Article:

Lu, H. orcid.org/0009-0009-9245-1389, Dong, Y. orcid.org/0000-0003-4898-331X, Wu, Z. et al. (2 more authors) (2025) New class detection in network traffic classification using confidence information embedded cascade structure. IEEE Transactions on Network Science and Engineering. ISSN 2327-4697

https://doi.org/10.1109/tnse.2025.3538564

26/01/2025 Accepted for publication on Transactions on Network Science and Engineering

New class detection in network traffic classification using confidence information embedded cascade structure

Haotian Lu, Yuning Dong, Zhiyuan Wu, Hua-Liang Wei, Guanming Lu

***Abstract*—Network traffic classification plays an important role in network management. With continued emergence of new applications, classifiers need to deal with unknown classes in an open set environment. However, the available open set flow recognition methods cannot well balance the performance of new class detection and the fine-grained classification of known classes. Moreover, these methods could pursue high accuracy at the cost of the classification speed. To address these problems, this paper proposes an unknown network traffic detection method based on confidence (difference) and a cascade structure, by analyzing the confidence distributions of both the known and new classes. The proposed method works as follows. Firstly, it uses a cascade structure to detect new class samples (having high confidence) which are difficult to identify using existing methods; secondly, it employs the maximum confidence difference to classify the new and known classes. In order to better detect new classes with high confidence, an algorithm is designed to select the pseudo-negative samples from the unlabelled dataset with an adaptive threshold. The proposed method is evaluated on real-world datasets. The results show that compared with the state-of-the-art methods, the proposed method can significantly improve the overall accuracy and the classification latency is also greatly reduced.**

***Index Terms*—open set flow recognition; confidence difference; new class detection; unlabelled dataset; network traffic classification.**

I. Introduction

Network traffic classification (NTC) is very important for network management, such as quality of service (QoS) assurance and network resource allocation [1]. With the rapid development of the Internet and multimedia technologies, the number of network traffic classes is increasing, and new types of network applications are emerging continuously. Most existing methods pretrain classifiers on closed dataset [2]–[5], but due to a lack of knowledge of new applications, these methods cannot be directly applied to unknown traffic scenarios. Machine learning (ML) approaches, including deep learning (DL), generally require a large number of labeled samples. However, in real network scenarios, the labeled samples of unknown classes are not available. The flow recognition containing unknown classes is called open set flow recognition (OSFR) [6]. The OSFR requires a classification model that can not only classify the known classes, but also identify new classes without any auxiliary information [7].

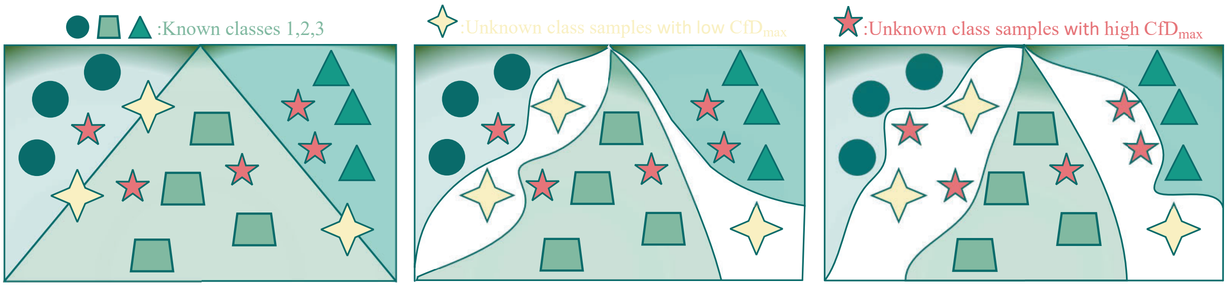
A. Motivation and challenges

Generally, the previous network traffic classification appear to be designed within the confines of a static dataset scenario, where the data classes of the overall framework are known in advance. As illustrated in Fig. 1(a), the three triangles positioned on the left, middle and right delineate the regions corresponding to the three known classes, and the green circles, trapezoids and triangles correspond to known classes 1, 2 and 3, respectively. It is evident that a classifier trained on a closed-set can only correctly classify known classes. However, the two kinds of unknown class samples (yellow four-pointed stars and red pentagrams) with distinct confidence differences are arbitrarily assigned to different known class regions. This discrepancy arises from the classifier lacking any prior knowledge of the unknown classes, resulting in the failure to identify them.

To achieve OSFR, some of the existing methods use single-class sample classification [8] (without negative samples) for detecting new classes. Such methods can detect new classes, and group all known classes into one class, but they cannot perform fine-grained classification. To address this problem, Yang et al. [9] proposed building a single-class classifier for each known class, which does not seem to work well.

In [10] and [11], a collective decision-based OSFR (CD-OSFR) model built on HDP (Hierarchical Dirichlet Process) was proposed, which does not require a classification threshold, and can automatically reserve space for new classes. However, CD-OSFR does not make full use of the labeled information of known classes, but only divides the training data into different groups. The performance of such a method still needs to be improved for new class detection and classification. In addition, collaborative clustering is used in the testing phase, which is time-consuming.

Another way is to generate negative samples by adversarial learning in an unsupervised/semi-supervised manner using adversarial sample generation-support vector machine (ASG-SVM) [12]. These negative samples generated are close to but not in the known classes. The detection of unknown classes is performed in a supervised way by training a support vector machine (SVM) classifier for each known class. Comparatively, the adversarial sample generation by ASG-based approach performs better in improving the detection rate for new classes. Nonetheless, as shown in Fig. 1(b), those classifiers trained in open environments


:Known classes 1,2,3
:Unknown class samples with low CfD$_{max}$
:Unknown class samples with high CfD$_{max}$

(a) Close-set methods in OSFR.

(b) Conventional methods in OSFR.

(c) CCS-UTD(Ours) in OSFR.

Fig. 1. Comparison of classification boundaries. In (a), the classifier trained on closed-set datasets separates three known classes (green circles, trapezoids, and triangles) into three regions (three triangles on the left, middle and right). It is evident that these methods fail to classify the two unknown classes (yellow four-pointed stars and red pentagrams) which are randomly assigned to the regions of known classes. In (b), some traditional OSFR methods can partition some easily distinguishable unknown class samples into new regions (the blank area in the figure), but there are still unknown class samples with high CfDmax that have not been correctly classified. In (c), CCS-UTD can identify most unknown class samples with high CfDmax and place them in new regions, further improving the accuracy of classification.

achieve certain success in distinguishing between known and unknown classes. Although the ASG-based approach can improve classification accuracy to some extent by generating fake new class samples close to the known class boundary, it has some limitations: 1) The generation time of negative samples is relatively long; 2) The classification performance still needs improvement. The challenge lies in the identification of the unknown class samples closer to the boundary of a known class (red pentagrams), leading to a limited classification space for unknown classes. Consequently, this constraint results in the misclassification of some unknown class samples.

To address the shortcomings of current methods, this paper develops a confidence information-embedded cascade structure, aiming at better detecting samples of novel classes that are easily confused with the samples of known classes. Specifically, through multiple screenings of training data and the introduction of two discriminators, the corresponding boundaries of each class shrink (as illustrated in Fig. 1(c)), which makes the classification space for unknown classes more open. As for the issue that the ASG process takes longer time, we design a novel algorithm to select pseudo-negative samples from unlabelled flow data that is readily available or easier to obtain from real-world networks, so as to achieve both higher new class detection rate and shorter preprocessing time.

B. Contributions of this article

In this paper, we propose a new OSFR method to perform online detection of new classes and fine-grained classification of known classes. With this method, by analyzing the confidence distribution patterns of known and new classes, the maximum confidence difference (CfDmax), i.e. the value difference between maximum and minimum confidences of the output labels, is used to distinguish between the known and new classes. For the known classes, the fine-grained classification is performed based on the confidence information of the random forest (RF) model. For the instances of new classes that cannot be detected by a CfDmax threshold, a binary classifier is designed to detect them. This binary classifier is trained using two types of data: one is the known class sample, and the other is the sample with maximum confidence difference higher than the threshold among the screened negative samples.

The main contributions of this paper are as follows.

(1) An unknown network traffic detection method using the confidence information-embedded cascade structure (CCS-UTD) is proposed, which can effectively detect new class instances while maintaining the fine-grained classification accuracy of known classes. Especially the introduction of cascade structure can effectively screen unknown class instances that are normally difficult to identify by existing methods and thus greatly improve the performance of new class detection.

(2) The confidence distribution patterns of known and new classes are analyzed, and a scheme to recognize the known and new classes using the maximum confidence difference information is designed. Meanwhile, a general threshold selection approach is proposed, which can provide a more accurate threshold to better distinguish known and new classes in different datasets.

(3) An algorithm is developed to select the pseudo-negative samples from the unlabelled flow data, which can not only make use of the easily available unlabelled flow data, but also greatly reduce the time of generating pseudo-negative samples for model training.

(4) To verify the performance of the proposed method, it is evaluated on five real network datasets, and compared with the state-of-the-art methods. The results show that the proposed method significantly outperforms the compared methods.

