Policy Research Journal
ISSN (E): 3006-7030 ISSN (P) : 3006-7022 Volume 4, Issue 3, 2026

ADAPTIVE DEEP LEARNING FRAMEWORK FOR ENCRYPTED NETWORK TRAFFIC CLASSIFICATION IN THE CASE OF CONCEPT DRIFT AND LIMITED LABELED DATA.

Naila Anwar

PhD Scholar, Dept. of Information Technology, University of Gujrat,

naila.anwwar@uog.edu.pk

DOI: https://doi.org/10.5281/zenodo.24027644

Keywords
Adaptively learning, labeled data, encrypted traffic, architecture

Article History
Received: 07 January 2026
Accepted: 17 March 2026
Published: 31 March 2026

Copyright@Author
Corresponding Author: *
Naila Anwar

Abstract
Network traffic classification has become hard with the rise of the encryption technologies, making traditional inspection methods less effective. However, the machine learning and deep learning solutions are not always robust to changes in traffic patterns, called concept drift, and lack of labeled data. This research aims to present an adaptive deep learning framework with CNN-LSTM architecture, concept drift detection, and label-efficient learning mechanism for encrypted network traffic classification. The main goal of the proposed framework is to increase the classification accuracy and increase the adaptability of the classification with the minimum retraining frequency, minimum computation overhead and minimum processing delay in dynamic network.

IEER
Institute for Excellence in Education & Research

Problem statement:
The increasing use of encryption protocols such as TLS and QUIC have made network traffic classification increasingly challenging by restricting access to packet payloads. Machine learning or deep learning methods rely on statistical features of traffic, but they are not adaptable to changing traffic behaviors due to their reliance on fixed datasets. The classification accuracy and model reliability further decreases over the time due to concept drift. Pre-existing methods also entail reiterated training of labeled data, which leads to high computational burden and processing delay. Furthermore, it is hard to detect traffic patterns that haven't been seen before. A deep learning framework that adapts to the change of concepts, learns new encrypted traffic patterns with little labeled information and classification accuracy with low retraining needs and overhead, is needed.

INTRODUCTION:

Rapid growth of Internet-based services, cloud computing, mobile applications and digital communication have significantly added volume and complexity to network traffic. Network Traffic Classification has become a vital part of network management and cyber security as it allows for the identification of applications, services, and communication activities. It helps in performing important functions like network monitoring, Quality of Service management, bandwidth optimization, anomaly detection and intrusion detection. The three most common methods of network traffic classification are port-based identification, protocol analysis and Deep Packet Inspection (DPI). These techniques have become rather ineffective since the advent of wide use of the encryption protocols as packet payloads can no longer be inspected directly [1, 2].

For sensitive communication, protocols like Transport Layer Security (TLS), TLS 1.3 and Quick UDP Internet Connections offer

confidentiality and integrity. While encryption adds security to the network, it makes it difficult to identify applications and services using traditional inspection methods. As a result, researchers have started studying the classification of encrypted traffic using Machine Learning (ML) and Deep Learning (DL) techniques. Instead of relying on packet payload information [2] [3], these approaches use the statistical and behavioral properties, such as packet size, length of the flow, number of packets, packet inter-arrival times and communication patterns.

A number of machine learning techniques have been shown efficient in the task of network traffic classification, such as Random Forest, Support Vector Machine and Decision Trees. In the same way, deep learning networks like Convolutional Neural Networks, Recurrent Neural Networks and Long Short-Term Memory networks are able to automatically learn complex traffic representations and temporal dependencies. CNNs excel in capturing traffic patterns and meaningful ones, while LSTMs capture sequential relationships in network communication. The combination of these architectures provides opportunities for representation and classification of encrypted traffic of networks [3, 4].

The traffic landscape is constantly changing in the real world as a result of the introduction of new applications, protocols, user behavior, and changes to network configurations. These changes may affect the statistical relationship between the network features and the different traffic classes, which is known as concept drift [5]. Concept drift can be abrupt, gradual or incremental and have an impact on classification accuracy and the reliability of the model. Algorithms such as Adaptive Windowing, Drift Detection Method, and Page-Hinkley can detect changes in the data streams received and aid in proper model updates. Some of these, such as ADWIN, adapt their observation window based on the changes they observed [6].

