Transform Sequential Data to Image for Detecting Covert Timing Channel

Xuwen Huang[1], Yonghong Chen[2], Xiaolong Zhuang[1], and Yuwei Lin[1]
(Corresponding author: Yonghong Chen)

School of Computer Science and Technology, Huaqiao University[1]
Xiamen Key Laboratory of Data Security and Blockchain Technology, Huaqiao University[2]
Xiamen, Fujian 361000, China
Email: djandcyh@163.com

(Received Aug. 30, 2023; Revised and Accepted Jan. 23, 2024; First Online Feb. 14, 2024)

Abstract

The network covert timing channel utilizes inter-packet delay to encode binary data to achieve information leakage and other purposes. In recent years, Covert Timing Channels (CTCs) have been demonstrated to be applicable across various protocols and networks, posing a significant threat to network security. Simultaneously, new CTCs have challenged the efficacy of CTC detection schemes. The recent ϵ-κlibur and ϵ-κlibur-O channels exhibit structural and characteristic similarities to legitimate channels, rendering ML-based detection methods like GAS and Snap ineffective. In this paper, we investigate an approach based on Gramian Angular Field (GAF), Markov Transition Field (MTF), and Recurrence Plot (RP) image processing. We further employ the knowledge-distilled and compressed image classification model MobileVit for detection. Our approach achieves a detection rate 98.24% for seven different channels within a sampling window of 64 IPDs. Experimental results demonstrate our proposed scheme's generality, sensitivity, and effectiveness.

Keywords: Covert Time Channel; Image Classification Network; Knowledge Distillation; MobileViT

1 Introduction

The network covert timing channel (CTC) is a technique that achieves information hiding through the manipulation of inter-packet delays based on constructed rules. In the past decade, numerous studies have utilized various protocols to implement CTCs. For instance, CTCs have been established through VoLTE traffic [46] and the MQTT protocol in the Internet of Things [27]. Moreover, CTCs have been implemented in diverse scenarios and networks, such as Vehicle Ad-Hoc Networks (VANETs) [38] and CAN bus communication in vehicular networks [12].

The characteristics of CTC make it have both positive and negative effects. CTCs are concealed, private and they possess low active drop rates. Malicious software employs CTCs to obfuscate its presence, making it challenging to detect [25]. On the other hand, CTCs can serve legitimate purposes, such as enabling reliable covert communications within MTS systems [20] or evading censorship [4]. They can also be applied in automotive control networks to enhance robustness and ensure stable communication [41].

Current detection methods for network covert timing channels predominantly rely on statistical and deep learning approaches. Statistical solutions extract statistical features, effectively detecting specific CTCs but lacking generality and sensitivity, particularly with larger sampling windows [17]. With hardware advancements, deep learning-based CTC detection schemes have emerged, demonstrating substantial effectiveness [10], although cost reduction remains an ongoing concern. Recently, Sebastian *et al.* [47] proposed ϵ-κlibur and ϵ-κlibur-O channels, imitating the distribution structure of normal Inter-Arrival Times (IATs and IPDs are the same) to undermine CTC detection performance. ML-based methods like Snap and GAS fail to effectively detect these channels. While enhanced ϵ-similarity has been proposed to improve ϵ-κlibur detection performance without significantly compromising original CTC detection. But it lacks generality, yielding AUC values between 0.80 and 0.88 for TRCTC. Therefore, new CTC detection solutions must simultaneously address both sensitivity and generality.

Inter-packet delay (IPD) represents univariate sequential data and CTC detection aims to classify it. This paper proposes leveraging IPD characteristics by transforming sequential data into images for classification. Inspired by significant achievements in computer vision, Wang *et al.* [43] introduced the encoding of time series data into various image types, such as GAF and MTF. By converting 12 standard datasets into combined GAF-MTF images, they demonstrated that the tiled CNN-based image classification model outperformed concurrent state-of-

the-art methods. GAF retains time dependencies between consecutive IPDs, MTF captures IPD probability transition patterns and Recursive Plot (RP) preserves non-stationarity and inherent similarity of IPDs. In previous work, our team applied GAF, MTF and RP image processing techniques for CTC detection. The combination of GAF-MTF-RP images yielded the best classification results [15]. Our approach achieved effective detection on publicly available and locally collected datasets, utilizing the MobileVit network as the optimal model and a sampling window of 64 IPDs. Building upon this prior research, this paper conducts detection for ϵ-κlibur and ϵ-κlibur-O. We also use knowledge distillation to compress the MobileVit model. In summary, the key contributions of this paper can be summarized as follows:

- We introduce detection methods for ϵ-κlibur and ϵ-κlibur-O, leveraging GAF, MTF and RP to extract feature matrices from sequential data IPD. These three matrices are combined into a three-channel image for classification.
- We employ knowledge distillation on the MobileVit network. Compared to a teacher network, using three teacher networks to train MobileVit network yields the best results. Distilled models offer advantages in terms of scale and speed, facilitating faster detection and reduced deployment costs.
- We conduct a series of comparative experiments, training the distilled network with optimal parameters, achieving the highest accuracy in classifying seven different channels. Furthermore, our distilled network outperforms popular distillation networks like Deit [40] and FasterNet [8] for this specific classification task.

The remaining sections of this paper are organized as follows: Section 2 discusses related work. In Section 3, we provide a detailed presentation of the detection approach, including its conceptual framework and intricate details. Section 4 outlines the configuration of our proposed scheme, including parameter settings and presents the results of our experimental analysis, which are subsequently showcased and analyzed. Finally, Section 5 concludes our work, summarizing the key findings and contributions.

2 Related Work

Research on the distribution patterns of normal traffic's inter-packet delay has shown that it does not adhere to a normal distribution. Early studies proposed that IPD follows distributions such as Pareto [31] and gamma distributions [29]. Recently Weibull distribution has been considered for studying anomalous IPD for Intrusion Detection Systems (IDS) [35]. We use the traffic data from GAS [17]. It has two datasets. Both datasets follow long-tailed distributions, while Backbone traffic has larger deviation than Lab, indicating more disperse IPDs and more fluctuant timing behavior [17]. It is sourced from CAIDA and some similar datasets have demonstrated adherence to the Weibull distribution such as MAWI, NUST [35]. Additionally, other laws like Benford's Law [30] and Zipf's Law [39] have also been employed to model normal traffic IPD characteristics. Some IDS systems construct models for binary classification based on the distribution patterns of normal IPD, without studying the IPD of CTCs. However, CTCs possess different IPD characteristics from normal traffic, allowing them to easily bypass IDS. In the aforementioned studies, variations in IPD distribution structure are observed due to different paths, periods and environments of the collected datasets. This variation may lead to differing findings. Similar challenges exist in current CTC detection research and CTC detection schemes which need to be validated across different datasets. Therefore, for CTC research, we advocate for the release of more publicly available datasets.