The rest of the paper is organized as follows: Section II reviews the relevant works on open set recognition. Section III presents the proposed method in detail, including the model framework, the principles of design and the specific implementation of the method. Section IV demonstrates the

TABLE I
MAIN NOTATIONS

Symbol	Description
Acc	Accuracy
CasStru	Cascade structure
Cfmax	Maximum confidence level
CfDmax	Maximum confidence difference
FE	Feature extraction
FS	Feature selection
HDP	Hierarchical Dirichlet Process
KnownC	Known class
NewC	New class
NTC	Network traffic classification
OSFR	Open set flow recognition

experiments to compare our method with the state-of-the-art methods, and shows the results. Section V concludes the paper.

For convenience of description, the main notations used in this paper are summarized in Table I.

II. Related Work

Currently, the methods based on ML [13] and DL [14] have become the mainstream NTC approaches because they do not need the information of port numbers and can be used for encrypted flows [15]. Chen et al. [16] proposed the concept of a flow bunch and developed a clustering method to process encrypted traffic. However, current methods still have limitations in the open set environment. Wu et al. [17] considered the issue of class imbalance in NTC, but ignored the emergence of new classes. In recent years, the OSFR has received more and more attention and becomes a hot topic in the field of machine learning at present. The current ML/DL-based OSFR methods commonly use SVM, adversarial learning or deep learing.

A. OSFR based on SVM

In OSFR, because the constructed hyperplane of SVM model tends to ignore the new classes, the new classes are often misclassified into known classes during the decision process. To solve this problem, researchers have proposed many SVM algorithms for OSFR by constraining the space occupied by the known classes [7], [18], [19].

The 1-vs-Set mechanism proposed by Scheirer et al. [18] is based on an SVM algorithm with linear kernel functions, which constrains the space occupied by the known class information to reduce the open space risk and cope with the single-class identification problem in an open environment. To achieve the multi-class recognition in an open environment, a Weibull-calibrated SVM (W-SVM) was proposed using the CAP model and EVT theory for probability estimation in [19]. It addresses the effect of openness on threshold selection.

B. OSFR based on adversarial learning

Wang et al. [20] integrated the generative adversarial network (GAN) with semi-supervised learning to achieve fine-grained NTC. Neal et al. [21] expanded the training set samples with the help of GAN to generate the pseudo open set samples. These samples are close to but do not belong to the known classes. Yang et al. [12] proposed an adversarial sample generation approach that can generate not only negative samples, but also positive samples of the known classes when there are few samples of the known classes.

C. OSFR based on deep learning

Chen et al. [22] employed a metric-based approach with a Siamese network (SEEN) to identify known and unknown traffic. A data skew-based classification method for TLS application unknown traffic (DSCU) [23] was proposed recently and created a classification space for unknown classes with its own constructed skew data. Le et al. [24] proposed an adaptive classification and updating method, constructing their own boundaries for each known class to achieve accurate identification of unknown flows in open network environments. Zhang et al. [25] designed a deep learning-based traffic clustering solution to classify unknown network traffic. Similar to our work, the classifier output is a vector of confidence scores that is further used in the traffic discriminator. For incoming traffic, if its score falls below a threshold, it is classified as an unknown flow; otherwise, it is recognized as a known flow. However, its classification space constructed using a single threshold is not accurate enough, resulting in misclassification of known and unknown classes. Thilini et al. [26] used well-regularized deep learning model to improve classification results of previous methods and developed a method based on k-logit neighbor distances (k-LND) for OSFR.

Although the above methods are effective in OSFR to a certain extent, they still have some shortcomings, as summarized in Table II. To address these shortcomings, a new method is proposed in this paper, which uses a binary classifier trained with pseudo-negative samples to detect new classes according to the maximum confidence difference, and exploits the maximum confidence to further classify the known classes.

TABLE II
COMPARISON OF EXISTING OSFR METHODS

Methods	Advantages	Disadvantages
SVM-based [7,18-19]	Simple structure	Classification accuracy (*Acc*) needs improvement
Adversarial learning, [12,21-22]	Good performance of new class detection	Longer training and inference time
DL-based [23-27]	Higher overall classification Acc	DL models are more complex

III. Methodology

A. Framework

The framework of the proposed method consists of two parts: the training phase and the testing phase, as illustrated in Fig. 2 and Fig. 3, respectively.

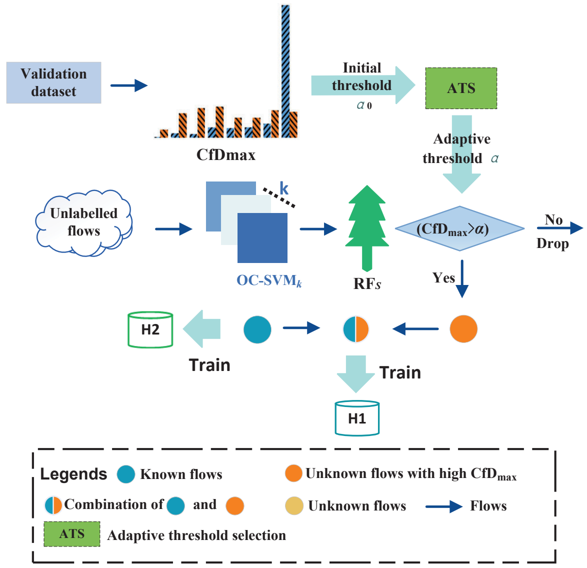


Fig. 2. Training phase of the CCS-UTD framework.

1) Traning phase: At the training phase, certain unlabelled samples are selected as the pseudo-negative samples of new class. These pseudo-negative samples are obtained by screening the unlabelled dataset for twice: 1) k single class support vector machines (One Class-SVM, OC-SVM) designed for k known classes are used to filter out the known classes; 2) an RF$_S$ (RF updated for S iterations) model is applied to filter the samples whose CfDmax are higher than an adaptive threshold α. The process of determining this threshold is illustrated in the upper half of Fig. 2. First, the CfDmax distribution is calculated using a validation dataset to obtain an initial threshold α_0. Subsequently, an adaptive threshold selection method is employed to get the best CfDmax threshold α for the current dataset. In the lower half of Fig. 2, H$_1$ is a binary classifier trained by the pseudo-negative samples of new classes with CfDmax values greater than α; the RF multi-classifier H$_2$ is trained by samples from the known classes.

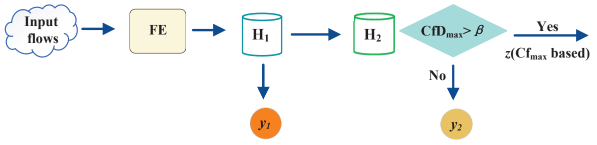


Fig. 3. Tesing phase of the CCS-UTD framework.

2) Testing phase: As shown in Fig. 3, the testing phase consists of a cascade structure: H$_1$ and H$_2$ that uses the threshold β to further distinguish between known and new class samples. After feature extraction (FE) of the input samples, H$_1$ is used to screen out unknown class samples (y_1) whose CfDmax is greater than threshold α. Then H$_2$ further refines the detection of new class by processing the remaning flows based on the threshold β. Meanwhile, the fine classification of known classes is completed by H$_2$ based on the maximum confidence level (Cfmax), denoted z.

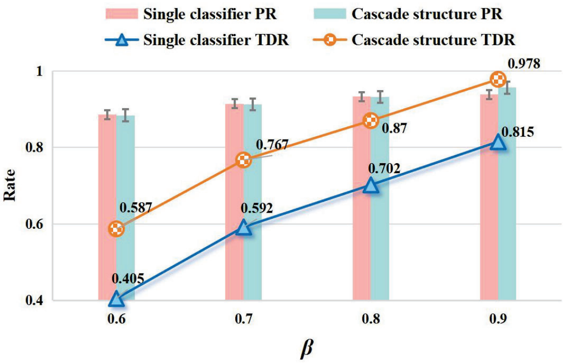


Fig. 4. *TDR* and *PR* of different model structures.

3) The cascade structure: The cascade structure model is a core part of this method. For complex open-set traffic classification problems, a single classifier is often ineffective to distinguish between new and known traffic types using a threshold.

For this reason, this paper, by following an idea of gradual refinement, designs a cascade structure composed of H$_1$ and H$_2$, as shown in Fig. 3. In the testing phase, H$_1$ is responsible for a first pass classification of input instances, mainly to screen out NewC samples that are difficult (closer to the boundary of a KnownC) to identify(y_1), and send the remaining samples to H$_2$. H$_2$ performs: 1) Further identification of NewC samples (y_2) by the threshold β; 2) A fine-grained classification of known classes. In this way, the proposed cascade structure(CasStru) can execute in a faster pipeline processing fashion. At the same time, the H$_1$'s identification capability trained by pseudo-negative samples with high CfDmax makes it possible to screen NewC samples that cannot be detected by thresholding, which in principle enhances the model's capability to detect unknown class samples.

To verify the effectiveness of CasStru, we conduct experiments on the MixD1 dataset (See Section IV-A for more details) by omitting H$_1$ and using only H$_2$ for classification based on the CfDmax threshold. Two performance evaluation metrics, Purity rate (PR) and True detection rate (TDR) [22], are used, as defined in Eqs.(1) and (2), where KP is the number of KnownC samples correctly identified, KN is the number of the KnownC samples which are misclassified as other known traffic, KU is the number of KnownC samples misclassified as unknown traffic, UP is the number of unknown traffic accurately detected, and UN is the number of NewC samples misclassified as known traffic.

