Optimal Classifier Alignment via Neural Collapse for Incremental Encrypted Traffic Classification

1st Qin Zeng
Information Engineering University
ZhengZhou, China
zengqin2361@163.com

2nd Yaqi Chen
Information Engineering University
ZhengZhou, China
chyaqi163@163.com

3rd Hao Zhang
Information Engineering University
ZhengZhou, China
haozhang0126@163.com

4rd Chaolong Hao
Information Engineering University
ZhengZhou, China
hcl_xdspeechlab@aliyun.com

5th Dan Qu*
Information Engineering University
ZhengZhou, China
qudan_xd@163.com

***Abstract*—The rapid evolution of internet technologies has intensified network traffic dynamics due to the emergence of novel encryption protocols, posing significant challenges to traffic classification. Incremental learning, which enables continuous adaptation to emerging tasks, has emerged as a promising approach to enhance the sustainability of encrypted traffic classification. However, existing methods fail to address the substantial feature representation disparities across incremental tasks, resulting in suboptimal model adaptability. This paper propose NCIL-ETC, a Neural Collapse-based Incremental Learning framework for Encrypted Traffic Classification. Our approach employs a pre-trained Mamba as the feature extraction backbone, leveraging its linear-complexity computational properties to significantly reduce resource overhead. Simultaneously, we introduce a pre-allocated ETF classifier that establishes an optimal classification structure covering observed classes. Through feature-classifier alignment constraints during incremental learning, our method enforces both new and historical class features to converge toward ETF vertices, thereby preserving globally optimal category relationships. Extensive experimental evaluations on two public benchmarks (CIC-IDS2017 and USTC-TFC) demonstrate that NCIL-ETC achieves state-of-the-art performance, surpassing baseline methods in both classification accuracy and incremental learning capability.**

***Index Terms*—encrypted traffic classification, class incremental learning, neural collapse.**

I. Introduction

With the advancement of information technologies, encrypted traffic classification has become a critical component in network management and cybersecurity [1], [2]. The ultimate objective is to categorize encrypted traffic from diverse network services and applications, enabling identification of application sources and user behaviors. However, the continuous emergence of new network services and applications necessitates classifiers to adapt to novel traffic types, requiring retraining on newly collected extended datasets. Retraining entire models demands substantial time and computational resources, especially for self-supervised pre-trained encrypted traffic classification frameworks [3], [4], posing significant challenges to comprehensive network traffic classification.

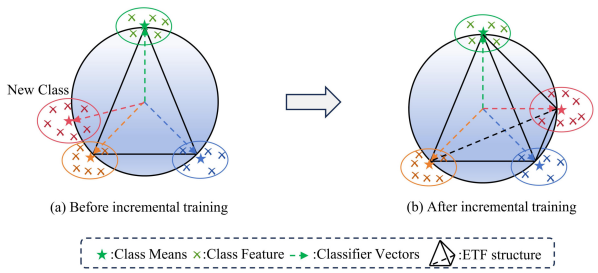


Fig. 1: (a) Before incremental training, the small discrepancy between the features of new tasks and old tasks makes it difficult for the model to distinguish them. (b) After incremental training, the feature vectors of new categories learn toward a predefined equiangular tight frame classifier structure, achieving neural collapse where all categories exhibit an optimal classification arrangement.

Recent years have witnessed the application of class-incremental learning (CIL) in computer vision and related domains [5], [6]. Built upon continual learning paradigms, CIL enables models to incrementally adapt to new classes while mitigating catastrophic forgetting of historical knowledge. Although CIL has been explored in encrypted traffic classification with promising results [7], existing approaches address catastrophic forgetting through diverse strategies including regularization, replay, and parameter isolation. Regularization methods [8] constrain parameter changes to preserve old knowledge but hinder new feature learning when inter-class correlations are weak. Replay-based approaches [9] mix stored or generated samples with incremental data, though they require large memory buffers and depend heavily on sample selection or generation quality. Parameter isolation techniques [10] increase model complexity through task-specific parameters, leading to parameter accumulation with

*Dan Qu is the corresponding author.

incremental steps. Moreover, as illustrated in Fig.1(a), these methods fail to account for significant feature representation disparities between old and new tasks, thereby unable to guarantee optimal inter-task category relationships.