In recent literature on CTC classification research, CTCs are mainly categorized into Fixed-IPD Channels, Dynamic-IPD Channels, Combinatorial-IPD Channels and Delayed-IPD Channels [17]. These are represented by IPCTC [6], Jitterbug [34], LNCTC [33] and TRCTC [5], respectively, which are commonly studied as the primary detection targets.

2.1 Channel Classification

This paper primarily investigates and introduces six types of covert timing channels, including four typical CTCs and two recent CTCs.

2.1.1 IPCTC

Cabuk *et al.* [6] proposed IP Covert Timing Channel (IPCTC), where the sender and the receiver select parameter ω as the time interval for sending packets. For sending a bit 1, the sender sends a packet within ω time. For sending a bit 0, the sender remains silent for ω time.

2.1.2 Jitterbug

Shah *et al.* [34] proposed the passive CTC JitterBug. The sender and the receiver choose parameter ω. For sending a bit 1, the sender increases the time interval to a multiple of ω. For sending a bit 0, the sender increases the delay to a multiple of $\omega/2$, while avoiding multiples of ω.

2.1.3 LNCTC

Sellke *et al.* [33] proposed the LNCTC, using parameters L-bit and N consecutive IPDs. L-bits are embedded into N consecutive IPDs. The consecutive IPDs are of the set T. $T = \{D,\ D + 2^0 * d,\ \cdots,\ D + 2^L * d\}$. The sender and the receiver negotiate to determine the value of D and d.

2.1.4 TRCTC

Cabuk *et al.* [5] proposed a time-replay covert timing channel (TRCTC), which uses a legal traffic IPD set S. S is divided into two equal parts, S_0 and S_1. When the sender needs to send a bit 0, an IPD is randomly selected from S_0 to introduce a delay before sending bit 0. Similarly, when sending a bit 1, an IPD is selected from S_1.

2.1.5 ϵ-κlibur

Sebastian *et al.* [47] proposed ϵ-κlibur, modifying delays without compromising transmission bandwidth or reliability. Given an IPD list D, $D = \{d_1,\ d_2, \cdots,\ d_i\}$, modified delays are obtained as $D = \left\{d_1^{'},\ d_2^{'}, \cdots,\ d_i^{'}\right\}$. A threshold t is set. If $d_i \leq t$ and $d_i > t$, or $d_i > t$ and $d_i^{'} \leq t$. The count of errors increases. The impact score $I = E/D$, where smaller I is acceptable. Modifications are applied according to certain rules, with τ and 2τ as examples for CTC IPD lists. If d_i is not greater than t, $d_i^{'}$ is drawn from a normal distribution $N\,(0,\ (threshold/7))$ and takes positive values. If $d_i^{'}$ at this point exceeds t, $d_i^{'}$ is set to t. If the original d_i is greater than t, $d_i^{'}$ is drawn from $(1.5\tau,\ 2.4\tau)$ with a step size of 0.001. The author maintains I at 0.

2.1.6 ϵ-κlibur-O

Building upon ϵ-κlibur, an additional outlier value of 10τ is introduced to extend the distribution of time. This step makes the IPD distribution of ϵ-κlibur-O more similar to that of normal channels, as real-world conditions often involve fluctuations, causing delays to occasionally stand out.

Senders and receivers agree upon parameter modifications that still allow distinguishing between sending 1 and 0 using a threshold t. However, detection schemes are unaware of t and the IPD distribution of ϵ-κlibur and ϵ-κlibur-O closely resembles that of normal channels. As a result, detection performance is reduced.

2.2 CTC Detection Scheme Classification

CTC detection methods vary based on their distinctive features. This paper categorizes them as five parts.

2.2.1 Regularity, ϵ-similarity and Entropy-based Detection

ϵ-similarity [6] detection and compressibility [7] were proposed very early for CTC detection. Wendzel *et al.* [44] used compressibility score, ϵ-similarity and regularity for changing countermeasures in CTC detection. Gianvecchio *et al.* [11] proposed entropy-based CTC detection and Conditional Entropy Estimation (CEE) for detection. These are classical detection methods, they are often effective for one or two channels and require a large detection window.

2.2.2 Non-parametric Detection Methods

Mou *et al.* [28] proposed a sliding serial port detection scheme based on wavelet transform and Support Vector Machines (SVM). Liu *et al.* [21] proposed a detection approach utilizing Discrete Wavelet Multi-Resolution Transformation (DWMT). Rezaei *et al.* [32] detected CTCs by three non-parametric statistical tests, Spearman Rho, Wilcoxon Signed-Rank and Mann-Whitney-Wilcoxon rank sum test.

2.2.3 Machine Learning Based Detection Methods

Shrestha *et al.* [36] proposed a SVM classifier training fingerprinting CTC flows for detection. iglesias *et al.* [16] utilized decision trees to classify traffic. Li *et al.* [18] collected eight statistical features of IPD as communication fingerprints for classification by a Random Forest classifier.

Darwish *et al.* [9] proposed a deep learning-based hierarchical statistical classification detection method for CTCs. Han *et al.* [9] utilized K-Nearest Neighbors (KNN) for classification based on various statistical indicators. Al-Eidi *et al.* [2] proposed a hybrid model of CNN and LSTM for CTC detection using IPD sequential data. li *et al.* [19] proposed similar model using CNN and Transformer architectures for detection .

2.2.4 Image Processing and Sequential Data Processing-based Detection

Snap is the first CTC solution using image processing [1]. Wu *et al.* [45] proposed an approach based on sequential data, firstly transforming it into symbol time series, computing State Transition Probability Matrices (STPM) and finally classifying based on similarity scores. Sun *et al.* [37] proposed a detection approach for CTCs using GAF images and GAN network(CD-ACGAN). Based on Snap, Al-Eidi *et al.* [3] proposed a CNN image classification model for detection.

2.2.5 Other Detection Methods

Lu *et al.* [24] proposed a multi-dimensional feature detection analyzed from the perspectives of shape, change pattern and data statistics. Wang *et al.* [42] proposed a detection scheme for CTCs based on perceptual hashing.

2.3 Image Classification Networks

Image classification has been an active research trend worldwide, greatly facilitated by the emergence of artificial intelligence. The introduction of deep learning algorithms has brought various innovations in image classification. Each year witnesses the emergence of important networks for image classification. Notable examples include ResNet [13], Swin Transformer [22], Convnext [23] and MobileVit [26] etc.