Partial test results are shown in Fig. 4. It can be seen from the *TDR* curve that although a single classifier can still detect NewC samples, its True detection rate at each threshold value is significantly lower than that of CasStru, indicating that more unknown classes are undetected. *PR* represents the fine-grained classification results of KnownC samples. When the threshold β reaches 0.9, the performance of CasStru is also superior (More details can be found in following subsections). In a word, the introduction of the cascade structure is an effective choice, which has the advantages of both faster classification speed and higher rate of NewC detection.

$$PurityRate = \frac{KP}{KP + KN + KU} \tag{1}$$

$$TrueDetectionRate{=}\frac{UP}{UP + UN} \tag{2}$$

B. Data preprocessing

1) Feature extraction: To achieve fast online classification, the flow data is partitioned into 1-second flow segments. The first ten packets of each segment are used to compute the flow features which are then used for classification.

The collected data comprise of six distinct sequences, i.e., packet size, packet arrival time, timestamp, packet difference, uplink rate and downlink rate. Seventeen statistical features, as described in Table III, are computed for each sequence.

In order to improve the efficiency of NTC, the conditional frequency feature was introduced by Quan et al. [27], which is defined as the count of varying combinations for the (coded) sizes of two adjacent packets occurring in the upstream or downstream direction. There are 25 downlink and 4 uplink conditional frequency features. The total number of flow features is 131. As an example, the downlink conditional frequency $CF(i\ j)$ for a flow sample is computed as follows:

$$CF(i\backslash j) = \sum_{flowsample} c(P_1(i), P_2(j)), i, j \in \{1, 2, 3\} \tag{3}$$

where, $c(P_1(i), P_2(j))$ indicates an event that occurs once when the coded size of the former packet P_1 is i, and that of the subsequent packet P_2 is j.

TABLE III
STATISTICAL FEATURES OF FLOWS (PACKET SEQUENCES)

Serial number	Feature name	Serial number	Feature name
1	Average value	6	Minimum value
2	Standard deviation	7	Number of singular values
3	Kurtosis	8	Mode percentage
4	Skewness	9-17	Percentiles (from 10% to 90%)
5	Maximum value		

2) Feature selection: The online classification requires feature extraction (FE) to be as fast as possible, so feature selection (FS) and dimensionality reduction are performed as follows:

(1) The time complexity analysis of FE is performed to choose a subset of features whose complexity is no more than O(n) (n is the number of data packets).

(2) The Pearson correlation coefficient (PCC) is calculated for each feature with respect to the label and between the features. For a feature pair having a PCC greater than 0.9, the feature with lower correlation with the label is removed.

(3) The remaining features are ranked by RF. The optimal feature subset is obtained by adding features one by one according to the degree of importance and by observing the change of classification *Acc*, so as to find the inflection point of performance. In the experiments, around 20 features are finally selected by our method.

C. Initial threshold selection and confidence (difference) distributions

To determine the initial threshold α_0 required for the adaptive threshold selection algorithm, we begin by analyzing the CfDmax distributions of KnownC and NewC samples. The detailed process will be explained below.

To obtain the confidence that a given sample belongs to each known class, most current DL-based methods [25] use softmax activation in the last layer to get a probability vector. However, most edge devices have limited computing and storage resources, and may be unsuitable for using DL models that commonly require much more hardware resources. In this study, we use RF to analyze the distributions of classification confidence with different patterns. The confidence distributions of known classes (with samples of 8 classes randomly selected from MixD1) and unknown classes (with samples from other 8 classes of MixD1), as well as the CfDmax distributions, are shown in Fig. 5 and Fig. 6, respectively.

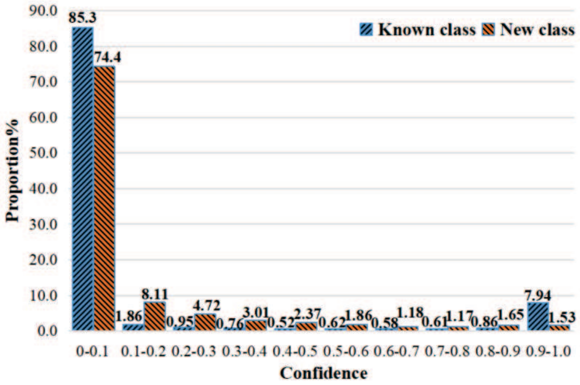


Fig. 5. Confidence distributions of known and new classes.

Fig.5 indicates that the known classes exhibit a higher percentage within the lowest confidence interval (0, 0.1] and the highest confidence interval (0.9, 1.0] compared to the new classes. From Fig. 6, it can be seen that the known

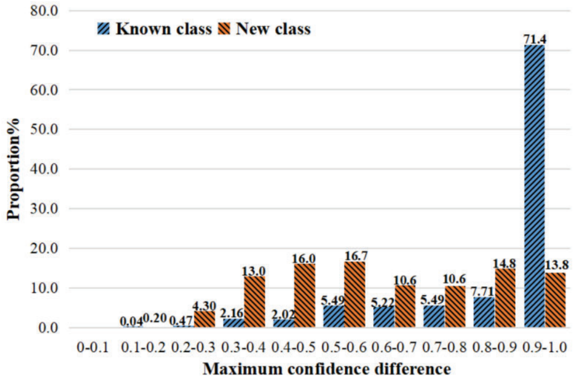


Fig. 6. CfDmax distributions of known and new classes.

classes have a significantly higher percentage of CfDmax within the range (0.9, 1.0] in comparison to the new classes. Therefore, CfDmax can be used to distinguish the KnownC from the NewC, and α_0 can be set to 0.9.

To validate the universality of this setup approach, we randomly select 5 combinations of known and new classes from MixD1 and MixD2 (See Section IV-A for more details) in Table IV. Then, for different combinations, we measure the proportions of known and new classes when CfDmax is between (0.9, 1.0]. As shown in Fig.7 and Fig. 8, the percentages for known classes are mostly higher than 80%, while those of new classes are mostly lower than 30%. In addition, the percentage of known classes is much higher than that of new classes in the range of (0.9, 1.0] of CfDmax, which demonstrates the right choice of initial threshold value 0.9.

In conclusion, for the determination of the initial threshold, the key is to identify a range where the number of KnownC samples significantly exceeds that of NewC samples, and use the lower bound of this range as the initial threshold. It is worth noting that the initial threshold does not need to be highly precise, only an approximate value is sufficient. Subsequent experiments (Section IV-D) demonstrate that our proposed adaptive threshold selection method is robust to the slight deviations of the initial threshold.

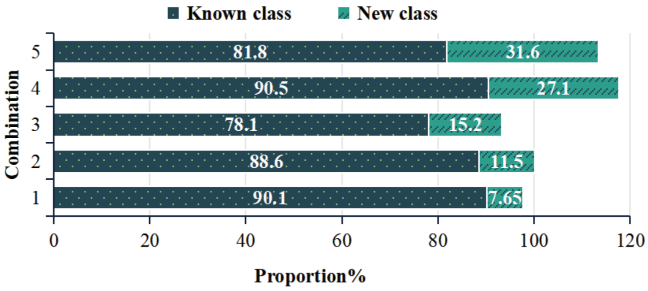


Fig. 7. Distributions of CfDmax between (0.9, 1.0] for different combinations of known and new classes on MixD1.

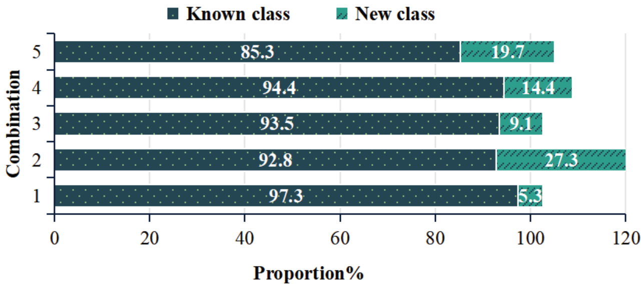


Fig. 8. Distributions of CfDmax between (0.9, 1.0] for different combinations of known and new classes on MixD2.

D. Adaptive threshold selection

Although the CfDmax distributions can effectively differentiate between new and known traffic types, applying a fixed CfDmax threshold to new datasets may lead to bias in the classification boundary. To ensure that CCS-UTD remains robust in identifying traffics in a new network environment, an Adaptive Threshold Selection (ATS) approach is proposed. ATS provides a general threshold selection method for different scenarios by using weighted operations on the mean and standard deviation of multiple predictions of the pseudo new classes samples.

Given the variations in the model's learning degree for samples and the inherent uncertainty of samples under different traffic characteristics, we design a multi-update scheme to retrain RF_0 (an initial RF model trained by KnownC samples) and introduce a prediction memory to store representative CfDmax values from each update. Specifically, in addition to unknown flows, there exists a certain number of known flows filtered by OC-SVM from unlabelled data, which can be used to update RF_0. Then, by calculating the mean and standard deviation of all representative CfDmax values in the memory, both the learning status and their fluctuations of the sample can be effectively reflected.