To address this limitation, Neural Collapse [11] reveals that final-layer features of deep neural networks collapse to class-mean vectors aligned with classifier prototypes, forming a Simplex Equiangular Tight Frame (ETF) structure characterized by maximum and uniform inter-class separability. Inspired by this geometric insight, we propose NCIL-ETC —a Neural Collapse-based Class-Incremental Learning framework for Encrypted Traffic Classification. As shown in Fig.1(b), our approach introduces a pre-allocated ETF classifier that establishes an optimal structure for all observed classes. Through feature replay strategies during incremental training, we maintain cross-task optimal category relationships. This method employs an incremental encrypted traffic classification network architecture. First, a Mamba-based feature extraction module derives generalizable encrypted traffic features. Then, a feature mapping module projects these unified features into a ETF classifier.

The key contributions of this paper are as follows:

- To the best of our knowledge, this work represents the first integration of Neural Collapse theory into incremental encrypted traffic classification.
- We devise a Mamba-based pre-trained feature extractor and introduce a ETF classifier to to ensure the capture of the most favorable class relationships in each learning process.
- We evaluate the proposed framework on two public datasets (CIC-IDS2017 [12] and USTC-TFC [13]), in comparison with other commonly-used approaches. Experimental results demonstrate that our method delivers the optimal performance in both incremental learning and anti-forgetting capabilities.

II. Related Work

A. Encrypted Traffic Classification

Due to the opaque nature of encrypted traffic, most early studies employ machine learning (ML) techniques that rely on manually designed statistical features. Kong et al. [14] extracted 41 IP traffic-related features for anomaly detection using support vector machines (SVMs). Taylor et al. [15] proposed AppScanner, a random forest-based method for real-time smartphone app identification from encrypted traffic. However, these feature-engineered approaches face generalization challenges in novel scenarios due to their dependency on predefined feature sets.

Deep learning methods automate feature extraction from raw traffic data, eliminating manual feature engineering. Wang et al. [16] transformed the first 784 bytes of traffic payloads into 28×28 grayscale images, applying 1D-CNNs for end-to-end classification. Liu et al. [17] introduced FS-Net, leveraging encoder-decoder architectures to capture latent sequential patterns. Graph-based techniques include DE-GNN [18], which models traffic interaction graphs (TIGs), and TFE-GNN [19], which constructs byte-level graphs for structural feature learning. While effective, these supervised methods require extensive labeled data.

Pre-trained models have gained prominence by learning latent representations from unlabeled traffic data. ET-BERT [3] employs BERT to pre-train on traffic burst patterns, while YaTC [4] uses masked autoencoding (MAE) with matrix-form representations (MFR).

B. Class-Incremental Learning

Class-incremental learning, a core paradigm of continual learning, addresses catastrophic forgetting—the tendency of models to overwrite previously learned knowledge when trained on new tasks. Existing methods can be categorized into three groups: regularization, replay, and parameter isolation.

Regularization: Regularization-based methods prevent the overwriting of old knowledge caused by overfitting to new tasks by constraining the update of model parameters. Li et al. [20] proposed Learning without Forgetting (LwF), which uses predictions from the old model on new tasks as soft labels, preserving old-task discriminability via distillation loss. Elastic Weight Consolidation (EWC) [21] quantifies parameter importance for old tasks using the Fisher information matrix, applying stronger regularization to critical parameters.

Replay: Replay-based methods alleviate forgetting by storing samples from old tasks or generating pseudo-samples to simulate joint training. iCaRL [22] combines nearest-mean classification and knowledge distillation. It employs a memory buffer to store samples from old classes, thereby balancing the influence of new and old samples. Chen et al. [23] introduced a framework called One vs Rest (OvR). This framework utilizes samples with a herding effect to automatically adjust the dataset capacity while updating the classifier.