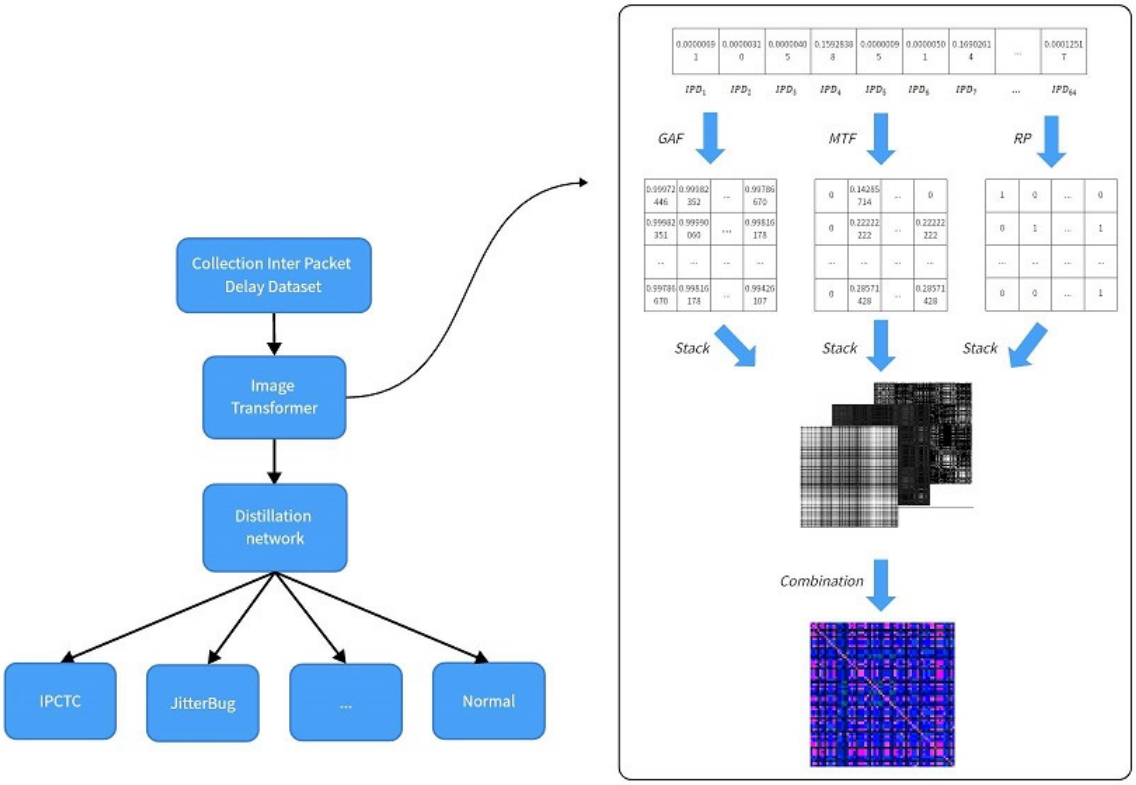

Figure 1: The proposed scheme

2.4 Knowledge Distillation

Knowledge distillation was proposed by Geoffrey Hinton [14], compresses knowledge learned by multiple models into a single model that is easier to deploy. Currently, knowledge distillation has been applied to lightweight networks like Deit and FasterNet in image classification.

3 Approach

This section provides a detailed description of our detection scheme. First, we extract three matrices of IPD, GAF, MTF and RP. These matrices are subsequently normalized and mapped to pixels, forming a three-channel image. Let the sequential data of IPD be represented by $X = \{x_1, x_2 \cdots x_n\}$, with a length of N. The proposed scheme is depicted as Figure 1.

3.0.1 GAF

Mapping X to the range$[-1,\ 1]$:

$$\widetilde{p_i} = \frac{(x_i - max(X) + x_i - min(X))}{max(X) - min(X)}$$

Encoding values as cosine angles and time stamps as radii, thus representing Cartesian coordinates in polar form:

$$\begin{cases} \varnothing_i = arccos\left(\widetilde{p_i}\right), -1 \le \widetilde{p_i} \le 1 \\ r_i = \frac{i}{N}, 1 \le i \le N \end{cases}$$

The GAF matrix is represented by the sum of triangles between each pair of points, preserving their correlation:

$$G = \begin{bmatrix} cos\left(\varnothing_1 + \varnothing_1\right) & cos\left(\varnothing_1 + \varnothing_2\right) & \cdots & cos\left(\varnothing_1 + \varnothing_n\right) \\ cos\left(\varnothing_2 + \varnothing_1\right) & cos\left(\varnothing_2 + \varnothing_2\right) & \cdots & cos\left(\varnothing_2 + \varnothing_1\right) \\ \cdots & \cdots & \cdots & \cdots \\ cos\left(\varnothing_n + \varnothing_1\right) & cos\left(\varnothing_n + \varnothing_2\right) & \cdots & cos\left(\varnothing_n + \varnothing_n\right) \end{bmatrix}$$

3.0.2 MTF

Dividing X into Q quantile units, represented by quantiles $q\left(i, j \in \{1, 2, \ldots, Q\}\right)$, each x_i corresponding to a q_i value in the one-dimensional data sequence. Each x_{i+1} corresponding to a q_j. W_{ij} is used to denote the probability that q_i followed by q_j:

$$W_{ij} = P\left(x_t \in q_i | x_{t+1} \in q_j\right), 1 \le t \le N - 1$$

The Markov transition matrix is constructed by these probabilities:

$$M = \begin{bmatrix} W_{ij}|x_1 \in q_i, x_1 \in q_j & \cdots & W_{ij}|x_1 \in q_i, x_n \in q_j \\ W_{ij}|x_2 \in q_i, x_1 \in q_j & \cdots & W_{ij}|x_2 \in q_i, x_n \in q_j \\ \cdots & \cdots & \cdots \\ W_{ij}|x_n \in q_i, x_1 \in q_j & \cdots & W_{ij}|x_n \in q_i, x_n \in q_j \end{bmatrix}$$

3.0.3 RP

Reconstructing the one-dimensional time series into an m-dimensional phase space. For a time series X, its sampling