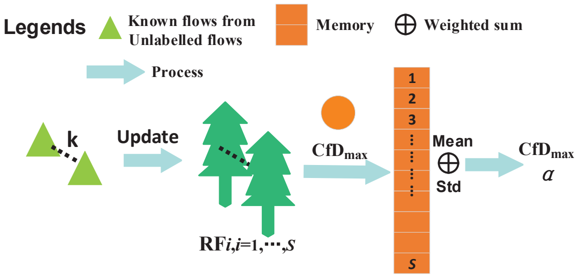


Fig. 9. Adaptive Threshold Selection

As shown in Fig. 9, ATS consists of three stages: (1) RF is updated S times using the filtered known classes samples, generating a series of RF models, denoted as RF_i, i=1, ..., S; (2) The pseudo-negative samples with high CfDmax are screened out from the unlabelled dataset by using the initial threshold α_0, and then perform S predictions on these pseudo-negative samples with the RF_i models to recalculate

TABLE IV
DIFFERENT COMBINATIONS OF KNOWN AND NEW CLASSES ON MIXD1 AND MIXD2

Dataset		MixD1		MixD2	
		Known classes*	New classes	Known classes**	New classes
Combination	1	2, 14, 16	5, 6, 7, 12, 13	24, 26, 31, 41, 42, 45	21, 23, 28, 32, 35, 44, 48, 49
	2	6, 8, 14, 16	3, 7, 10, 11, 13	23, 25, 28, 36, 43, 44, 47	29, 30, 32, 34, 35, 41, 45, 48, 50
	3	3, 8, 9, 16	4, 7, 10, 11, 12	21, 25, 27, 28, 32, 45, 48, 50	23, 26, 30, 33, 34, 41, 43, 44, 49
	4	4, 5, 7, 14, 16	3, 6, 10, 11, 12	21, 28, 29, 30, 32, 45, 46, 48	22, 23, 26, 27, 31, 42, 44, 46, 49
	5	2, 4, 7, 12, 14	6, 9, 11, 13, 15	28, 29, 32, 34, 36, 47, 50	22, 23, 27, 31, 35, 41, 43, 45, 48

*Correspondence between digital labels and data categories: 1-BitTorrent, 2-Email, 3-Facebook_audio, 4-Facebook_chat, 5-Ftp, 6-Hangouts_audio, 7-Skype_audio, 8-Skype_file, 9-Skype_ video, 10-Youtube, 11-douyu_480p, 12-huya_480p, 13-tencent_480p, 14-tencent_720p, 15-youku_720p, 16-douyu_1080p
**: 21-Cridex, 22-Geodo, 23-Htbot, 24-Miuref, 25-Neris, 26-Nsis_ay, 27-Shifu, 28-Tinba, 29-Virut, 30-Zeus, 31-Facebook_audio, 32-Hangouts_audio, 33-SFTP, 34-Skype_audio, 35-Spotify, 36-Vimeo41- Distance,42-Flame_Sensor, 43-Heart_Rate, 44-IR_Receiver, 45-Modbus, 46-phValue, 47-Soil_Moisture, 48-Sound_Sensor, 49-Temperature, 50-Water_Level

their CfDmax values. The 10th percentile of the predicted CfDmax values on these pseudo-negative samples after each update is stored in memory; (3) Using the stored values of mean and standard deviation, the adaptive threshold α that best fits the current dataset is determined by weighted calculation. The implementation details of the ATS method are shown in Algorithm 1.

E. Selection of pseudo NewC samples from unlabelled datasets

From the above analysis (Fig. 5 - Fig. 8), it can be seen that in the testing stage, H_2 can detect new classes with the threshold β (=0.9). However, as shown in Fig. 8, there are still a fraction of NewC instances with CfDmax exceeding 0.9, resulting in misclassification.

As mentioned above, H_1 is added into the cascade structure that is trained by the pseudo unknown class samples selected from the unlabelled datasets in combination with the KnownC samples. Since gathering unlabelled flow data is relatively easy from the real networks [28], this study attempts to choose negative samples from the unlabelled data to train H_1, for the purpose to obtain negative samples with CfDmax values greater than the threshold α. The specific process is shown in Algorithm 2.

During the process of selecting adaptive thresholds, while our primary focus is on setting the parameter α, for the threshold β used in the testing phase, we recommend $\beta = \alpha$. Subsequent experiments (Section IV-D) will verify this. Note that the thresholds α and β have distinct uses, though they may have the same numerical value. In the screening of unlabelled samples, any sample whose CfDmax is higher than the value of α is classified as a negative sample, and these samples are similar to but not belong to the KnownC. On the other hand, during the testing phase, if the CfDmax of the input sample is higher than β, it is recognized (by H_2) as belonging to the known classes.

F. Cascade classification

In the testing stage, H_2 trained on the KnownC dataset detects the NewC instances by CfDmax thresholding, and classifies the known classes using the Cfmax information. Algorithm 3 describes the cascade classification process. Note that U_{i1} and U_{i-1} (for $i = 1, 2, \ldots, k$) are the positive and negative classes derived from the OC-SVM$_i$

Algorithm 1 Adaptive Threshold Selection

```
Require: Size of memory $S$, RF$_0$ trained with known class
    samples, prediction memory, initial threshold $\alpha_0$
Input: Unlabelled flow dataset $U$
Output: Adaptive threshold $\alpha$
Obtain known flows $M_2$ by Algorithm 2
Obtain subset $M$ of unlabelled data with unknown
    classes whose CfDmax $> \alpha_0$ by Algorithm 2
# Update RF$_0$ to recalculate CfDmax for $M$
Divide $M_2$ into $S$ subsets:
        $M_2 = \{Sub_1, Sub_2, \ldots, Sub_S\}$
for $i = 1$ to $S$ do
    Define Training set $Set_i = Sub_1 \cup Sub_2 \cup \cdots \cup Sub_i$
    Retrain RF$_i$ using RF$_{i-1}$ as the initial model and
    Training set $Set_i$
    Obtain and store updated model RF$_i$
end for
Initialize a temporary array: temp
for $j = 1$ to $S$ do
    for each $m \in M$ do
        Compute CfDmax$-m \leftarrow$ calculation using RF$_j$
        on $m$
        Store CfDmax$-m$ in temp
    end for
    Calculate the 10th percentile of temp and store it in
    memory
end for
# Calculate mean and standard deviation in memory to
obtain adaptive threshold $\alpha$
Calculate the mean in memory: mean
Calculate the standard deviation in memory: std
Adaptive threshold $\alpha = \text{mean} - \text{std}$
Return $\alpha$
```

Algorithm 2 Selection for pseudo unknown class samples from unlabelled flow datasets

```
Require: KnownC$_k$ $(z_1, z_2, \ldots, z_k)$, CfDmax adaptive
    threshold $\alpha$, RF$_0$ trained with known class samples
Input: Unlabelled flow dataset $U$
Output: Subset $M$ of unlabelled data with unknown classes
    whose CfDmax $> \alpha$, subset $M_2$ of unlabelled data with
    known classes
for $i = 1$ to $k$ do
    Train with KnownC $z_i$ to obtain OC-SVM$_i$
end for
for $i = 1$ to $k$ do
    Classify $U$ with OC-SVM$_i$ and obtain $U_{i1}$ and $U_{i-1}$
end for
$M_1 = U_{11} \cup U_{21} \cup U_{31} \cup \cdots \cup U_{k1}$
for each $m \in M_1$ do
    if $m$ only in $U_{11} \vee U_{21} \vee U_{31} \vee \cdots \vee U_{k1}$ then
        $m \in M_2$
    else
        $m \in M_3$
    end if
end for
$M_4 = U_{1-1} \cap U_{2-1} \cap U_{3-1} \cap \cdots \cap U_{k-1}$
$M_5 = M_3 \cup M_4$
RF output confidence set $m_t = \{t_1, t_2, \ldots, t_k\}$ for each
    sample $m$ in $M_5$
Calculate the difference between maximum confidence
    and minimum confidence in $m_t$: $\alpha_m = t_{\max} - t_{\min}$
for each $m \in M_5$ do
    if $\alpha_m > \alpha$ then
        $m \in M$
    else
        Discard $m$
    end if
end for
Return $M$, $M_2$
```

classification, respectively.

IV. Experiments

A. Datasets

TABLE V
ISCX PARTIAL DATA

Categories	Applications	#samples
File transfer	BitTorrent, Ftp, Skype	1000
Voice calls	Facebook, Hangouts, Skype	1000
Mail	Email	1000
Chat	Facebook	1000
Video	Skype, Youtube	1000

Comprehensive experiments are carried out on five real network datasets: ISCX non-VPN (ISCX) [29], VideD video dataset, ISCX-Tor [30], USTC-TFC [31] malware dataset and Edge-IIoTset [32] network security dataset. Data in the VideD dataset was collected in the campus network of Nanjing University of Posts and Telecommunications during 2019-2020 using the Wireshark [33] software. Table V ∼ Table IX show the specific information of each dataset. To verify the generalizability of the proposed method, the ISCX and VideD datasets are combined to form the hybrid dataset 1 (MixD1). Similarly, a hybrid dataset 2 (MixD2) composed of Edge-IIoTset, ISCX-Tor, VideD and USTC-TFC datasets, containing 36 traffic classes, is used to simulate larger traffic loads.