One of the other great challenges is lack of labelled encrypted traffic. Most deep learning models depend on a good amount of labeled training data, and new network traffic is hard to obtain and label. Supervised learning is also complicated by the emergence of new applications and the presence of traffic categories that are not seen. As such, frequent retraining adds up to extra computational overhead, memory, and processing latency.

In response to these problems, a method is proposed to classify encrypted network traffic by using an adaptive deep learning system with limited labeled data under concept drift. The framework will include concept drift detection, label-efficient adaptive learning mechanisms and feature learning using CNN-LSTM-based approach. The proposed approach will be evaluated using the datasets like ISCX VPN-nonVPN 2016 and USTC-TFC2016 against the conventional approaches of ML and DL.

2. Literature Review:

Network traffic classification is a fundamental study field in network security and network management. Traditional methods of network application/service identification involve port numbers, protocol information and Deep Packet Inspection. Conventionally used classification techniques have been limited, however, by the growing use of encryption, which has limited access to the packets' payloads. Velan et al. [8] did a survey of techniques to classify encrypted traffic and emphasized the role of observable traffic properties like connection behavior and statistical patterns. In the same way, Sharma and Lashkari [9] studied the machine learning and deep learning techniques, datasets and technical problems related to encrypted traffic analysis.

Popular machine learning algorithms are used for network traffic classification such as Random Forest, Support Vector Machine (SVM), Decision Trees and Naive Bayes. These algorithms classify the traffic type based on statistical parameters such as a packet size, flow duration, packet frequency and inter-arrival time. Rezaei and Liu [10] studied encrypted traffic classification using deep learning methods and discussed their capability of automatically extracting complex traffic representations. They noted in their study that their approach could be used to lessen reliance on manually created features. In addition, Aceto et al. [11] suggested a multimodal deep learning solution called MIMETIC, for mobile encrypted traffic classification.

One of the biggest drawbacks of traditional machine learning and deep learning models is assumption that training and testing data are distributed the same way. Traffic patterns in practical network environments are ever-changing thanks to new applications, protocol changes, shifts in user behavior, and network configuration changes. These changes may involve the input feature and its corresponding classes, and may cause concept drift. Gama et al. [12] provided a complete survey of concept drift adaptation techniques and how changes in the data distribution can impact on performance of the predictive model. They studied various techniques in order to detect and adapt to changing data streams.

CNN and LSTM architectures have potential to offer some valuable features to be used for creating adaptive traffic classification systems. CNNs capture representative patterns from the structured traffic features, while LSTMs capture sequential relationship and temporal dependency. They can be used in conjunction with CNN-LSTM architecture to enhance the modeling of complex network traffic patterns. In most cases, traditional systems rely on static training sets. As new types of traffic appears, accuracy in classification may fall, if the model is not updated. More retraining leads to higher computational overhead and memory usage as well as higher processing latency.

Yet another challenge for encrypted network traffic classification is limited labeled training data. Most of the deep-learning models need adequate samples of labeled data to learn representative traffic patterns. However, gathering and marking of encrypted traffic can be time consuming, especially if novel communication patterns and applications are appearing on the scene. A survey on few-shot encrypted traffic classification: few-shot encrypted traffic classification was studied which includes data augmentation, transfer learning, metric learning, meta-learning and pertaining-based approaches [14].

Recent studies have focused on combining data efficient learning with adaptive classification. Rahaman et al. [8] surveyed few-shot, zero-shot, meta-learning and self-supervised methods for encrypted traffic classification for modern protocols like TLS 1.3 and QUIC. Their results pointed out the lack of empirical assessment for concept drift and the need for more extensive temporal experiments. Likewise, Enisoglu and Inan [16] mentioned some of the key challenges in modern network classification, like concept drift, label scarcity, open-world classification, and deployment efficiency.

Research questions:

- What are the challenges in designing an adaptive deep learning framework to detect and adapt the concept drift in encrypted network traffic classification when there is limited labeled data?
- How can this be said that proposed framework enhances accuracy and adaptability while lowering model retraining demand and processing latency, and computational overhead with respect to existing approaches?