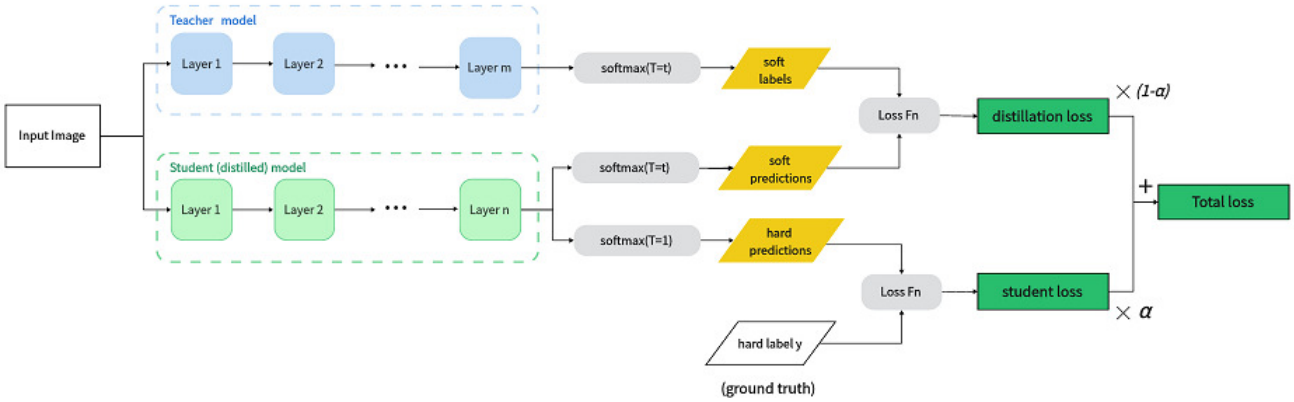

Figure 2: Knowledge distillation process

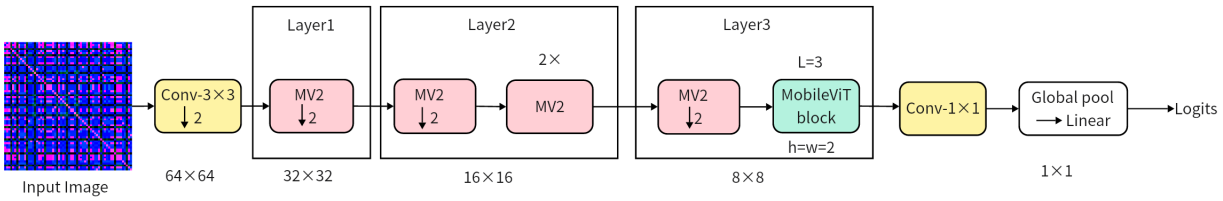

Figure 3: Our modified MobileViT structure

time interval is determined as Δt, along with embedding dimension m and delay time τ, to reconstruct X. The reconstructed dynamic system is defined as:

$$X_i = \left[x_i, x_{i+\tau}, ..., x_{i+(m-1)\tau}\right], i = 1, 2, 3, ..., n-(m-1)\tau$$

Calculating the distance S_{ij} between x_i and x_j in the reconstructed phase space:

$$S_{ij} = \|x_i - x_j\|, \ j = 1, 2, 3, ..., n-(m-1)\tau$$

$\|\bullet\|$ represents the norm. Computing the recursive value:

$$R(i,j) = \Theta(\varepsilon - S_{ij})$$

ε is the threshold. $\Theta(\bullet)$ is the Heaviside function. $\Theta(x \geq 0) = 1, \Theta(x < 0) = 0$, RP is composed of these recursive values.

3.0.4 Knowledge Distillation

For the GAF-MTF-RP images transformed from IPD, multiple models with different weights were trained, including ResNet, Swin-Transformer and ConvNeXt, which generally outperformed lightweight models [15]. However, these heavyweight models showed drawbacks in terms of model size, training time and detection time. In this study, ResNet, Swin-Transformer and ConvNeXt were selected as teacher networks, with MobileVit chosen as the lightweight student network. The experiments employed ResNet50, the tiny version of Swin-Transformer and the small version of ConvNeXt, all following the original paper's structures. Figure 2 shows the knowledge distillation flow chart.

During the experimental process, our student network, MobileVit, was tailored to accommodate 64-pixel images. While ensuring accuracy, we aimed to minimize the number of layers and parameters to meet scalability and speed requirements. Our optimized student network, MobileVit, differs from the xx-small (xxs) model of MobileVit by transitioning from a 5-layer architecture to a 3-layer architecture, resulting in a significant reduction in parameter count. Our modified MobileViT structure is shown in Figure 3. In the 3rd layer, a transformer block is present, with a total of 4 heads in the multi-head self-attention mechanism. The sequence length of intermediate tokens within the Feed-Forward Network (FFN) is set to 128.

After passing through the teacher networks and undergoing softmax, the resulting probability distribution had significant disparities, contributing minimally to the loss function. The concept of "temperature" was introduced to smooth the Logits. The smoothing formula is as follows:

$$q_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}$$

z_i represents the value of the i-th Logit, T is the distillation temperature and q_i is the probability value. Furthermore, the distillation loss is computed and then backpropagation optimizes the student neural network. The distillation loss formula is as follows:

$$L_{dis} = L_{KL}\left(q_i^{teacher}, q_i^{student}\right)$$

$L_{KL}()$ is the KL divergence, $q_i^{teacher}$ is the smoothed value of the i-th teacher network and $q_i^{teacher}$ the smoothed value of

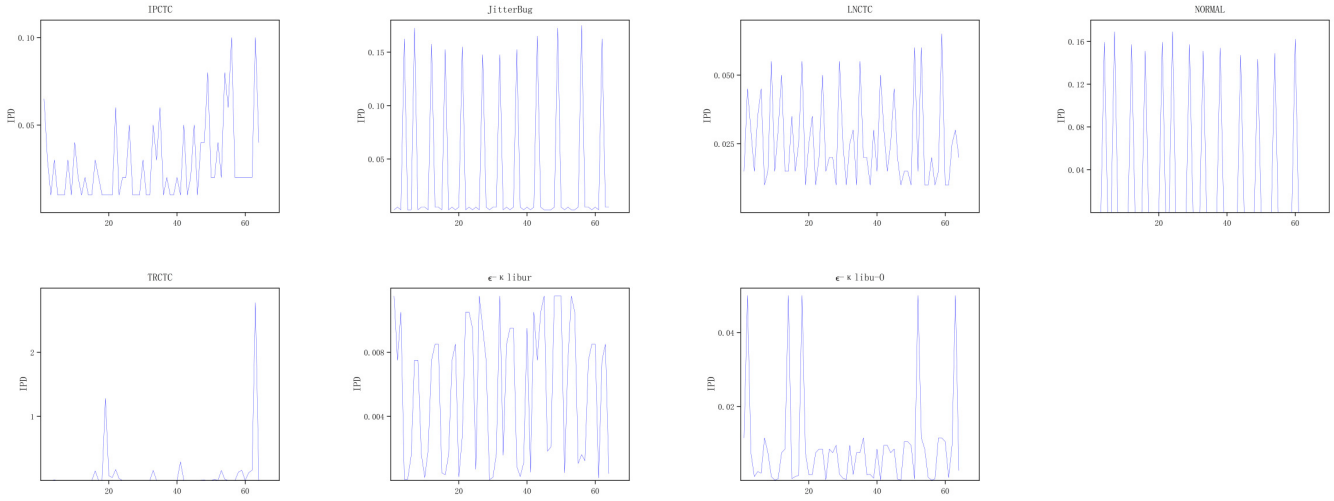

Figure 4: Statistical images of seven channels

the i-th student network. For combining multiple teacher networks, a set of weights w_k is introduced, calculated as follows:

$$w_k = softmax\left(v_k\right) = \frac{exp\ v_k}{\sum_{k=1}^{nt} exp\ v_k}, \sum_{k=1}^{nt} w_k = 1, w_k \in [0,1]$$

where nt represents the number of teacher networks, which is three. The outputs obtained from each teacher network after processing the input data are weighted accordingly:

$$z_i = x_1 \bigotimes w_1 + x_2 \bigotimes w_2 + \cdots + x_{nt} \bigotimes w_{nt}$$