Parameter Isolation: Parameter-isolation-based methods offer dedicated space for acquiring new knowledge by gradually expanding the network scale or increasing model parameters, while ensuring that the previously learned information remains undisturbed. Wang et al. [24] proposed a novel two-stage learning paradigm called FOSTER, which achieves a balance between knowledge retention and new-class learning through dynamic module expansion and knowledge distillation. Zhou et al. [25] proposed the SimpleCIL method, which utilizes frozen pre-trained models, and the Aper method, which efficiently fine-tunes pre-trained models and constructs classifier prototypes. Both methods make full use of the generalization ability of pre-trained models.

In recent years, some researchers have introduced incremental learning techniques into encrypted traffic classification. Chen et al. [14] prevented catastrophic forgetting by sampling and storing training samples to replay the knowledge of old classes. Du et al. [10] extended the classifier for class-incremental learning in network intrusion detection by designing a parameter-isolation approach with branched classifiers. Cai et al. [26] addressed the issues of traffic ambiguity and data class imbalance in incremental encrypted classification

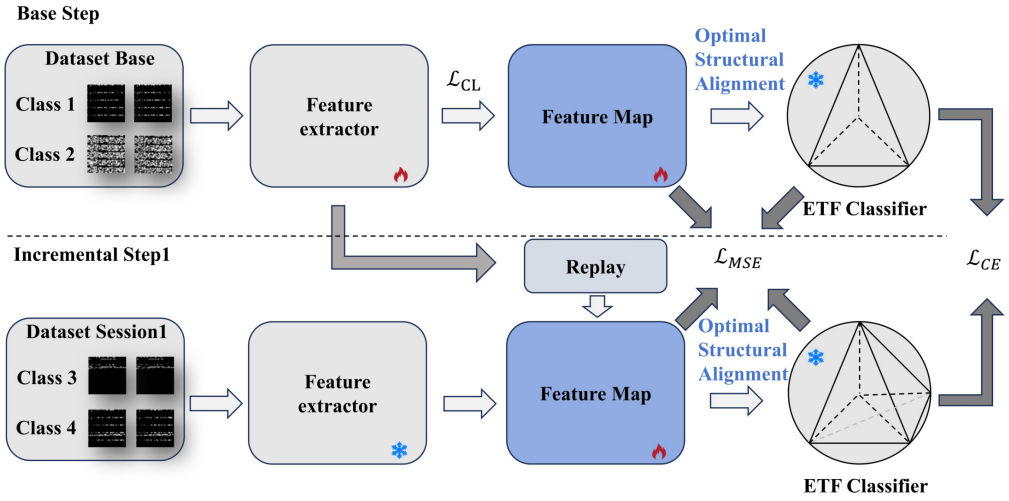


Fig. 2: Overview of NCIL-ETC. The model architecture primarily consists of a feature extractor, a feature mapping module, and an ETF classifier. During the base training phase, the feature extractor is fine-tuned to adapt to downstream tasks, while it is frozen in each incremental learning step to ensure the generalization capability of the extracted features. The feature mapping module aligns the features with the optimal structure, satisfying the neural collapse properties. In the incremental learning phase, it is fine-tuned through a feature replay strategy to continuously align with the prototypes of the ETF classifier.

through pre-allocating vector prototypes and applying knowledge distillation.

The method proposed in this paper improves upon those based on replay and parameter isolation. By freezing the feature extraction module and efficiently generating pseudo-samples based on stored class vector prototypes for incremental learning training, it significantly reduces the spatial complexity associated with storing old samples. Moreover, due to the introduction of a frozen ETF classifier in this paper, it ensures the optimal relationships among classes without increasing model parameters during the incremental process.

III. Preliminaries

A. Equiangular Tight Frame Classifier

Neural Collapse refers to the phenomenon where, after a model is trained on balanced data, the features in the model's last layer and the classifier form a specific geometric structure. This structure can be defined as follows:

Definition 1: A simplex equiangular tight frame refers to a matrix composed of K vectors in the R^d space that satisfies the following conditions:

$$E = \sqrt{\frac{K}{K-1}} U \left(I_K - \frac{1}{K} 1_K 1_K^T \right) \tag{1}$$

Let $E = [e_1, \ldots, e_K] \in \mathbb{R}^{(d \times K)}$, and $U \in \mathbb{R}^{d \times K}$ allows for rotation and satisfies $U^T U = I_K$, where I_K is the identity matrix and 1_K is the all-ones vector.