Research objectives:

- To conduct and define concept drift in encrypted network traffic based on changing pattern of network traffic and the effect of changes on performance of current deep learning network traffic classification models.
- To formulate and test an Adaptive Deep Learning (ADL) with drift detection and label-efficient learning mechanisms to demonstrate its performance in comparison to existing methods in terms of classification, performance and adaptation capability, frequency of retraining and processing latency.

3. Research Methodology:

In this study, we propose and assess an adaptive deep learning network traffic classification framework under concept drift and limited labeled data through experimental and quantitative approach. The methodology consists of collection of datasets, data preprocessing, feature extraction, detection of drift, construction of framework, and experimental evaluation.

3.1. Dataset Collection:

Three publicly available data sets have been used namely ISCX, VPN-non, VPN 2016, USTC-TFC2016 and CIC-Darknet2020. The dataset

includes variety of network traffic types and application behaviors. The PCAP files are processed on the datasets to generate network flows and each flow is then correlated with the traffic category to assign ground truth.

During the preprocessing, incomplete records, duplicate records and irrelevant attributes are removed. No decrypting of payloads – statistical and behavioral features are extracted. These are the sizes of the packets, flow duration, number of packets, inter-arrival times of packets, forward and backward traffic statistics and byte transmission rates.

3.2. Concept Drift Analysis:

Concept drift is analyzed through monitoring changes of network traffic distributions over time. Traffic streams are grouped together into successive windows, which are used to model the evolution of network conditions. Sudden, gradual and incremental drift scenarios are addressed. ADWIN, DDM and Page-Hinkley test are assessed for their ability to detect changes in incoming traffic and observe their impact on classification.

3.3. Proposed Adaptive Deep Learning Framework:

The framework proposed is a combination of deep learning based traffic classification, concept drift detection and adaptive model update. CNN-LSTM is employed to extract the spatial and temporal features from encrypted traffic features. Drift detection component constantly checks incoming traffic. If drift is detected, the model is trained with the newly available labeled samples. To avoid catastrophic forgetting, and to prevent the need for complete model re-training, continual and incremental learning mechanisms are integrated.

3.4: Experimental Design:

A series of controlled experiments test the framework in static traffic, sudden concept drift, gradual concept drift, as well as under varying proportions of labeled data. The labeled-data scenarios include 1%, 5%, 10%, 20%, and 100%. Random Forest, Support Vector Machine is compared with the proposed model.

3.5. Performance Evaluation:

Accuracies, precision, recall and F1-score are used to measure performance of the classification. Adaptability is measured by adaptation time, post drift performance and recovery rate. Model updating time, CPU utilization, memory consumption, retraining frequency, and classification latency are used to measure the computational efficiency. Comparative experiments are performed under the same conditions and repeated experiments are performed to check the consistency of the results.

4. Analysis:

Following section presents analysis of results obtained.

4.1. Feature Extraction and Data Preprocessing:

Without decrypting the payload of the packets, flow-level statistical features are extracted. These features are indicative of traffic behavior, communication patterns and packet characteristics. Table 1 and 2 describe features extracted for encrypted network traffic and the samples of encrypted network traffic.

Table 1: features extracted for encrypted network traffic

ID	Primary key to be imported	integer
Flow end time	End time of network flow	Numerical
Tunnel number	Number of packets transmitted	numerical
Fully backward packets	Packets sent from destination to source	numerical
Backward packet length mean	Average size of packets	numerical
Backward packet length mean	Average size of packets	numerical
Flow packets	Average packets transferred	numerical

Table 2: Sample of extracted network traffic data

Flow ID(bytes)	Flow duration(bytes)	Forward flow packets	Flow packets	Backward packets	Forward mean length	Traffic label
F001 2.45	15	12	620	1250	11.02	E-mail
F002 3.21	22	18	780	1480	12.46	VPN-Email
F003 1.85	10	8	450	920	9.73	Chat
F004 4.32	28	21	850	1850	11.34	VPN-Chat
F005 5.10	35	29	920	2100	12.55	Streaming
F006 2.90	19	14	690	1350	11.38	VPN-streaming
F007 3.75	24	17	740	1620	10.93	VOIP
F008 4.65	31	25	880	1950	12.04	VPN-VoIP

Concept drift is change in statistical relationships between the network traffic features and traffic classes. Three types of drift are explored:

- **Sudden drift:** When there is a sudden change in the characteristics of the network traffic.
- **Gradual drift:** A progressive transition from an existing traffic pattern to a new pattern is known as a gradual drift.
- **Incremental drift:** Continuous evolution of traffic characteristics (incremental drift).