4 Evaluation and Results

4.1 Dataset and Environment

The dataset used in this study is obtained from the GAS method's publicly available dataset [15]. We use the dataset from WAN. Its IPD is more dispersed, more fluctuating and more difficult to detect than the traffic generated by the experiment. The Python version is 3.9, the Torch version is 1.11.0 and the GPU used is the NVIDIA GeForce RTX 3080. The learning rates for ResNet, Swin-T, ConvNeXt and MobileVit are set to 0.0001. The batch size is 64 and the number of epochs is 100.

4.2 Evaluation Metrics

We use the following four metrics to measure the performance of our classification models, TP (True Positive) represents images that are correctly predicted as a CTC by MobileVit. TN (True Negative) represents images that are correctly predicted as legitimate traffic by MobileVit. FP (False Positive) represents images that are incorrectly predicted as a CTC. FN (False Negative) represents images that are incorrectly predicted as legitimate traffic. These metrics were used to calculate the performance indicators of the models:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN)$$

$$Precision\ = TP / (TP + FP)$$

$$Recall\ = TP / (TP + FN)$$

$$F1 - Score = 2 \times Precision\ \times Recall / (Precision\ + Recall)$$

4.3 Image Processing

A dataset created using the backbone traffic from GAS and tools used in ϵ-κlibur and ϵ-κlibur-O was employed to produce ϵ-κlibur and ϵ-κlibur-O flow IPD. The statistical images for the seven data types are as Figure 4.

From the line chart, certain patterns such as relativity, periodicity and stability can be observed. These characteristics serve as the basis for IPD classification. IPCTC exhibits periodic patterns with upward segments, while JitterBug and Normal channels have many sharp sections, indicating significant fluctuations. LNCTC shows less pronounced periodicity, with the distribution primarily in the middle range of 0.012 to 0.06. TRCTC's IPD distribution has significant gaps, as it randomly delays an IPD time to send either 0 or 1, resulting in uneven distribution due to the delay of one small TRCTC segment among the larger Normal segments. For normal flow data and TRCTC, some IPD values are very small, close to 0, making them difficult to observe in the images. In real environments, most data packets are transmitted quickly, resulting in generally lower IPD values. ϵ-κlibur's IPD distribution is coherent, lacking severe fluctuations. ϵ-κlibur-O also exhibits uneven distribution but with more prominent extreme values compared to Normal and fewer than TRCTC. While classification based on the line chart is not accurate, transforming sequential data into images yields seven distinct images as shown in Figure 5.

4.4 Detection Effect of Teacher and Student Networks

Initially, the effects of the three networks without distillation were compared. As seen in the Figure 6, ConvNeXt achieved the highest overall accuracy of 98.43%, followed by Swin-T and ResNet at 98.33% and 98.18% respectively. MobileVit had the lowest accuracy at 97.15%. Compared to heavyweight networks, the lightweight MobileVit had lower accuracy. Additional parameters and layers of heavyweight networks contributed to their stronger learning capability in this image classification task.

Distillation was performed using one-to-one and three-to-

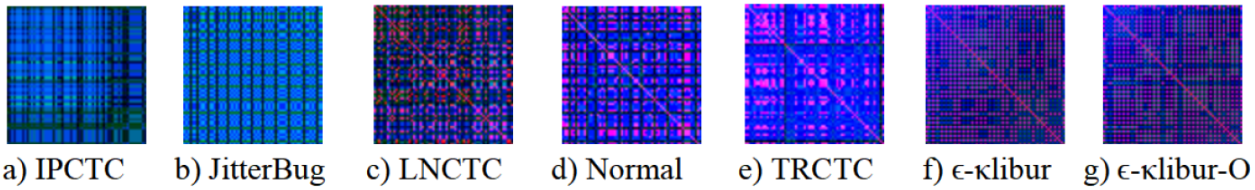

Figure 5: Transformed images of seven channels

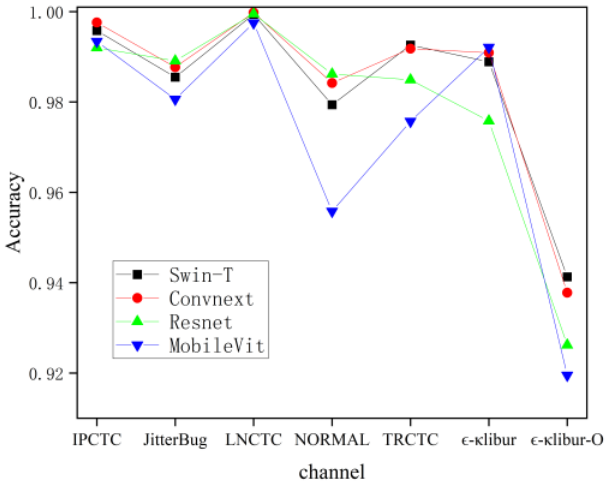

Figure 6: Seven channels accuracy of undistilled networks

one network configurations, as illustrated in Figure 7. In the one-to-one network, the highest accuracy achieved when ResNet trained MobileVit was 98.06%. Swin-T and ConvNeXt both had an accuracy of 98.01%, with little difference. In the three-to-one network, the highest accuracy was 98.24% (with $T = 7, \alpha = 0.3$), surpassing the accuracy of individual teacher networks. Soft labels provided by the three networks played a role in training the student network. Although the accuracy of the distillation network model decreased compared to the individual teacher network for classification, using multiple teacher networks enhanced the accuracy by 1.09% compared to the standalone student network, proving that the distilled MobileVit benefited from the knowledge of the teacher networks.

To find the best distillation network model, T and α were adjusted. The resulet is shown in Figure 8. When $T = 7$, the average accuracy of each channel reached a maximum of 98.24%. The next best accuracy was 98.10% when $T = 10$, followed by $(T = 3) > (T = 5) > (T = 1)$, with accuracies of 98.02%, 97.98% and 97.67% respectively. The comprehensive accuracy of the combined parameters $\alpha = 0.3$ was 98.24%, higher than 98.13% with $\alpha = 0.5$ and 98.01% with $\alpha = 0.7$. According to Figure 8 and Figure 9, the optimal values were T=7 and $\alpha = 0.3$.