All column vectors in E have the same l_2 norm, and the inner product between any pair of vectors is $-\frac{1}{K-1}$, that is:

$$e_{k_1}^T e_{k_2} = \frac{K}{K-1} \delta_{k_1 k_2} - \frac{1}{K-1}, \forall k_1, k_2 \in [1, K] \tag{2}$$

such that $\delta_{k_1 k_2} = 1$ when $k_1 = k_2$ and $\delta_{k_1 k_2} = 0$ otherwise.

Neural collapse has revealed the optimal geometric structure between the features of the model's final layer and the classifier. Therefore, we introduce an ETF classifier, which is initialized as a simplex ETF classifier during the base step and remains fixed throughout the training process to preserve this optimality. For newly added classes in the incremental steps, we assign them pre-reserved optimal classification vectors as guidance to ensure the transfer of the optimal class structure with features.

IV. Method

In this section, we present the encrypted traffic representation, model architecture and training methodology in Sections IV-A, IV-B and IV-C respectively. Fig. 2 provides an overview of the NCIL-ETC method.

A. Encrypted Traffic Representation

In this section, we propose a simple yet effective traffic representation method that converts raw traffic bytes into grayscale images while preserving header and interaction information. We partition the traffic into bidirectional flows based on the five-tuple and filter out some network flows that contain no valid information or are erroneous. The process of converting raw traffic bytes into grayscale images is illustrated in Fig. 3. First, we take the first B_0 bytes from the first M packets of each bidirectional flow. If a packet has insufficient bytes, we pad it with zero bytes; otherwise, we truncate the excess bytes. Next, we reshape the B_0 bytes in each packet into a matrix $\in \mathbb{R}^{h \times W}$. Finally, we concatenate the samples to form a grayscale image $\in \mathbb{R}^{1 \times (M \times h) \times W}$. For ease of processing, we typically set the image to be square, with

$W = H = M \times h$. Additionally, for security and privacy considerations, we replace the MAC and IP addresses in the packets with all zeros.

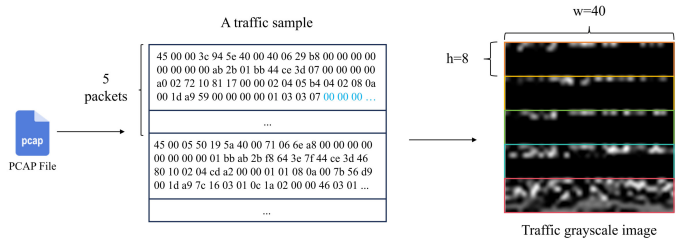


Fig. 3: The process of converting raw traffic into grayscale images

B. Model Architecture

The model architecture of the proposed method in this paper consists of three main components: feature extractor, feature mapping module, and classifier.

The Feature Extractor (FE) is implemented based on the Mamba model. Mamba [27] is a state space model. Compared to the quadratic computational complexity of the Transformer's attention mechanism, it boasts efficient linear computational complexity.

To make the model conform to the characteristics of neural collapse, we introduce a Feature Mapping module (FM), denoted as $P(\cdot)$. This module consists of two layers of Multi-Layer Perceptrons (MLPs) and a batch normalization layer. Positioned after the feature extractor, the Feature Mapping module is used to map features to the ETF classifier and align them with the pre-allocated prototypes in the ETF classifier:

$$z_i = f(x_i), z_i^{'} = P(z_i) \tag{3}$$

where, z_i and $z_i^{'}$ represent the extracted and mapped features, respectively, and $f(\cdot)$ represents the feature extractor.

In the classifier part, we introduce ETF classifier as the classifier to guide the optimization of feature representations. The vectors in the classifier are defined by Eq.(1).

C. Training Method

During the pre-training phase, this paper adopts Masked Autoencoder(MAE) pre-training approach from the field of computer vision for unsupervised pre-training, aiming to obtain generalized traffic feature representations.

In the base step, we fine-tune the feature extractor to adapt it to the classification task and bridge the domain gap, while freezing the feature extractor during incremental learning to preserve generalized feature representations.