Table 3: Configuration of Concept drift detection

Drift type	Description	Detection Mechanism	Expected Model response
Smooth drift update	Slow change of traffic distribution	ADWIN/DDM	Slow model update
Slow change in traffic pattern	Progressive change in traffic pattern	ADWIN/Page-Hinkley	Incremental adaptation
Incremental Drift	Continuous Evolution of traffic behavior	Pagehinkley/ ADWIN	Continuous Model adjustment

The chronologically ordered traffic streams are used for evaluation of the detectors. Changes in natural data sets are separated from changes imposed by controlled drift scenarios during experiments.

In this section, we compare the performance of the classifiers under various concept drift conditions. The first experiment is to compare the Random Forest, SVM, CNN, LSTM, CNN-LSTM and the adaptive model, under different traffic conditions.

Table 4: Comparison of F1-scores for different concept drift

Models	Initial Distribution (%)	Sudden Drift (%)	Gradual drift (%)	Incremental Drift (%)
Random Forest	92.1	76.4	79.2	81.0
SVM	90.5	74.8	77.5	79.3
CNN	94.2	80.1	83.4	85.2
LSTM	93.6	81.3	84.1	85.8
CNN-LSTM	95.1	83.5	86.2	87.6
Proposed adaptive model	96.4	92.1	93.5	94.2

The results indicate that the performance of conventional models decreases after drift events. Different performance evaluation approaches in the presence of limited labeled data are discussed. Different traffic sample proportions (1%, 5%, 10%, 20%, and 100%) of the traffic samples are used in this experiment to study the classification performance. The rest of observations are left unlabeled.

Table 5: Results of classification under varying proportions of labeled data

Labeled data	Random forest F1(%)	CNN F1(%)	LSTM F1(%)	CNN-LSTM F1(%)	Proposed Model F1(%)
1%	70.2	73.5	72.1	75.4	84.2
5%	78.5	82.1	80.4	83.6	89.5
10%	84.3	87.5	86.2	88.4	92.1
20%	89.1	91.2	90.5	92.3	94.3
100%	92.1	94.2	93.6	95.1	96.4

As can be seen from the results, the more the labeled data, better is performance. However, even with a small number of labeled samples, the experimented framework demonstrates higher F1-scores, indicating its potential use in scenarios where labeling is costly or time-consuming.

4.2 Combined Evaluation under Concept Drift and Limited Labeled Data:

This experiment assesses effects of both concept drift and restricted labeled data. The CNN-LSTM model is compared with proposed adaptive framework for sudden, gradual and incremental drift conditions.

Table 6: F1 score with varying drift conditions and proportions of labeled data

Drift scenario labeled data (%)	CNN –LSTM F1(%)	PROPOSED Model f1(%)
Initial distribution 100%	95.1	96.4
Sudden drift 1%	65.2	80.5
Sudden drift 5%	73.4	87.2
Sudden drift 10%	78.6	90.1
Gradual drift 1%	68.5	82.3
Gradual drift 5%	76.2	88.4
Gradual drift 10%	81.5	91.3
Incremental drift 1%	70.1	83.5
Incremental drift 5%	78.3	89.2
Incremental drift 10%	83.2	92.0

The results indicate that the framework achieves better classification performance in cases where the drift also comes with a limited number of labelled data. This difference is more evident if only 1% or 5% of the newly arrived traffic samples are labeled.

4.3. Ablation Study:

An ablation study looks at configurations of drift detection and adaptive learning and their contribution.