4.5 Epoch and Accuracy Curves

In the field of image classification networks, Deit and FasterNet are two popular lightweight image classification networks that achieve good classification results on ImageNet-1K. In this experiment, the distilled MobileVit was compared to Deit and FasterNet. Deit achieved a peak accuracy of 96.27%, while FasterNet reached 98.06%. The code was obtained from the original paper. For this classification task, MobileVit outperformed in terms of classification accuracy. The comparison results are shown in Figure 10.

4.6 Detection Metrics for MobileVit in Different Channels

We collected various evaluation indicators of the scheme under seven channels and the results are shown in Table 1. In terms of various channel metrics, MobileVit achieved detection metrics above 98% for all channels except ϵ-κlibur-O, demonstrating effective detection. Specific attention was given to ϵ-κlibur and ϵ-κlibur-O. For ϵ-κlibur, the model's classification accuracy was close to 99%.

Although the IPD distribution of ϵ-κlibur is close to that of legitimate channel traffic, differences still exist. We believe that the differences in the construction process, where authors modified di smaller than t using a normal distribution, might be improved by using a Weibull distribution. In the case of ϵ-κlibur-O, the detection accuracy was only 93.75%, making it more likely to be classified as TRCTC and normal channels. We attribute this to the same obstacles encountered by the Snap method when detecting ϵ-κlibur-O. The presence of outliers in ϵ-κlibur-O affects the correlations between values, leading to lower classification accuracy.

4.7 Compare with Other CTC Detection Methods

We compared nine CTC detection methods, as shown in Figure 11. Among them, five are classical methods often used for comparison: ϵ-similarity, K-S, Regularity, Entropy and CEE. These five methods have very low average detection accuracy for the seven channels because they typically work effectively for only one or two channels and require a larger IPD sampling window. In this study, all methods sampled 64 IPDs as a window, limiting their detection performance. Compared to Darwish's hierarchical statistics and deep learning methods (referred to as DNN), their detection accuracy is 86%. DNN's accuracy is lower for the detection of ϵ-κlibur and ϵ-κlibur-O channels, leading to an average accuracy that is not very high. SnapCatch, which extracts image features in a relatively simple way and distinguishes images based on eight statistical values, also has lower average detection accuracy compared to our approach. In comparison to Sun's DC-ACGAN method, although there are similarities in the image feature extraction, the difference in the performance of the image classification network leads to our method achieving higher average accu-

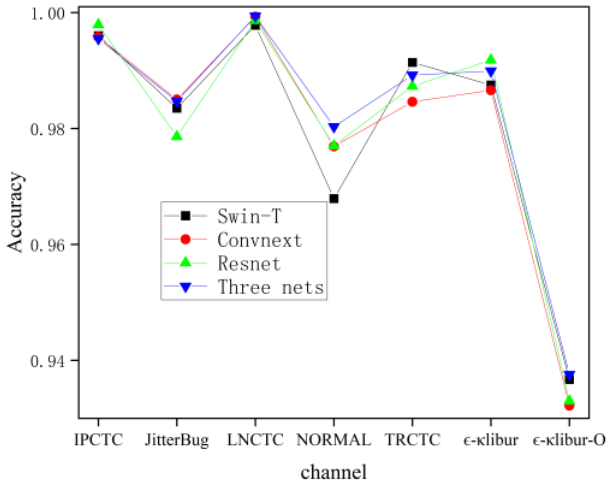

Figure 7: Seven channels accuracy of distilled networks

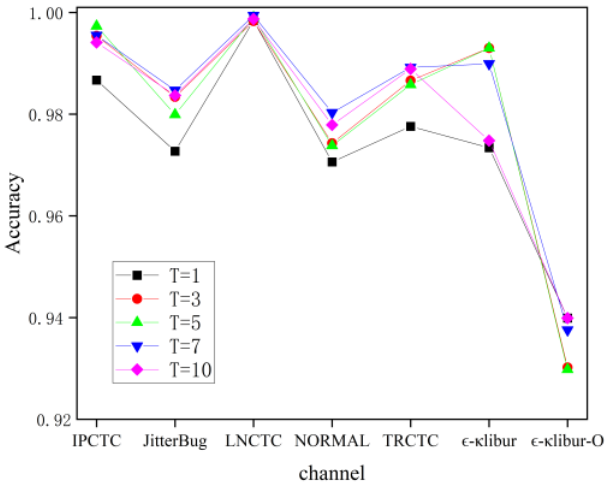

Figure 8: Seven channels accuracy with different T

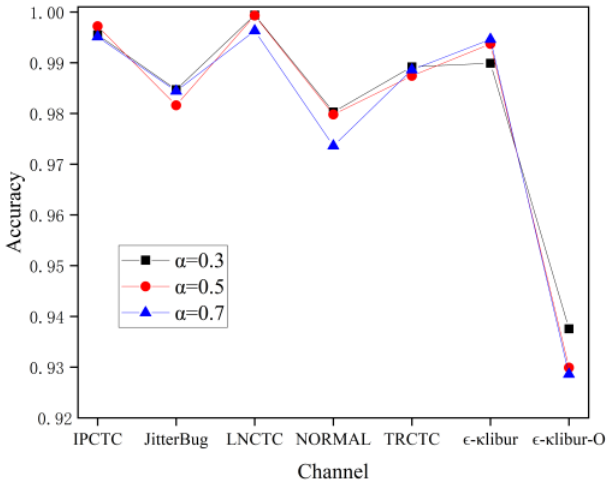

Figure 9: Seven channels accuracy with different α.