Algorithm 3 Cascade classification algorithm

```
Require: Mixed flow dataset $X$; CfDmax threshold $\beta$
Output: KnownC label $z$ $(z_1, z_2, \ldots, z_k)$; NewC label $y$
FE is applied to the dataset $X$ to obtain the input sample
    $x$
H$_1$ classifies $x$ to obtain $y_1$ and $x_0$, where $y_1 \in y$
H$_2$ outputs the confidence set $lt = \{t_1, t_2, \ldots, t_k\}$ for
    each sample $l$ in $x_0$
Calculate the difference between maximum confidence
    and minimum confidence in $lt$: $\beta_l = t_{\max} - t_{\min}$
for $l$ in $x_0$ do
    if $\beta_l > \beta$ then
        $l \in z$
        Find the subscript index $j$ of the maximum
    confidence in the confidence set $lt$ of $l$
        if $j = 1$ then
            $l \in z_1$
        else if $j = 2$ then
            $l \in z_2$
        else if $j = k$ then
            $l \in z_k$
        end if
    else
        $l \in y_2$
    end if
end for
$y = y_1 \cup y_2$
```

TABLE VI
VIDED DATASET

Categories	Applications	#samples
Video live	douyu_480p, huya_480p, douyu_1080p	1500
Video on demand	tencent_480p, tencent_720p, youku_720p	1500

B. Evaluation indexes

The proposed method is evaluated in terms of classification accuracy and time efficiency. For the classification accuracy, four metrics are used, which are the normalized accuracy (*NA*) of the open set recognition [7] (which weights the accuracy for known classes (AKS) and the accuracy

TABLE VII
ISCX-Tor partial data

Category	Applications	#samples
File transfer	SFTP	1000
Voice calls	Facebook, Hangouts, Skype	1000
Music	Spotify	1000
Video	Vimeo	1000

TABLE VIII
USTC-TFC malware dataset

Categories	Applications	#samples
Malware	Cridex, Geodo, Htbot, Miuref, Neris, Nsis_ay, Shifu, Tinba, Virut, Zeus	1000

for new classes (AUS)), the precision (P), recall (R) and F_1 score (F_1), where P represents the proportion of correctly predicted positive examples; R is the proportion of positive samples that are correctly identified; F_1 score is the harmonic mean of P and R. The specific computations are shown in Eq.(4) - Eq.(9), where, for KnownC i, TP_i, TN_i, FP_i, and FN_i represent the numbers of positive and negative samples correctly classified, and the numbers of positive and negative samples incorrectly classified, respectively; TU and FU refer to the numbers of samples of new classes correctly and incorrectly identified, respectively; λ is the regularization coefficient, with 0 <λ<1 (it is set to 0.5 in the experiments). The evaluation of time performance includes the training and inference time.

$$NA = \lambda AKS + (1 - \lambda)AUS \tag{4}$$

$$AKS = \frac{\sum_{i=1}^{k}(TP_i + TN_i)}{\sum_{i=1}^{k}(TP_i + TN_i + FP_i + FN_i)} \tag{5}$$

$$AUS = \frac{TU}{TU + FU} \tag{6}$$

$$P = \frac{TP}{TP + FP} \tag{7}$$

$$R = \frac{TP}{TP + FN} \tag{8}$$

$$F1 = \frac{2 * P * R}{P + R} \tag{9}$$

C. Experimental environment

The experiments are performed on a Dell Vostro 14-5480 laptop with Windows 10 operating system, Intel i5-5200U CPU@2.20GHz CPU, and 8GB RAM. The 5-fold cross-validation is used in the experiments, with 80% samples randomly selected from the total samples as the training set, and the remaining 20% as the test set. OC-SVM is implemented using LIBSVM [34]; the number of trees in RF is set to 100, and min_samples_leaf is 1.

TABLE IX
Edge-IIoTset DATASET

Categories	Applications	#samples
Network security	Distance, Flame_Sensor, Heart_Rate, IR_Receiver, Modbus, phValue, Soil_Moisture, Sound_Sensor, Temperature, Water_Level	1200

D. *Effects of different initial threshold values*

In the ATS algorithm, the initial threshold is an important factor. To better evaluate the performance of the model with varying initial thresholds, several KnownC and NewC combinations are randomly selected from both MixD1 and MixD2. Specifically, Combination 1 includes 6 known classes and 6 new classes, while Combination 2 consists of 6 known classes and 5 new classes. Table X and Table XI list the results obtained from different data combinations and initial thresholds.

TABLE X
Classification results using different initial thresholds on MixD1

α_0		0.7	0.8	0.85	0.9	0.95
α/β		0.816	0.863	0.895	0.925	0.943
KnownC	F_1	0.898	0.905	0.919	**0.9393**	0.923
	P	0.895	0.920	0.978	0.9816	**0.985**
	R	**0.925**	0.913	0.909	0.893	0.878
NewC	F_1	0.856	0.895	0.911	**0.9382**	0.897
	P	0.898	0.903	**0.912**	0.901	0.883
	R	0.793	0.894	0.922	**0.979**	0.969
NA		0.932	0.941	0.971	**0.986**	0.966

TABLE XI
Classification results using different initial thresholds on MixD2

α_0		0.7	0.8	0.85	0.9	0.95
α/β		0.798	0.848	0.869	0.882	0.918
KnownC	F_1	0.786	0.809	0.849	**0.854**	0.827
	P	0.827	0.859	0.876	0.898	**0.902**
	R	**0.889**	0.871	0.853	0.826	0.81
NewC	F_1	0.808	0.819	0.825	**0.832**	0.817
	P	0.823	0.815	**0.826**	0.804	0.798
	R	0.771	0.808	0.813	**0.865**	0.858
NA		0.856	0.897	0.903	**0.919**	0.898

As mentioned in Section III-C, the optimal initial threshold for MixD1 and MixD2 is 0.9. From Table X and Table XI, it can be observed that when the initial threshold α_0 deviates from 0.9, the adaptive threshold α converges toward the optimal threshold under the guidance of the ATS algorithm and thus the decrease in NA does not exceed 6%, which demonstrates the robustness of ATS to slight deviations.

Regarding the impact of α_0 on the classification performance of known and new classes of CCS-UTD, increasing α_0 improves the precision of KnownC but results in a decline in recall. In contrast, for the new classes, increase of α_0 will generally tend to increase recall but decrease precision. This is because the CfDmax of the KnownC is mainly distributed between (0.9, 1.0], while the CfDmax of the NewC is relatively evenly distributed in all intervals. When the threshold is increased to 0.9, fewer KnownC samples will be misclassified into new classes, and more new classes will be detected.

For the threshold β in the testing phase, it is suggested to use the same value as α. Specifically, the binary classifier H_1 is used to detect the NewC samples with CfDmax greater than β. The NewC samples with $\beta <$CfDmax $<\alpha$ cannot be detected during the classification process if $\alpha >\beta$. Conversely, if $\alpha <\beta$, H_1 may misclassify more KnownC samples into new classes. Table XII and Table XIII provide a comparison of the classification performance for different values of α and β. According to the comparisons, if $\alpha >\beta$, the recall of NewC decreases, while when $\alpha <\beta$, the recall of KnownC drops. The overall classification accuracy is optimal when $\alpha = \beta$.

TABLE XII
CLASSIFICATION EFFECTS OF INCONSISTENT THRESHOLDS ON MIXD1

α/β		0.9/0.8	0.8/0.9	0.9/0.9
KnownC	F_1	0.9188	0.9098	**0.9313**
	P	0.9016	**0.9786**	**0.9786**
	R	**0.9367**	0.8500	0.8883
NewC	F_1	0.9000	0.9193	**0.9362**
	P	**0.9321**	0.8671	0.8976
	R	0.8700	**0.9783**	**0.9783**
NA		0.9270	0.9820	**0.9836**

TABLE XIII
CLASSIFICATION EFFECTS OF INCONSISTENT THRESHOLDS ON MIXD2

α/β		0.9/0.8	0.8/0.9	0.9/0.9
KnownC	F_1	0.8455	0.8418	**0.8483**
	P	0.8201	**0.8933**	0.8916
	R	**0.8453**	0.8136	0.8130
NewC	F_1	0.7915	0.8036	**0.8297**
	P	**0.8191**	0.7905	0.8001
	R	0.7473	0.8576	**0.8632**
NA		0.8685	0.9066	**0.9137**

E. Comparison of different methods

Fig.10 and Fig.11 present the comparisons of the classification results of the proposed method CCS-UTD with three other methods of CD-OSFR [10], ASG-SVM [12] and k-LND [26] on MixD1 and MixD2. Table XIV illustrates the time performances of these methods.

It can be seen from Fig.10 and Fig. 11 that on different datasets, the F_1 score and *NA* indexes for KnownC and NewC obtained by CCS-UTD are significantly better than those obtained by other methods. According to Table XIV, CCS-UTD also has the shortest training and inference time.

Note that the k-LND method [26] defined the logit layer output as the class center and named it Mean Activation Vector (MAV). They assumed that a sample from a known class would be distant from MAV of the neighbors of its class, in addition to being closer to its own MAV, which improves the closed-set and open-set classification accuracies. However, as can be seen from the Fig.10 and Fig. 11, although the F_1 score of k-LND is close to that of CCS-UTD, its P is particularly low, that is, its false positive rate is high, which indicates that its classification boundary is too loose. In contrast, CCS-UTD not only maintains the highest F_1 and *NA* scores, but also improves the P of known classes by nearly 20% compared to k-LND. This shows that our method successfully separates most of NewC samples that are similar to KnownC samples into a new classification space, which effectively avoids misclassification of known and unknown classes.