To enable the feature extractor to achieve a better feature distribution representation, we introduce a contrastive learning loss to guide the training of the feature extractor in the base stage:

$$\mathcal{L}_{CL} = -\sum_{i=1}\sum_{j\neq i}\beta_{ij} log \frac{e^{sim(z_i,z_j)/\tau^{'}}}{\sum_{k\neq i} e^{sim(z_i,z_k)/\tau^{'}}} \tag{4}$$

where, β_{ij} is a binary indicator that denotes whether z_i and z_j belong to the same class, $sim(\cdot)$ calculates the cosine similarity, and τ is the temperature coefficient for contrastive learning.

To prevent significant variations in the ETF classifier weights during the incremental steps, we allocate ETF weights for each class during the base stage. During incremental learning, we assign new weights to the newly emerging classes while keeping the weights of the existing classes unchanged. We employ both the Mean Squared Error (MSE) loss and the Cross-Entropy (CE) loss to guide the optimal structural alignment:

$$\mathcal{L}_{MSE} = \sum_i (z_i^{'} W_p - e_k)_i^2 \tag{5}$$

$$\mathcal{L}_{CE} = BCE((z_i^{'} W_p)^T E, y_i) \tag{6}$$

where, W_p represents the weights of the feature mapping module, and e_k denotes the ETF classifier weight corresponding to the respective class.

The final objective function of the model during the base step is expressed as:

$$\mathcal{L}_{BASE} = \mathcal{L}_{MSE} + \mathcal{L}_{CE} + \xi \mathcal{L}_{CL} \tag{7}$$

where, ξ is the hyperparameter for contrastive learning.

To address the catastrophic forgetting problem, we adopt a feature replay strategy, consistent with the approach in [28], by replaying features from previous tasks during current training. Specifically, we compute the mean μ and covariance $\sum$ of the features for all existing classes extracted by the feature extractor for use in replay. Given that the pre-trained model can generate well-distributed and highly generalizable representations, and each class tends to exhibit a unimodal distribution. Therefore, during incremental steps, we generate synthetic samples from Gaussian distributions based on the mean and variance of each known class, combining them with the original training set for joint training. The training objective function is defined as:

$$\mathcal{L}_{CIL} = \mathcal{L}_{MSE} + \mathcal{L}_{CE} \tag{8}$$

V. Experiments

A. Dataset and Experiment Setup

Datasets: We utilize the CIC-IDS2017 [12] and USTC-TFC [13] datasets as benchmark datasets. The processing for each dataset is as follows:

The CIC-IDS2017 dataset contains 12 classes in total, with the base session comprising 8 classes and the new session including 4 classes, with 2 class per session.

The USTC-TFC dataset contains a total of 20 classes, with the base session set to 12 classes and the new session containing 8 classes, with 4 classes per session.

Comparison Baselines: We select two regularization methods (EWC [21] and LwF [20]), two replay methods (OvR [23] and iCaRL [22]), and one parameter isolation method (FOSTER [24] and SimpleCIL [25]) as baselines for comparison.

Implementation Details: In traffic representation, we extract the first B bytes from the first M packets of bidirectional flows, padding with zero bytes when insufficient or truncating otherwise. Each packet's B bytes are reshaped into a matrix $\in \mathbb{R}^{h \times W}$, and samples are ultimately concatenated into grayscale images $\in \mathbb{R}^{1 \times (M \times h) \times W}$. This paper sets $M = 5$ packets and $B = 320$ bytes. The MAE pre-training process is conducted on USTC-TFC dataset training set, and the MAE encoder employs four Mamba blocks, while the decoder uses two. All experiments are conducted in a Python 3.9 and PyTorch environment, with the Adam optimizer used for optimization.

Evaluation Metrics: To quantitatively assess the performance and anti-forgetting capabilities of different methods, following [10], we employ four typical metrics: accuracy (AC), precision (PR), recall (RC), and F1 score, along with the performance degradation rate (PD) to evaluate and compare model performance. PD is used to measure the accuracy drop rate of the sessions, that is, PD=A-B, where A is the accuracy of the base session, and B is the accuracy of the last incremental session. Additionally, we employ the normalized confusion matrix to specifically analyze the classification performance of each class during the incremental process, where the y-axis corresponds to the true classes and the x-axis to the predicted classes.