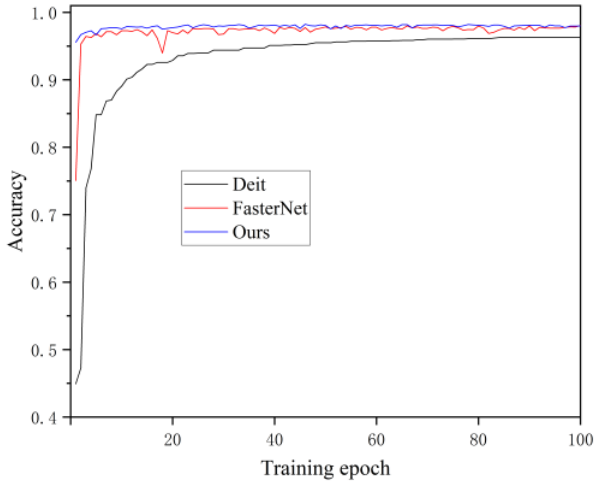

Figure 10: Experimental results compared with Deit and FasterNet

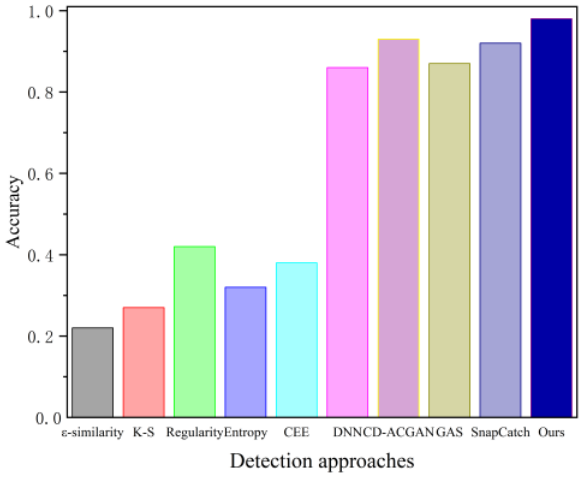

Figure 11: Accuracy of different CTC detection methods

racy. During the restoration phase, GAN network training is unstable, with significant accuracy fluctuations and requiring a longer training period. For the GAS method, because it was originally designed for blind detection, adapting it from binary to multi-class methods resulted in an accuracy of 87%, with lower detection accuracy for ϵ-κlibur and ϵ-κlibur-O channels. In summary, compared to the other eight detection methods, our detection scheme achieves the highest average detection accuracy for each channel.

4.8 Network Params and FLOPs

We employed the thop library in Python to meticulously document the parameters and computational load of the networks utilized in our experiments, as outlined in Table 2. Notably, within the context of this study, the optimized MobieVit model exhibited a significant reduction in parameters, distinguishing itself as the network with the least parameter count and computational load among all the networks investigated.

Table 1: Evaluation indicators of seven channels

Channel Class	Accuracy	Precision	Recall	F1 Score
IPCTC	99.55%	99.49%	99.50%	99.50%
JitterBug	98.47%	99.33%	97.99%	98.66%
LNCTC	99.94%	99.75%	99.90%	99.82%
Normal traffic	98.03%	97.83%	97.08%	97.40%
TRCTC	98.92%	97.25%	99.16%	98.20%
ϵ-κlibur	98.99%	96.08%	98.86%	96.95%
ϵ-κlibur-O	93.75%	98.79%	93.03%	96.46%
Total	98.24%	98.36%	97.93%	98.14%

Table 2: Network Params and FLOPs

Network	Params	FLOPs
Swin-T	27.50	737.61
ConvNext	27.83	364.62
Resnet	21.29	300.27
Deit	5.68	93.06
FasterNet	3.91	29.24
Ours	0.16	14.72

5 Conclusion

In conclusion, this study adopted a novel approach to transforming sequential data into images using GAF-MTF-RP, a three-channel image representation of IPD. The multi-teacher distillation was applied to the MobileVit network for detecting various covert channels. Through parameter tuning, we achieved classification accuracy above 98% for six types of channels, although not as high for ϵ-κlibur-O. However, our approach demonstrates higher generality compared to GAS and Snap, requiring only a detection window size of 64 for high sensitivity. GAS uses 250 IPDs on average to achieve effective detection and Snap uses 256 IPDs. Our distilled MobileVit model outperforms some popular distillation networks in this classification task.

Looking ahead, we aim to enhance the classification performance on ϵ-κlibur-O. Additionally, we plan to investigate the inherent structure of images from normal channels and utilize semi-supervised image classification networks to achieve blind detection capabilities.

References

[1] S. Al-Eidi, O. Darwish, Y. Chen, G. Husari, "Snapcatch: automatic detection of covert timing channels using image processing and machine learning," *IEEE Access*, vol. 9, pp. 177–191, 2020.

[2] S. Al-Eidi, O. Darwish, Y. Chen, M. Maabreh, Y. Tashtoush, "A deep learning approach for detecting covert timing channel attacks using sequential data," *Cluster Computing*, pp. 1–11, 2023.

[3] S. Al-Eidi, O. Darwish, G. Husari, Y. Chen, M. Elkhodr, "Convolutional neural network structure to detect and localize ctc using image processing," in *2022 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, pp. 1–7. IEEE, 2022.

[4] D. Barradas, N. Santos, L. Rodrigues, V. Nunes, "Poking a hole in the wall: Efficient censorship-resistant internet communications by parasitizing on webrtc," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pp. 35–48, 2020.

[5] Serdar Cabuk. *Network covert channels: Design, analysis, detection, and elimination*. PhD thesis, Purdue University, 2006.

[6] S. Cabuk, C. E. Brodley, C. Shields, "Ip covert timing channels: design and detection," in *Proceedings of the 11th ACM conference on Computer and communications security*, pp. 178–187, 2004.

[7] S. Cabuk, C. E. Brodley, C. Shields, "IP covert channel detection," *ACM Transactions on Information and System Security (TISSEC)*, vol. 12, no. 4, pp. 1–29, 2009.

[8] J. Chen, S. H. Kao, H. He, W. Zhuo, S. Wen, C. H. Lee, and S. H. G. Chan, "Run, don't walk: Chasing higher flops for faster neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12021–12031, 2023.

[9] O. Darwish, A. Al-Fuqaha, G. B. Brahim, I. Jenhani, and A. Vasilakos, "Using hierarchical statistical analysis and deep neural networks to detect covert timing channels," *Applied Soft Computing*, vol. 82, p. 105546, 2019.

[10] M. A. Elsadig and A. Gafar, "Covert channel detection: machine learning approaches," *IEEE Access*, vol. 10, pp. 38391–38405, 2022.

[11] S. Gianvecchio and H. Wang, "An entropy-based approach to detecting covert timing channels," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 6, pp. 785–797, 2010.

[12] B. Groza, L. Popa, and P. S. Murvay, "Canto-covert authentication with timing channels over optimized traffic flows for can," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 601–616, 2020.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[14] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[15] X. Huang, Y. Chen, Z. Li, and T. Zhan, "Detection of network time covert channels based on image processing," in *Proceedings of the 2023 4th International Conference on Computing, Networks and Internet of Things*, pp. 701–707, 2023.

[16] F. Iglesias, V. Bernhardt, R. Annessi, and T. Zseby, "Decision tree rule induction for detecting covert timing channels in tcp/ip traffic," in *Machine Learning and Knowledge Extraction: First IFIP TC 5, WG 8.4, 8.9, 12.9 International Cross-Domain Conference, CD-MAKE 2017, Reggio, Italy, August 29–September 1, 2017, Proceedings 1*, pp. 105–122. Springer, 2017.

[17] H. Li, T. Song, and Y. Yang, "Generic and sensitive anomaly detection of network covert timing channels," *IEEE Transactions on Dependable and Secure Computing*, 2022.

[18] Q. Li, P. Zhang, Z. Chen, and G. Fu, "Covert timing channel detection method based on random forest algorithm," in *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, pp. 165–171. IEEE, 2017.