Table 7: Ablation study of proposed framework

Precision (%)	**Recalls** (%)	**F1 Score** (%)
CNN-LSTM without Adaptation 89	88.5	87.9
CNN-LSTM + Drift Detection 92.4	91.8	91.2
CNN-LSTM + Adaptive Learning 94.1	93.5	93.2
Complete proposed Framework 96.4	96.0	96.2

4.4. Experimental Environment:

The framework is based on machine learning and deep learning libraries in python for preprocessing, model development, drift detection, and performance evaluation.

Table 8: Experimental tools and technologies

Component	**Technology**	**Purpose**
pylight	4.0.1	Framework implementation
Traditional Methods	Python & Sklearn	RNN, LSTM and CNN implementation
Machine Learning	Scikit-learn	Implementation of Random Forest and SVM
Adaptive Learning	River	Online learning and drift detection
Data Cleaning and Feature processing	Pandas, NumPy	Data Processing
Visualization	Matplotlib	Experimental result visualization

5. Analysis and Discussion:

Followings is the classification performance under concept drift: The proposed framework is compared with CNN-LSTM under various traffic conditions in Table 9. The proposed model shows F1-scores of 96.4%, 92.1%, 93.5%, and 94.2% for initial drift, sudden drop, gradual drift and incremental drift respectively. The greatest gain is for sudden drift (8.6 percentage points), indicating the potential value of drift detection and adaptive updating.

Table 9: Improvement over CNN-LSTM with respect to F1 -score

0.25 (no congestion)	**96.93 (CNN-LSTM)**	**97.69 (Proposed Model)**	**0.76 (CNN-LSTM)**
Initial Distribution	95.1	96.4	1.3
Sudden Drift	83.5	92.1	8.6
Gradual Drift	86.2	93.5	7.3
Incremental Drift	87.6	94.2	6.6

5.1. Limited Labeled Data Performance:

The F1-scores at different percentages of labeled data are shown in Table 10. The framework gets the best result when just 1% of the data is labelled. The better the improvement it is even on less amount of labeled data which shows its potential in label-scarce environment.

Table 10: Increase in F1 score with limited labelled data

Labeled Data	CNN-LSTM (%)	Proposed Model (%)	Improvement (Percentage Points)
1%	75.4	84.2	8.8
5%	83.6	89.5	5.9
10%	88.4	92.1	3.7
20%	92.3	94.3	2.0
100%	95.1	96.4	1.3

5.2. Computational Performance Analysis:

Table 11 shows that the proposed approach requires less computation than CNN-LSTM. The proposed model achieves 65.3% reduction in adaptation time, 70% in retraining frequency, 18 percentage points in CPU usage, 28.1% in memory usage and 29.7% in classification latency.

Table 11: Performance of CNN-LSTM model compared with proposed model

Parameter	CNN-LSTM		Proposed Model Difference
Adaptation Time (seconds)	19.6	6.8	65.3% lower
Retaining frequency	10	3	70% lower
CPU usage	64%	46%	18% lower
Memory usage	960 MB	690 MB	28.1% lower
Classification Latency	7.4 ms	5.2 ms	29.7% lower

IERR
Institute for Excellence in Education & Research

These results indicate that adaptive updating decreases amount of computation and unnecessary updating of the model.

5.3. Combined Analysis of Concept Drift and Limited Labeled Data:

The average F1-scores in sudden, gradual and incremental drift scenarios are shown in Table 6. The improvements at 1%, 5% and 10% labeled data are 14.2%, 12.3% and 10.0% respectively with the proposed framework. The best results are achieved with 1% labelled data.

Table 12: Average F1 Score from drift scenarios

Labelled Data	CNN-LSTM Average F1 (%)	Proposed Model Average F1 (%)	Improvement
1%	67.9	82.1	14.2 percentage
5%	76.0	88.3	12.3 percentage
10%	81.1	91.1	10.0 percentage

5.4. Ablation Study Analysis

The full system has the best accuracy (96.4%) and:

F1 score (96.1%). The results indicate that both the components play a role in the classification.