TABLE I: Comparison of Baseline Methods on Two Datasets.

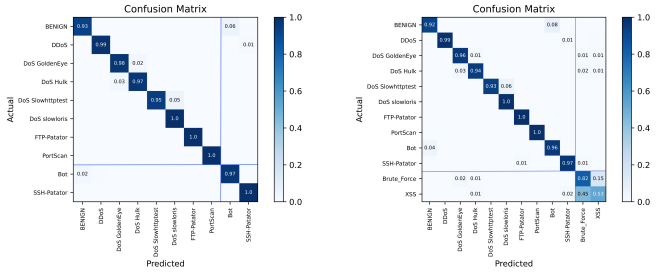
Method	CIC-IDS2017				USTC-TFC			
	Acc in each session(%)			PD↓	Acc in each session(%)			PD↓
	0	1	2		0	1	2	
Finetune	98.60	80.24	65.78	32.82	99.45	74.13	63.71	35.74
EWC [21]	98.60	88.47	74.68	23.92	99.45	83.65	72.34	27.11
LwF [20]	98.60	89.22	78.39	20.21	99.45	84.48	75.86	23.59
OvR [23]	98.60	90.81	82.23	16.37	99.45	87.69	83.53	15.92
iCaRL [22]	**98.85**	91.34	83.88	14.97	99.53	88.02	84.91	14.62
FOSTER [24]	98.60	95.16	90.34	8.26	99.45	92.53	89.92	9.53
SimpleCIL [25]	98.60	93.62	89.37	9.23	99.45	92.38	88.84	10.61
NCIL-ETC	**98.85**	**97.83**	**95.01**	**3.84**	**99.97**	**97.02**	**96.07**	**3.9**

B. Incremental Learning Capability Analysis

The results are illustrated in Table I. Our method achieves the highest accuracy and the lowest PD value across all session rounds, outperforming the second-best baseline by 4.42% and 5.63% on the two datasets, respectively. Fine-tuning suffers from severe catastrophic forgetting as it focuses solely on new classes. While EWC and LwF preserve prior knowledge through regularization, they constrain the learning of new information. OvR and iCaRL strike a balance between old and new classes by retaining selected old samples, yet their performance depends heavily on the quality and quantity of these samples. FOSTER and SimpleCIL introduces new parameters while freezing old ones, but increases model size and overlooks optimal inter-class relationships. In contrast, our approach leverages the strong feature representations of pre-trained models and optimizes class relationships based on neural collapse during each learning phase, leading to superior performance on encrypted traffic classification tasks compared to all baseline methods.

TABLE II: Ablation Study on the Proposed Modules. FT: Fine-tuning the feature extractor, ETF: ETF classifier, FM: feature mapping module, OSA: optimal structure alignment. √: Module is enabled, ×: Module is disabled.

FT	ETF	FM	OSA	CIC-IDS2017				USTC-TFC			
				Acc in each session(%)			PD↓	Acc in each session(%)			PD↓
				0	1	2		0	1	2	
✓	✗	✗	✗	98.60	85.53	66.39	32.21	99.45	74.13	63.71	35.74
✗	✗	✓	✓	98.60	91.47	80.18	18.42	99.45	84.38	78.26	21.19
✓	✓	✗	✗	**98.85**	96.52	88.93	9.92	**99.97**	95.96	93.35	6.62
✗	✓	✓	✗	**98.85**	97.20	91.84	7.01	**99.97**	96.25	94.59	4.71
✗	✓	✓	✓	**98.85**	**97.83**	**95.01**	**3.84**	**99.97**	**97.02**	**96.07**	**3.9**



(a) CIC-IDS2017 Session 1. (b) CIC-IDS2017 Session 2.

Fig. 4: Normalized confusion matrices on CIC-IDS2017 dataset between two sessions. We use bold blue lines to separate regions of the base session classes and incremental classes of new sessions.