[19] Z. Li, Y. Chen, Z. Teng, and X. Huang, "Contra: A covert timing channel detection approach for little covert information in a network," in *Proceedings of the 2023 4th International Conference on Computing, Networks and Internet of Things*, pp. 614–620, 2023.

[20] C. Liang, T. Baker, Y. Li, R. Nawaz, and Y. A. Tan, "Building covert timing channel of the iot-enabled mts based on multi-stage verification," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[21] A. Liu, J. Chen, and H. Wechsler, "Real-time covert timing channel detection in networked virtual environments," in *Advances in Digital Forensics IX: 9th IFIP WG 11.9 International Conference on Digital Forensics, Orlando, FL, USA, January 28-30, 2013, Revised Selected Papers 9*, pp. 273–288. Springer, 2013.

[22] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

[23] Z. Liu, H. Mao, C. Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11976–11986, 2022.

[24] S. Lu, Z. Chen, G. Fu, and Q. Li, "A novel timing-based network covert channel detection method," in *Journal of Physics: Conference Series*, vol. 1325, p. 012050. IOP Publishing, 2019.

[25] W. Mazurczyk and L. Caviglione, "Information hiding as a challenge for malware detection," *arXiv preprint arXiv:1504.04867*, 2015.

[26] S. Mehta and M. Rastegari, "Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer," *arXiv preprint arXiv:2110.02178*, 2021.

[27] A. Mileva, A. Velinov, L. Hartmann, S. Wendzel, and W. Mazurczyk, "Comprehensive analysis of mqtt 5.0 susceptibility to network covert channels," *Computers & security*, vol. 104, p. 102207, 2021.

[28] S. Mou, Z. Zhao, S. Jiang, Z. Wu, and J. Zhu, "Feature extraction and classification algorithm for detecting complex covert timing channel," *Computers & Security*, vol. 31, no. 1, pp. 70–82, 2012.

[29] A. Mukherjee, "On the dynamics and significance of low frequency components of internet load," 1992.

[30] M. E. Newman, "Power laws, pareto distributions and zipf's law," *Contemporary physics*, vol. 46, no. 5, pp. 323–351, 2005.

[31] V. Paxson and S. Floyd, "Wide area traffic: the failure of poisson modeling," *IEEE/ACM Transactions on networking*, vol. 3, no. 3, pp. 226–244, 1995.

[32] F. Rezaei, M. Hempel, and H. Sharif, "Towards a reliable detection of covert timing channels over real-time network traffic," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 3, pp. 249–264, 2017.

[33] S. H. Sellke, C. C. Wang, S. Bagchi, and N. Shroff, "TCP/IP timing channels: Theory to implementation," in *IEEE INFOCOM 2009*, pp. 2204–2212. IEEE, 2009.

[34] G. Shah, A. Molina, M. Blaze, *et al.*, "Keyboards and covert channels.," in *USENIX Security Symposium*, vol. 15, p. 64, 2006.

[35] R. Sharma, A. Guleria, and R.K. Singla, "An overview of flow-based anomaly detection," *International Journal of Communication Networks and Distributed Systems*, vol. 21, no. 2, pp. 220–240, 2018.

[36] P. L. Shrestha, M. Hempel, F. Rezaei, and H. Sharif, "A support vector machine-based framework for detection of covert timing channels," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 2, pp. 274–283, 2015.

[37] C. Sun, Y. Chen, H. Tian, and S. Wu, "Covert timing channels detection based on auxiliary classifier generative adversarial network," *IEEE Open Journal of the Computer Society*, vol. 2, pp. 407–418, 2021.

[38] S. Taheri, M. Mahdavi, and N. Moghim, "A dynamic timing-storage covert channel in vehicular ad hoc networks," *Telecommunication Systems*, vol. 69, pp. 415–429, 2018.

[39] T. Tao, "Benford's law, zipf's law, and the pareto distribution," *Retrieved from*, 2009.

[40] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

[41] S. Vanderhallen, J. Van Bulck, F. Piessens, and J. T. Mühlberg, "Robust authentication for automotive control networks through covert channels," *Computer Networks*, vol. 193, p. 108079, 2021.

[42] L. Wang and Y. Chen, "A perceptual hash-based approach to detect covert timing channels.," *Int. J. Netw. Secur.*, vol. 22, no. 4, pp. 686–697, 2020.

[43] Z. Wang, T. Oates, *et al.*, "Encoding time series as images for visual inspection and classification using tiled convolutional neural networks," in *Workshops at the twenty-ninth AAAI conference on artificial intelligence*, vol. 1. AAAI Menlo Park, CA, USA, 2015.

[44] S. Wendzel, F. Link, D. Eller, and W. Mazurczyk, "Detection of size modulation covert channels using countermeasure variation.," *J. Univers. Comput. Sci.*, vol. 25, no. 11, pp. 1396–1416, 2019.

[45] S. Wu, Y. Chen, H. Tian, and C. Sun, "Detection of covert timing channel based on time series symbolization," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 2372–2382, 2021.

[46] X. Zhang, L. Zhu, X. Wang, C. Zhang, H. Zhu, and Y. Tan, "A packet-reordering covert channel over volte voice and video traffics," *Journal of Network and Computer Applications*, vol. 126, pp. 29–38, 2019.

[47] S. Zillien and S. Wendzel, "Weaknesses of popular and recent covert channel detection methods and a remedy," *IEEE Transactions on Dependable and Secure Computing*, 2023.

Biography

Xuwen Huang was born in Jiangxi province, China in 1998. Between 2016 and 2020, he majored in software engineer with East China Jiaotong University School of Science and Technology, Nanchagn, China. He reveived the bachelor's degree in engineering. He is currently working toward the M.S degree with Huaqiao University, Xiamen, China. His research interests include: cyber security, machine learning and time series analysis.

Yonghong Chen received his Ph.D. degree in engineering, from the School of Automation of Chongqing University in July 2005. From September 2006 to October 2007, he went to Kyoto University, Japan as an academic visitor for one year. Now he is a professor and master's tutor of Huaqiao University. He is mainly engaged in the research of computer network and information security, mobile Internet technology, Internet of things technology, technology integration of mobile communication and Internet of things, network and information security.

Xiaolong Zhuang was born in Quanzhou, China in 1999. He obtained a Bachelor's degree in Software Engineering from Jinling College, Nanjing University. He is currently pursuing a master's degree at Quanzhou Overseas Chinese University in China. His main research direction is covert channel detection and the use of perceptual hashing.

Yuwei Lin was born in Fujian province, China in 1997. He received his Bachelor's degree in Engineering from Qing Gong College, North China University of Science and Technology. He is currently pursuing a master's degree at Huaqiao University. His main research interests are network security and network covert channel detection.