Compared to CD-OSR, by using CCS-UTD, the F_1 score is improved by around 8-10% and 8-9% for KnownC and NewC, respectively, and *NA* index is improved by 10-11%. This improvement may be attributed to the fact that CD-OSR does not fully utilize the label information of the KnownC. In other words, CD-OSR relies on the HDP automatic clustering, and does not strictly classify samples into corresponding classes based on their labels. Thus, the samples of different labels may be clustered into the same class, or the samples having the same label clustered into different classes during the training process. In contrast, CCS-UTD takes advantage of the label information explicitly, and applies it to the classification pipeline. Concerning the time performance, CD-OSR needs longer training and inference time as it is bound by HDP's inherent complexity, thus resulting in higher computational cost.

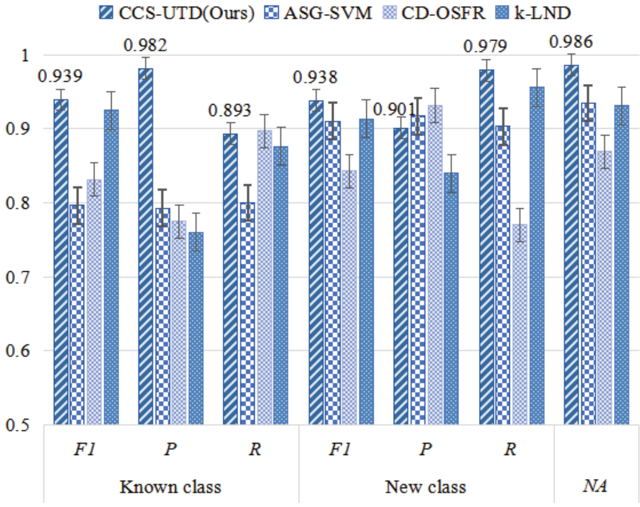


Fig. 10. Comparison of classification performances of different methods on MixD1.

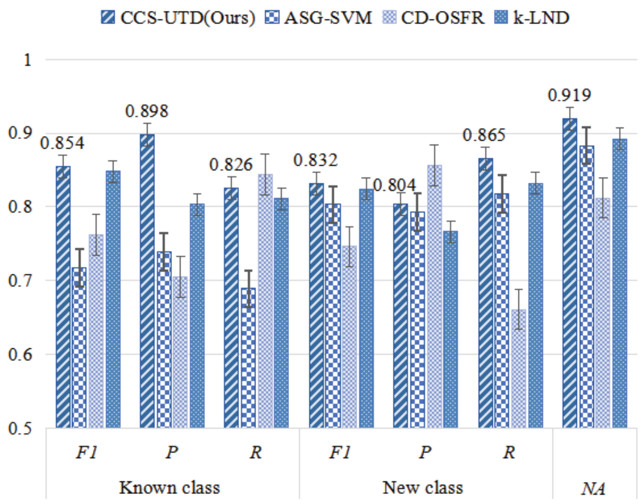


Fig. 11. Comparison of classification performances of different methods on MixD2.

Compared with ASG-SVM, CCS-UTD demonstrates an improvement of approximately 13-14% in F_1 score for KnownC, 3% in F_1 score for NewC, and 2-5% in *NA* index. Although ASG-SVM outperforms CCS-UTD in detecting new classes than in sub-classifying known classes, it generates negative samples akin to each KnownC, leading to lower recall rates for the known classes. In addition, ASG-SVM trains one SVM classifier for each KnownC using both positive and negative samples, thus requiring training multiple SVMs, leading to longer training time. In the testing stage, only when all trained SVMs recognize an instance as a negative one, will ASG-SVM classify it as a NewC sample, resulting in longer inference time.

F. Real-time applicability

Most DL-based NTC models contain tens of thousands of parameters, resulting in model sizes ranging from a few MB to several hundred MB, and normally need dedicated computational hardware such as GPU. However, many edge devices commonly have limited storage of only a few MB, and limited computation power, which makes it challenging to accommodate medium to large-scale models [16]. Our approach with an RF model only requires a memory of a few hundred KB and common CPU, that is affordable for most network edge devices. Moreover, the packet forwarding rate of common edge routers is usually around a few million packets per second (Mpps) [35]. As shown in Table XIV, for the training time, CCS-UTD is much shorter than other methods; for the inference time, our method can process a flow segment of 10 packets in around 0.04 ms on average, achieving a throughput of 0.25 Mpps, which roughly aligns with the computational capabilities of common edge routers. With its notably low classification latency, the proposed CCS-UTD demonstrates significant potential for deployment in real-world network traffic classification systems. In comparison, k-LND reduces model parameters through model quantization. However, its packet processing time is still twice as long as that of our method, which implies higher hardware requirements for the device. Meanwhile, ASG-SVM and CD-OSR require devices with even higher computational resources, making them unsuitable for deployment on resource-limited edge devices.

TABLE XIV
COMPARISON OF TIME PERFORMANCES OF DIFFERENT METHODS ON 2 MIXED DATASETS (AVE. MS/SAMPLE)

Method	Training Time	Inference Time
CCS-UTD	**0.0788**	**0.0415**
ASG-SVM	34.8650	2.0129
CD-OSR	7.4768	0.3887
k-LND	0.913	0.0806

G. Sensitivity analysis

Table XV explores the impact of the length *S* of prediction memory on NewC detection. Overall, the adaptive threshold method exhibits similar *NA* on the MixD2 dataset across different memory lengths, indicating a low sensitivity to changes in *S*. Therefore, when handling new data distributions, adjusting this parameter does not need to be prioritized. As shown in Table XV, CCS-UTD can achieve the best performance when *S*=15. Generally, while a longer memory length provides a more comprehensive assessment of confidence information, an excessively long memory may fail to accurately reflect the current status of samples and causes additional storage costs. On the contrary, if the memory length is too short, it is difficult to effectively capture fluctuations in CfDmax. Therefore, setting *S*=15 allows ATS to enhance flexibility and efficiency while being more adaptable to high traffic loads.

Next, we present the experimental CfDmax values and corresponding *NA* indexes in intervals of different percentiles based on the CfDmax distributions (in the range of its min. value of 0.665 and max. value of 1). As show in Table XVI, at the 1st percentile, CfDmax is significantly lower than in other intervals, with a comparatively low *NA*. In contrast, *NA* reaches its peak at the 10th percentile, while the changes in CfDmax thereafter are relatively small. Therefore, we select

TABLE XV
NA FOR DIFFERENT MEMORY LENGTHS *S*

S	2	4	10	**15**	20	50	100
NA	0.914	0.913	0.905	**0.919**	0.909	0.913	0.914

TABLE XVI
NA FOR DIFFERENT PERCENTILES

Percentile	1%	**10%**	20%	40%	60%
CfDmax	0.74	**0.88**	0.96	0.99	1
NA	0.818	**0.919**	0.891	0.88	0.88

TABLE XVII
NA UNDER DIFFERENT η IN ADAPTIVE THRESHOLD

mean + η*std	η	-5	-2	**-1**	0	1	2
	NA	0.915	0.914	**0.919**	0.914	0.916	0.912

the 10th percentile of each update as the representative value and store it in memory.

The proposed ATS determines the adaptive threshold α through linear combination operations, where the key factor is the weight η of the standard deviation. It can be seen from the Table XVII that *NA* is the highest when η is -1. This is because this setting effectively reduces the uncertainty in predictions while measuring the learning status of samples after multiple updates, thereby enhancing the stability of ATS. Further experiments with varying levels of noise in unlabelled dataset (Section IV-I) demonstrate that this approach also improves the robustness of CCS-UTD to noises in the dataset.

H. Ablation experiments

1) *Algorithm module:* In order to further evaluate the proposed model, ablation experiments are conducted on three additional datasets. Here we mainly focus on the comparisons of the micro-F-measure used in CD-OSFR, where the *FN* and *FP* also consider the false unknown classes and false known classes.

First, for CfDmax, we remove H_1 trained by the screened samples and only use H_2 and maximum confidence as thresholds for the identification of KnownC and NewC samples. Then, to demonstrate the robustness of adaptive threshold in different scenarios, we select three datasets LETTER [36], USPS [37] and PENDIGITS [38] used in CD-OSFR, and MixD2 dataset containing the most traffic classes. At the same time, a fixed threshold version of CCS-UTD, called CCS-UTD-Fix, is also used.

H_2-alone, CD-OSFR and CCS-UTD-Fix are used as the baselines for comparison. As shown in Fig.12, CCS-UTD obtains the best F_1 score on almost every dataset, while H_2-alone performs the worst on every dataset. This is because using only the maximum confidence as a threshold can only distinguish some obvious unknown classes and confuse other unknown classes with known classes. Training H_1 with unlabelled samples selected by CfDmax can enhance the model's discriminative ability and allocate better classification regions for NewC.

Compared to CCS-UTD-Fix, CCS-UTD using ATS can find the optimal threshold in each dataset, thus achieving the best performance. Especially on dataset PENDIGITS, using a fixed threshold of 0.9 is not fully applicable, resulting in inferior performance to CD-OSFR.

2) *Model selection:* In terms of model selection, we evaluate the fine-grained classification F_1 scores (excluding new classes), new class detection performance (*NA*), and training time of KNN [39], XGBoost [40], and RF on two mixed datasets. As shown in Table XVIII, in fine-grained classification (Fine-F_1), RF and XGBoost have similar performance, while KNN performs worse. For new class detection, RF has a higher *NA* index than the other models, demonstrating better new class detection capability. In terms of training time, due to the fact that KNN's training process only requires data storage, its training time is the shortest, but its testing time is longer. In contrast, XGBoost has the longest training time.