C. Ablation Analysis

We conducted ablation experiments on two datasets to analyze the effectiveness of different modules in the model, with the results presented in Table II. For ease of representation, we define fine-tuning the feature extractor, ETF classifier, feature mapping module, and optimal structure alignment as FT, ETF, FM, and OSA. Using the ETF classifier reduced PD by 8.5% and 14.57%, demonstrating its efficacy in feature alignment. When freezing the feature extractor and using the feature mapping module, the PD values further decreased by 2.91% and 1.91%, demonstrating the strong generalization capability of the pre-trained model. Incorporating OSA yielded additional reductions of 3.17% and 0.81%, indicating its role in guiding optimal feature structure for each class.

D. False Positive and False Negative Analysis

In cybersecurity applications, False Positives and False Negatives are critical metrics determining the deployability of Intrusion Detection Systems. A high false negative rate leads to failures in timely intrusion detection, risking the theft of critical assets or sensitive data. Conversely, excessive false positives impose prohibitive costs on human analysis and response time.

So we employ normalized confusion matrices to evaluate the false positive and false negative performance across different incremental learning tasks. Since the accuracy in the base session phase approaches 1, we focus on illustrating the confusion matrices for the subsequent two incremental stages. As shown in Fig. 4, the confusion matrix for the CIC-IDS2017 dataset utilizes thick blue lines to demarcate the base classes from the new classes introduced in subsequent sessions.

Our analysis first examines the False Negative Rate (FNR) of NCIL-ETC. In the CIC-IDS2017 dataset, the FNR for Session 1 and Session 2 are 2% and 4%, respectively, with a maximum FNR of only 4%. Furthermore, regarding the false positives between benign and attack categories, benign samples in Session 1 and Session 2 were misclassified as Bot attacks with probabilities of 6% and 8%, respectively. In conclusion, these results demonstrate that our proposed method maintains robust and superior classification capabilities even after multiple incremental learning phases.

VI. Conclusion

In this paper, we simultaneously address encrypted traffic classification and incremental learning, proposing an incremental learning method for encrypted traffic classification that utilizes optimal structural alignment to align traffic features with the ETF classifier, thereby achieving optimal category relationships. Experimental results on two encrypted traffic classification datasets demonstrate that NCIL-ETC exhibits the best incremental performance.

Acknowledgment

This work was supported by the Henan Zhongyuan Science and Technology Innovation Leading Talent Project (No.234200510019).

References

[1] E. Papadogiannaki and S. Ioannidis, "A survey on encrypted network traffic analysis applications, techniques, and countermeasures," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 123:1–123:35, 2022.

[2] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-to-end encrypted traffic classification with one-dimensional convolution neural networks," in *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*. 2017, pp. 43–48, IEEE.

[3] X. Lin, G. Xiong, G. Gou, Z. Li, J. Shi, and J. Yu, "Et-bert: A contextualized datagram representation with pre-training transformers for encrypted traffic classification," in *ACM Web Conference (WWW)*. 2022, pp. 633–642, ACM.

[4] R. Zhao, M. Zhan, X. Deng, Y. Wang, Y. Wang, G. Gui, and Z. Xue, "Yet another traffic classifier: A masked autoencoder based traffic transformer with multi-level flow representation," in *Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI)*. 2023, pp. 5420–5427, AAAI Press.

[5] P. Mazumder, P. Singh, and P. Rai, "Few-shot lifelong learning," in *Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*. 2021, pp. 2337–2345, AAAI Press.

[6] D.-W. Zhou, H.-J. Ye, L. Ma, D. Xie, S. Pu, and D.-C. Zhan, "Few-shot class-incremental learning by sampling multi-phase tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 12816–12831, 2023.

[7] W. Zhu, X. Ma, Y. Jin, and R. Wang, "Iletc: Incremental learning for encrypted traffic classification using generative replay and exemplar," *Comput. Networks*, vol. 224, pp. 109602, 2023.

[8] Y. Chen, Y. Tong, H. Gwee Bah, Q. Cao, S. G. Razul, and Z. Lin, "Encrypted mobile traffic classification with a few-shot incremental learning approach," in *2023 IEEE 18th Conference on Industrial Electronics and Applications (ICIEA)*, 2023, pp. 40–45.

