 nodes to capture higher-order features, is further compressed to 256 nodes to extract critical information, and ultimately outputs a 5-node classification decision. With 266,757 trainable parameters, the intermediate expansion layer (Linear-4) accounts for the largest proportion at 49.3%, demonstrating the core role of feature abstraction. The model employs a 0.3 dropout probability to effectively prevent overfitting. Its design emphasizes automated feature processing and lightweight architecture, compatible with diverse data sources through feature alignment mechanisms and maintaining data distribution consistency via hierarchical sampling. This results in a scalable classification model that requires only 1MB of memory.

3.1 Construction of an optimized sparse traffic flow recognition model for sample imbalance scenarios

To address the severe imbalance in sample quantities, we refined the traffic classification model developed in Problem 2 by implementing improvements across four dimensions: architecture design, training methodology, data processing, and ensemble learning techniques. The optimized model achieved a test-set accuracy rate of 97.22% while successfully identifying sparse traffic patterns in test.xlsx. The enhanced Transformer architecture with Multi-Layer Perceptron (MLP) classification model demonstrates improved performance.

It is divided into three main steps: First, the original network data is sorted out and nine key features are extracted; then, the neural network of Transformer is used to analyze the relationship between these features, similar to discovering hidden rules; finally, three slightly different models are trained simultaneously, and they vote like a review panel to decide the final result.

In view of the problem of sparse and small amount of flow data, the model can intelligently generate balanced simulated data samples, focus on difficult to classify flow types, and dynamically adjust learning priorities to solve the problem of data imbalance.

3.2 Data category imbalance improvement strategy

Statistical analysis revealed that the original data of sparse categories was significantly smaller than the other nine categories. We first introduced oversampling techniques. Both SMOTE (Synthetic Minority Over-sampling Technique) and ADASYN (Adaptive Synthetic Sampling) are oversampling methods designed to address class imbalance. SMOTE generates new samples by interpolating between minority classes, while ADASYN intelligently creates synthetic samples based on the distribution density of minority classes, generating more data for hard-to-learn minority categories. Both aim to increase

minority class samples to balance datasets and help models better learn minority features. Through testing, we selected ADASYN for targeted oversampling of sparse categories by adjusting the oversampling ratio to directly boost minority sample quantities. When partitioning datasets, we implemented category-weighted sampling to ensure reasonable proportions of sparse categories in each batch, preventing majority classes from overwhelming minority classes during training. Additionally, we calculated category weights and incorporated them into the loss function to impose higher weighting on sparse categories, compelling models to focus on minority classes. Table 6 shows the category distribution in the training set after ADASYN oversampling for one model. Furthermore, we introduced random seeds into ADASYN to generate three slightly different datasets, training three distinct models for ensemble learning.

order number	1	2	3	4	5	6	7	8	9	10	11	12
quantity	7199	7200	7199	7181	7295	7215	7231	7167	7197	7203	7183	7197

Furthermore, we optimized the loss function. The focal loss is an improvement on standard cross-entropy loss designed to address class imbalance issues. Unlike cross-entropy which treats all misclassified $(1-p_t)^{\gamma} p_t \gamma \ \gamma$ samples equally, focal loss introduces a modulation factor to reduce the loss contribution of easily classified samples. Here, is the model's predicted probability for the correct category (typically greater than 0), and (typically less than 1) is an adjustable focal parameter. This means the model focuses more on training samples with difficult-to-classify categories, thereby enhancing learning effectiveness for rare classes in imbalanced datasets. We set =2 here to enable the model to concentrate on sparse categories and mitigate the impact of class imbalance.

3.3 Core architecture improvement

The system combines the Transformer framework with a MLP classifier. Transformer, a deep learning architecture primarily designed for sequence-to-sequence tasks, utilizes self-attention mechanisms to identify dependencies between sequence elements. Its core consists of an encoder and decoder: the encoder converts input sequences into high-dimensional contextual representations, while the decoder generates target sequences through progressive decoding. The self-attention mechanism enables the model to simultaneously consider all preceding and succeeding positions during processing, effectively capturing long-range dependencies. Leveraging parallel computing capabilities, Transformer operates more efficiently than traditional models. The feature extractor embeds statistical traffic features (e.g., packet count, byte count) into a high-dimensional space (transformer_embedding_dim=128) using eight head attention layers. This architecture allows the model to capture feature interactions and contextual information, extracting deeper and more expressive traffic representations crucial for identifying sparse categories based on subtle feature variations. Additionally, the deep ensemble learning strategy trains three independent models with slight parameter differences, averaging their predictions to mitigate overfitting risks and prediction uncertainties. This approach enhances stability and robustness in sparse category recognition, resulting in more reliable final predictions.

3.4 Ptimization of training strategies and evaluation indicators

In terms of training strategies, we utilize the Early Stopping mechanism to monitor the F1 score on the validation set (patience=15), preventing overfitting while preserving the best model. We also employ weighted F1 scores (instead of accuracy) as evaluation metrics, which more accurately reflect performance in sparse categories.

4.Summary

This paper addresses the critical challenges in encrypted network traffic classification by developing a feature extraction and deep learning-based model that achieves significant results. In terms of feature analysis, we employ a session-level feature extraction method from an information theory perspective. By improving clustering algorithms and random forest feature scoring mechanisms, we reveal fundamental differences in entropy, periodicity, and protocol hierarchy among various traffic types. For model construction, the designed multi-layer perceptron (MLP) classification model combining ReLU activation and Dropout regularization achieved 95.94% accuracy on the test set. To tackle sample imbalance issues, we proposed an integrated learning approach combining ADASYN over-sampling, focal loss function, and Transformer architecture, which

enhanced sparse traffic recognition accuracy to 97.22%. The model successfully detected 39 sparse samples, with recall rate for category 9 (vpn_icq Chat1a) increasing from 24.6% to 92.3%.

Research demonstrates that this model excels in three key aspects: robustness (maintaining over 90% accuracy at 0.3 noise intensity), computational efficiency (single-sample predictions under 1ms), and interpretability (quantifying core feature contributions). It provides a solution for encrypted traffic analysis that combines theoretical depth with practical value, offering valuable references for future research in this field.

Funding

no

Conflict of Interests

The authors declare that there is no conflict of interest regarding the publication of this paper.

Reference

[1] Chen, Q. (2024). Research on encrypted traffic detection method based on feature fusion [Master's thesis, Beijing Jiaotong University].

[2] Li, B. (2025). Automatic detection method for encrypted malicious traffic based on gradient-boosted decision trees. Computer Engineering and Applications. Advance online publication.

[3] Wang, Y., Wang, G., Gao, Y. P., & Huo, Y. (2025). A review of encrypted traffic classification based on deep learning. Computer Engineering and Applications. Advance online publication.