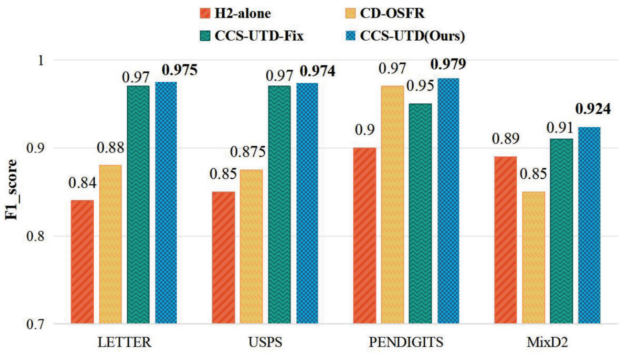


Fig. 12. Ablation experimental results on different datasets.

Further, we evaluate the performance of two deep learning models, AutoEncoder and CNN1D, on two mixed datasets. Specifically, we follow the settings of Deep Packet [41] to convert raw data packets into byte vectors as features. As shown in Table XVIII, deep learning methods do not show significant performance advantages over RF and XGBoost and require longer training time. So, we select the RF model for this paper.

I. Selection of pseudo-negative samples on noisy datasets

In real-world network environments, noise is unavoidable when gathering unlabelled data. To evaluate the effect of noise on NewC detection performance, we add different levels of Gaussian noise to two mixed real-world datasets and compares *NA* performance under both adaptive and fixed thresholds. Specifically, the added noise level represents the standard deviation of Gaussian noise, which is proportional to the feature value of each sample.

As shown in Table XIX, after multiple rounds of updates, the ATS scheme demonstrates better noise robustness across all datasets compared to the fixed threshold. With the increase of noise level, the performance degradation rate of ATS is significantly lower than that of the fixed threshold. This is attributed to the linear combination of mean and standard deviation in the ATS algorithm, which comprehensively considers the learning status and stability of the samples after each update. Specifically, noise affects the fluctuations of model predictions for samples, and standard deviation is one of the key measure of this fluctuation. Intuitively, if the predicted CfDmax value for a sample remain stable over multiple updates, it indicates that the model's knowledge of the sample is consistent, thus making the prediction more reliable.

TABLE XVIII
PERFORMANCE COMPARISON OF DIFFERENT MODELS ON TWO DATASETS

Dataset	MixD1			MixD2		
Models	Fine-F_1	*NA*	Training Time(ms)	Fine- F_1	*NA*	Training Time(ms)
RF	0.94	**0.986**	0.0767	**0.94**	**0.919**	0.0893
XGBoost	**0.95**	0.935	0.23	**0.94**	0.88	0.305
KNN	0.81	0.94	**0.009**	0.77	0.874	**0.016**
AutoEncoder	0.84	0.851	1.31	0.66	0.801	1.8
CNN1D	0.8	0.842	1.4	0.56	0.788	1.54

TABLE XIX
COMPARISON OF *NA* BETWEEN ADAPTIVE THRESHOLD AND FIXED THRESHOLD AT DIFFERENT NOISE LEVELS

Method	Adaptive Threshold		Fixed Threshold	
Dataset Noise level	MixD1	MixD2	MixD1	MixD2
10%	0.972	0.918	0.966	0.901
30%	0.96	0.907	0.948	0.884
50%	0.948	0.901	0.933	0.869
70%	0.94	0.892	0.921	0.841

TABLE XX
COMPARISON OF CLASSIFICATION PERFORMANCE USING DIFFERENT NEGATIVE SAMPLES ON MIXD1 AND MIXD2

Datasets		MixD1		MixD2	
Method		UL*	AL	UL	AL
KnownC	F_1	**0.9393**	0.848	**0.854**	0.786
	P	0.9816	**0.985**	0.898	**0.909**
	R	**0.893**	0.751	**0.826**	0.692
NewC	F_1	**0.9382**	0.882	**0.832**	0.803
	P	**0.901**	0.799	**0.804**	0.712
	R	0.979	**0.99**	0.865	**0.908**
NA		0.986	**0.987**	**0.919**	0.914

* UL: unlabelled data, AL: adversarial learning.

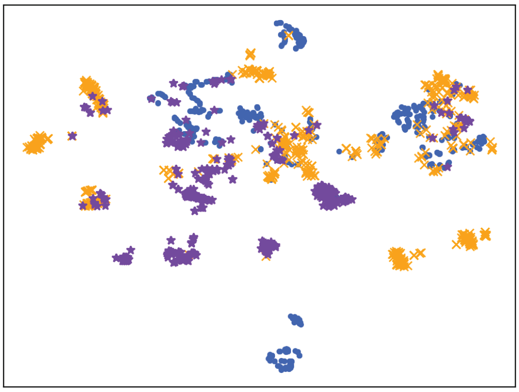


Fig. 13. Visualization of known classes (blue circle), new classes (orange cross), and negative samples filtered from unlabelled data (purple pentagon) on MixD1.

As for the generation of pseudo-negative samples, ASG-SVM generates negative samples through adversarial learning (AL), while CCS-UTD chooses negative samples from the unlabelled data (UL). Table XX compares the two pseudo-negative sample generation methods by presenting the classification results of H_1 trained with various negative samples on both datasets.

As can be seen from Table XX, our selection method is better than the adversarial generation method; it can improve the F_1 score by around 8-9% for the KnownC and by about 5% for the NewC, only at the cost of a slight decrease of *NA* index. This is because the negative samples generated by ASG-SVM are the surrounding boundary data of KnownC, which are easily confused with the KnownC, while the distribution of the negative samples chosen by our method is slightly more random.

Taking MixD1 as an example, a graphical illustration of different pseudo-negative samples, known classes and new classes using TSNE [42] is given in Figs 13 and 14. From the plots, it can be seen that there is no significant correlation between the KnownC and the distribution of negative samples by our method; while with the adversarial method, there is a greater degree of entanglement between the KnownC and the generated negative samples.

The different classification outcomes can be explained by the different distributional relationships between the pseudo-negative samples and the known classes generated by the two methods. The negative samples generated with the adversarial method enhance the ability of the method to detect new classes, but also increase the likelihood of misclassifying the known classes as new classes, leading to higher recall for new classes and higher precision for known classes. However, this method has lower precision for new classes and recall for known classes compared to our method. To sum up, by utilizing the pseudo-negative samples filtered by our method, it can enhance the ability of our method to sub-classify known classes, identify new classes, and greatly reduce the computational cost.

V. CONCLUSIONS

In order to improve the performance of open set flow recognition, this paper proposes a NewC detection method, called CCS-UTD, based on confidence (difference) and a cascade structure. The associated algorithms are imple-

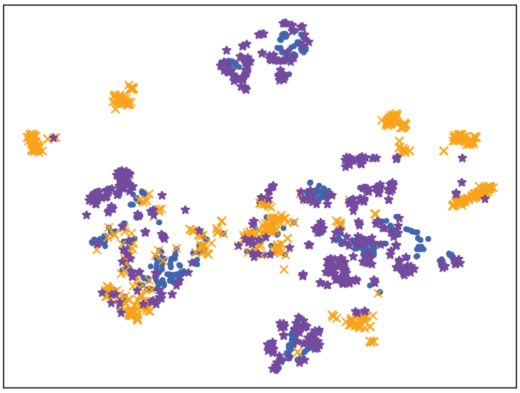


Fig. 14. Visualization of known classes (blue circle), new classes (orange cross), and adversarial-based generation of negative samples (purple pentagon) on MixD1.

mented by analyzing the confidence distributions of the known and new classes, and an algorithm is designed to filter out the pseudo-negative samples from the unlabelled dataset. The NewC instances exceeding a threshold are first detected by a binary classifier. The remaining data is then distinguished using CfDmax to separate the NewC samples from the KnownC samples, which are further sub-classified using Cfmax for known classes. The proposed method is evaluated on two hybrid datasets consisting of five real network datasets, reaching an overall accuracy higher than 90%. Compared with the state-of-the-art methods, the F_1 and NA scores of known and new classes are significantly improved by using our method, and the training and inference time are greatly reduced.

However, the proposed method has some limitations: When screening the negative samples, the randomness of unlabelled data and the threshold condition may restrict the number of obtained negative samples, thus lowering the utilization rate of the dataset. Our future work will consider fast updating of the model and investigating scenarios where the instances of different new classes appear in the flow traffic data.

References

[1] Q. Ma, W. Huang, Y. Jin, and J. Mao, "Encrypted traffic classification based on traffic reconstruction," in *2021 4th International Conference on Artificial Intelligence and Big Data (ICAIBD)*, 2021, pp. 572–576.

[2] B. Pang, Y. Fu, S. Ren, and Y. Jia, "High-performance network traffic classification based on graph neural network," in *2023 IEEE 6th Information Technology,Networking,Electronic and Automation Control Conference (ITNEC)*, vol. 6, 2023, pp. 800–804.

[3] N. Bhatla and M. Malik, "Network traffic classification techniques: A review," vol. 984 LNEE, 2023, pp. 371 – 388.

[4] Y. Yang, Y. Yan, Z. Gao, L. Rui, R. Lyu, B. Gao, and P. Yu, "A network traffic classification method based on dual-mode feature extraction and hybrid neural networks," *IEEE Transactions on Network and Service Management*, vol. 20, no. 4, pp. 4073–4084, 2023.