[9] G. Sun, T. Chen, Y. Su, and C. Li, "Internet traffic classification based on incremental support vector machines," *Mob. Networks Appl.*, vol. 23, no. 4, pp. 789–796, 2018.

[10] L. Du, Z. Gu, Y. Wang, L. Wang, and Y. Jia, "A few-shot class-incremental learning method for network intrusion detection," *IEEE Trans. Netw. Serv. Manag.*, vol. 21, no. 2, pp. 2389–2401, 2024.

[11] V. Papyan, X. Y. Han, and D. L. Donoho, "Prevalence of neural collapse during the terminal phase of deep learning training," *CoRR*, vol. abs/2008.08186, 2020.

[12] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *International Conference on Information Systems Security and Privacy*, 2018.

[13] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *2017 International Conference on Information Networking (ICOIN)*. 2017, pp. 712–717, IEEE.

[14] L. Kong, G. Huang, and K. Wu, "Identification of abnormal network traffic using support vector machine," in *18th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*. 2017, pp. 288–292, IEEE Computer Society.

[15] V. F. Taylor, R. Spolaor, M. Conti, and I. Martinovic, "Appscanner: Automatic fingerprinting of smartphone apps from encrypted network traffic," in *IEEE European Symposium on Security and Privacy (EuroS&P)*. 2016, pp. 439–454, IEEE.

[16] W. Wang, M. Zhu, X. Zeng, X. Ye, and Y. Sheng, "Malware traffic classification using convolutional neural network for representation learning," in *2017 International Conference on Information Networking (ICOIN)*. 2017, pp. 712–717, IEEE.

[17] C. Liu, L. He, G. Xiong, Z. Cao, and Z. Li, "Fs-net: A flow sequence network for encrypted traffic classification," in *IEEE INFOCOM Conference on Computer Communications*, 2019, pp. 1171–1179.

[18] X. Han, G. Xu, M. Zhang, Z. Yang, Z. Yu, W. Huang, and C. Meng, "De-gnn: Dual embedding with graph neural network for fine-grained encrypted traffic classification," *Comput. Networks*, vol. 245, pp. 110372, 2024.

[19] H. Zhang, L. Yu, X. Xiao, Q. Li, F. Mercaldo, X. Luo, and Q. Liu, "Tfe-gnn: A temporal fusion encoder using graph neural networks for fine-grained encrypted traffic classification," *CoRR*, vol. abs/2307.16713, 2023.

[20] Z. Li and D. Hoiem, "Learning without forgetting," in *14th European Conference on Computer Vision (ECCV)*. 2016, vol. 9908 of *Lecture Notes in Computer Science*, pp. 614–629, Springer.

[21] J. Kirkpatrick, R. Pascanu, N. C. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," *CoRR*, vol. abs/1612.00796, 2016.

[22] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 5533–5542, IEEE Computer Society.

[23] Y. Chen, T. Zang, Y. Zhang, Y. Zhou, L. Ouyang, and P. Yang, "Incremental learning for mobile encrypted traffic classification," in *IEEE International Conference on Communications (ICC)*. 2021, pp. 1–6, IEEE.

[24] F.-Y. Wang, D.-W. Zhou, H.-J. Ye, and D.-C. Zhan, "Foster: Feature boosting and compression for class-incremental learning," in *17th European Conference on Computer Vision (ECCV)*. 2022, vol. 13685 of *Lecture Notes in Computer Science*, pp. 398–414, Springer.

[25] D.-W. Zhou, Z.-W. Cai, H.-J. Ye, D.-C. Zhan, and Z. Liu, "Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need," *Int. J. Comput. Vis.*, vol. 133, no. 3, pp. 1012–1032, 2025.

[26] W. Cai, C. Hou, M. Cui, B. Wang, G. Xiong, and G. Gou, "Incremental encrypted traffic classification via contrastive prototype networks," *Comput. Networks*, vol. 250, pp. 110591, 2024.

[27] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *CoRR*, vol. abs/2312.00752, 2023.

[28] K. Wei, Z. Xu, and C. Deng, "Compress to one point: Neural collapse for pre-trained model-based class-incremental learning," in *AAAI-25 Conference on Artificial Intelligence*. 2025, pp. 21465–21473, AAAI Press.