Table 13: Each Framework contribution

Frame work	True Positive Rate(%)	True negative rate(%)
CNN-LSTM without Adaptation	89.2	88.2
CNN-LSTM + Drift Detection	92.4	91.5
CNN-LSTM + Adaptive Learning	94.1	93.3
Complete Proposed Framework	96.4	96.1

Conclusion:

This work aims to present an adaptive deep learning approach for classifying encrypted network traffic since concept drift and lack of labeled data are the challenges to be addressed. The framework combines CNN-LSTM architecture, concept drift detection, and label-efficient adaptive learning to enhance classification performance under continuously changing network conditions. It also aims to make the model less dependent on frequent retraining while keeping the complexity of the calculations low and the delay of the process low. This way can serve as a basis for creation of more robust and flexible encrypted traffic classification systems. The future scope of work is implementation of the framework, testing it with benchmark datasets, and comparing the results with the existing machine learning and deep learning approaches

REFERENCES:

[1] A. Sharma and A. H. Lashkari, "A survey on encrypted network traffic: A comprehensive survey of identification/classification techniques, challenges, and future directions," Computer Networks, vol. 257, Art. no. 110984, 2025. https://doi.org/10.1016/j.comnet.2024.110984.

[2] C. Zou et al., "Encrypted Network Traffic Analysis and Classification Utilizing Machine Learning," Sensors, vol. 24, no. 11, Art. no. 3509, 2024. https://doi.org/10.3390/s24113509.

[3] "Machine Learning-Powered Encrypted Network Traffic Analysis: A Comprehensive Survey," IEEE Communications Surveys & Tutorials, vol. 25, no. 1, pp. 791–824, 2023. IEEE Xplore.

[4] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of encrypted and VPN traffic using time-related features," in Proc. 2nd Int. Conf. Information Systems Security and Privacy (ICISSP), 2016, pp. 407–414. https://doi.org/10.5220/0005740704070414.

[5] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," ACM Computing Surveys, vol. 46, no. 4, Art. no. 44, 2014.

[6]A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," in Proc. SIAM Int. Conf. Data Mining, 2007, pp. 443–448. https://doi.org/10.1137/1.9781611972771.42.

[7] G. Draper-Gil et al., "ISCX VPN-nonVPN 2016 dataset," Canadian Institute for Cybersecurity, University of New Brunswick. Dataset and documentation.

[8] P. Velan, M. Čermák, P. Čeleda, and M. Drašar, "A survey of methods for encrypted traffic classification and analysis," International Journal of Network Management, vol. 25, pp. 355–374, 2015. https://doi.org/10.1002/nem.1901.

[9] A. Sharma and A. H. Lashkari, "A survey on encrypted network traffic: A comprehensive survey of identification/classification techniques, challenges, and future directions," Computer Networks, vol. 257, Art. no. 110984, 2025. https://doi.org/10.1016/j.comnet.2024.110984.

[10] S. Rezaei and X. Liu, "Deep learning for encrypted traffic classification: An overview," arXiv preprint arXiv:1810.07906, 2018. https://arxiv.org/abs/1810.07906.

[11] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapè, "MIMETIC: Mobile encrypted traffic classification using multimodal deep learning," Computer Networks, vol. 165, Art. no. 106944, 2019. https://doi.org/10.1016/j.comnet.2019.106944.

[12] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," ACM Computing Surveys, vol. 46, no. 4, Art. no. 44, 2014. https://doi.org/10.1145/2523813.

[13] A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," in Proceedings of the SIAM International Conference on Data Mining, 2007, pp. 443–448.

[14] "Few-shot encrypted traffic classification: A survey," in Proceedings of the 2024 Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), 2024. https://doi.org/10.1109/IPEC61310.2024.00115.

[15] M. Rahaman, A. Mahmud, A. Abd Aziz, O. M. S. Abujawa, and J.-J. Chin, "A systematic literature review on data-efficient and adaptive learning techniques for encrypted traffic classification under modern protocols," Computers, vol. 15, no. 5, Art. no. 319, 2026. https://doi.org/10.3390/computers15050319.

[16] R. Enisoglu and B. A. Inan, "Network traffic classification from handcrafted features to foundation models: A comprehensive survey," Computer Networks, vol. 287, Art. no. 112578, 2026. https://doi.org/10.1016/j.comnet.2026.112578.

IEER
Institute for Excellence in Education & Research