[5] S. Fathi-Kazerooni and R. Rojas-Cessa, "Countering machine-learning classification of applications by equalizing network traffic statistics," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 4, pp. 3392–3403, 2021.

[6] X. Mu, K. M. Ting, and Z.-H. Zhou, "Classification under streaming emerging new classes: A solution using completely-random trees," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 8, pp. 1605–1618, 2017.

[7] C. Geng, S.-J. Huang, and S. Chen, "Recent advances in open set recognition: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3614–3631, 2021.

[8] S. F. A. Zaidi and C.-G. Lee, "One-class classification based bug triage system to assign a newly added developer," in *2021 International Conference on Information Networking (ICOIN)*, 2021, pp. 738–741.

[9] Z. Yang, J. Long, Y. Zi, S. Zhang, and C. Li, "Incremental novelty identification from initially one-class learning to unknown abnormality classification," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 7, pp. 7394–7404, 2022.

[10] C. Geng and S. Chen, "Collective decision for open set recognition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 192–204, 2022.

[11] A. R. Lubis, S. Prayudani, Y. Fatmi, and O. Nugroho, "Latent semantic indexing (lsi) and hierarchical dirichlet process (hdp) models on news data," in *2022 5th International Conference of Computer and Informatics Engineering (IC2IE)*, 2022, pp. 314–319.

[12] Y. Yu, W.-Y. Qu, N. Li, and Z. Guo, "Open-category classification by adversarial sample generation," *arXiv preprint arXiv:1705.08722*, 2017.

[13] J. Kwon, D. Jung, and H. Park, "Traffic data classification using machine learning algorithms in sdn networks," in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, 2020, pp. 1031–1033.

[14] L. Yang, A. Finamore, F. Jun, and D. Rossi, "Deep learning and zero-day traffic classification: Lessons learned from a commercial-grade dataset," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4103–4118, 2021.

[15] T. Obasi and M. O. Shafiq, "An experimental study of different machine and deep learning techniques for classification of encrypted network traffic," in *2020 IEEE International Conference on Big Data (Big Data)*, 2020, pp. 4690–4699.

[16] Z. Chen, G. Cheng, Z. Wei, D. Niu, and N. fu, "Classify traffic rather than flow: Versatile multi-flow encrypted traffic classification with flow clustering," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2023.

[17] Z. Wu, Y.-n. Dong, J. Jin, H.-L. Wei, and G. Xie, "Multimedia traffic classification for imbalanced environment," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 3, pp. 1838–1852, 2022.

[18] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1757–1772, 2013.

[19] W. J. Scheirer, L. P. Jain, and T. E. Boult, "Probability models for open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 11, pp. 2317–2324, 2014.

[20] P. Wang, Z. Wang, F. Ye, and X. Chen, "Bytesgan: A semi-supervised generative adversarial network for encrypted traffic classification in sdn edge gateway," *Computer Networks*, vol. 200, p. 108535, 2021.

[21] L. Neal, M. Olson, X. Fern, W.-K. Wong, and F. Li, "Open set learning with counterfactual images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 613–628.

[22] Y. Chen, Z. Li, J. Shi, G. Gou, C. Liu, and G. Xiong, "Not afraid of the unseen: a siamese network based scheme for unknown traffic discovery," in *2020 IEEE Symposium on Computers and Communications (ISCC)*, 2020, pp. 1–7.

[23] H. He, Y. Lai, Y. Wang, S. Le, and Z. Zhao, "A data skew-based unknown traffic classification approach for tls applications," *Future Generation Computer Systems*, vol. 138, pp. 1–12, 2023.

[24] S. Le, Y. Lai, Y. Wang, and H. He, "An adaptive classification and updating method for unknown network traffic in open environments," *Computer Networks*, vol. 238, p. 110114, 2024.

[25] J. Zhang, F. Li, F. Ye, and H. Wu, "Autonomous unknown-application filtering and labeling for dl-based traffic classifier update," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 2020, pp. 397–405.

[26] T. Dahanayaka, Y. Ginige, Y. Huang, G. Jourjon, and S. Seneviratne, "Robust open-set classification for encrypted traffic fingerprinting," *Computer Networks*, vol. 236, p. 109991, 2023.

[27] Y. xuan Quan, Y. ning Dong, Y. Xiang, S. shan Chen, Z. jian Wang, and J. Jin, "Fast online classification of network traffic using new feature-embedded hierarchical structure," *Computer Networks*, vol. 237, p. 110106, 2023.

[28] K. Lin, X. Xu, and Y. Jiang, "A new semi-supervised approach for network encrypted traffic clustering and classification," in *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 2022, pp. 41–46.

[29] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of encrypted and vpn traffic using time-related," in *Proceedings of the 2nd international conference on information systems security and privacy (ICISSP)*, 2016, pp. 407–414.

[30] A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features," in *International Conference on Information Systems Security and Privacy*, vol. 2. SciTePress, 2017, pp. 253–262.

[31] C. Zhao, Q. Li, X. He, R. Wang, K. Chen, and Z. Liu, "Data augmentation of discrete sequential protocol messages based on recurrent generative adversarial networks," in *2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE)*, 2022, pp. 393–400.

[32] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-iiotset: A new comprehensive realistic cyber security dataset of iot and iiot applications for centralized and federated learning," *IEEE Access*, vol. 10, pp. 40 281–40 306, 2022.

[33] R. Das and G. Tuna, "Packet tracing and analysis of network cameras with wireshark," in *2017 5th International Symposium on Digital Forensic and Security (ISDFS)*, 2017, pp. 1–6.

[34] X. Qi, X. Wu, Y. Ji, X. Wang, and H. Li, "Research on classification of power load data based on libsvm," in *2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, vol. 2, 2019, pp. 158–162.

[35] M. Gallo, A. Finamore, G. Simon, and D. Rossi, "Fenxi: Deep-learning traffic analytics at the edge," in *2021 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 2021, pp. 202–213.

[36] P. W. Frey and D. J. Slate, "Letter recognition using holland-style adaptive classifiers," *Machine Learning*, vol. 6, no. 2, pp. 161 – 182, 1991.

[37] J. Hull, "A database for handwritten text recognition research," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 550–554, 1994.

[38] M. Bilenko, S. Basu, and R. J. Mooney, "Integrating constraints and metric learning in semi-supervised clustering," in *Proceedings of the Twenty-First International Conference on Machine Learning*, ser. ICML '04. New York, NY, USA: Association for Computing Machinery, 2004, p. 11.

[39] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "Knn model-based approach in classification," in *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings*. Springer, 2003, pp. 986–996.

[40] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[41] M. Lotfollahi, M. J. Siavoshani, R. S. H. Zade, and M. Saberian, "Deep packet: A novel approach for encrypted traffic classification using deep learning," *Soft Computing*, vol. 24, no. 3, pp. 1999–2012, 2020.

[42] Y. Fujiwara, Y. Ida, S. Kanai, A. Kumagai, and N. Ueda, "Fast similarity computation for t-sne," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, 2021, pp. 1691–1702.

Haotian Lu received the BE degree in electrical engineering and its automation from Changzhou Institute of Technology, in 2020. He is currently pursuing the Doctoral degree with the School of Communications and Information Engineering, Nanjing University of Posts and Telecommunications. His research interests include multimedia communications, and network traffic identification.

Yuning Dong received the M.Phil. degree in computer science from the Queen's University of Belfast (QUB) and the Ph.D. degree in electrical engineering from Southeast University. He is currently a Professor with the School of Communications and Information Engineering, Nanjing University of Posts and Telecommunications. He has authored/coauthored over 200 papers in IEEE and other technical journals and referred conference proceedings. He was a British Council Postdoctoral Fellow with Imperial College London from 1992 to 1993; a Visiting Scientist with the University of Texas from 1993 to 1995; and a Research Fellow with QUB and the University of Birmingham from 1995 to 1998. His research interests include wireless networking, multimedia communications, and network traffic identification.

Zhiyuan Wu received his ME. degree from Nanjing University of Posts and Telecommunications in 2023. His research interests include multimedia communications and network traffic identification.

HuaLiang Wei received the Ph.D. degree in the Department of Automatic Control and Systems Engineering, the University of Sheffield, UK, in 2004. He is currently a Senior Lecturer with the Department of Automatic Control and Systems Engineering (ACSE), The University of Sheffield, Sheffield, U.K. He previously held academic positions (Assistant Professor, Lec- turer and Associate Professor, 1992 - 2000) at Beijing Institute of Technology, China; he joined the department of ACSE in 2004 initially as a Senior Research Fellow immediately after the completion of the PhD study. His research interests include nonlinear system identification, machine learning, computational intelligence, data-driven modelling, and data mining, with applications in many multidisciplinary study areas such as engineering, bioengineering, computational medicine and neurophysiology, energy, space weather, social and environ- mental sciences, among others. Dr Wei is head of the laboratory of Dynamic Modelling, Data Mining and Decision Making.

Guanming Lu received the BE degree in radio engineering, in 1985, and the MS degree in communication and electronic systems, in 1988, both from the Nanjing University of Posts and Telecommunications, Nanjing, China, and the PhD degree in communication and information systems from Shanghai Jiao Tong University, Shanghai, China, in 1999. He is currently a professor with the School of Communication and Information Engineering, Nanjing University of Posts and Telecommunications, China. His current research interests include image processing, affective computing and machine learning.