

Received July 17, 2020, accepted July 26, 2020, date of publication August 7, 2020, date of current version August 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3015080

Detection and Forensics of Encryption Behavior of Storage File and Network Transmission Data

SONGBIN LI¹ AND PENG LIU¹

Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China

Corresponding author: Songbin Li (lisongbin@mail.ioa.ac.cn)

This work was supported in part by the Important Science and Technology Project of Hainan Province under Grant ZDKJ201807, in part by the Hainan Provincial Natural Science Foundation of China under Grant 618QN309, in part by the Scientific Research Foundation Project of Haikou Laboratory, Institute of Acoustics, Chinese Academy of Sciences, and in part by the IACAS Young Elite Researcher Project under Grant QNYC201829 and Grant QNYC201747.

ABSTRACT In recent years, with the widespread application of encryption technology, criminals can hide malicious data without being discovered by security regulatory authorities, which has brought serious challenges to computer forensic investigation. Therefore, it is urgent to study the technology of detection and forensics of encrypted data. This paper proposes a method for encryption detection based on a deep convolutional neural network. The method first converts the raw data into two-dimensional matrixes as the input of the convolutional neural network. Then, the multiscale feature extraction mechanism with multiple activation functions is utilized to provide representative features as the input of subsequent layers. Next, the residual learning operation can further enhance the discrimination of features. By this mean, a network which can automatically extract and learn global contextual information of encrypted data is constructed. The experiment results show that the proposed method achieves high accuracy in the detection of storage file and network transmission data compare to the competitive methods and the detection accuracy on different types of mixed data is higher than 99%. Moreover, the proposed method can accurately detect data encrypted with different algorithms. The average detection rate of DES-encrypted data is higher than that of competitors by more than 5%.

INDEX TERMS Encryption detection and forensics, storage file, network transmission data, deep learning.

I. INTRODUCTION

With the increasing popularity of the Internet and increasing awareness of privacy protection, encryption technology has been widely used in all aspects of daily life [1]. For example, people encrypt important files stored in their computers to prevent browsing and theft, use encryption technology to pre-process when transferring files through communication tools, or transfer important information through VPNs. Encryption technology can provide legal users with higher data security, but like many other technologies, encryption technology is also a double-edged sword. It can be used to protect the privacy of users and prevent legal data leakage, but it can also be used by criminals to hide their illegal data. Encryption technology enables criminals to transmit malicious data through a secure channel. In addition, the widespread use of encryption technology has also brought challenges to digital investigations and forensics. The main

objective of a computer forensic investigation is to search and analyze any data and files that may contain any illegal information. Generally, most of the illegal information is hidden by encryption. How to find the encrypted data is the main problem faced by computer forensics which is also the key to further analysis of criminal information. Thus, research on detection and forensics of encryption data is urgent.

There are two scenarios of data encryption: one is to encrypt the data and store it in various storage media (such as hard disks, USB flash disks, optical disks, etc), these media can be easily transported. The second is to encrypt the data and then transmit it over the network or directly encrypt the original data through an encrypted channel (such as VPN). Whatever encryption method is used, the input accepted by the detector is a binary data stream. Thus, the essence of encryption detection is to quickly determine whether the binary data is encrypted.

As a basic step for more refined identification, encryption detection is of great importance. Currently, the entropy-based method [2] is a general method to identify whether data

The associate editor coordinating the review of this manuscript and approving it for publication was Lefei Zhang¹.

is encrypted or not. However, the entropy-based method is ineffective in distinguishing between compressed media data and encrypted data [3]. Therefore, it is insufficient for detecting a large amount of audio and video data. On the other hand, the machine learning-based method [4], [5] needs to carry out the random statistical test on test data to obtain random feature values, and then feature selection is conducted. The feature selection operation is often completed by manual design, which leads to the problems of unstable quality, low representativeness, and low robustness, etc. Finally, the machine learning algorithm is used for model training and detection. The machine learning-based method is not end-to-end because it requires hand-crafted features. Therefore, the process is cumbersome and the detection accuracy is not high.

The application of deep learning (DL) algorithms has grown tremendously in the last few years. Thus, some scholars have proposed methods for applying deep learning frameworks to encrypted data classification [6], [7]. However, most of the current deep learning frameworks are used in the detection and classification of malicious traffic. The method utilizing deep learning models for detection and forensics of encryption data in storage files and network transmission data has not been proposed yet.

In this paper, we propose an efficient end-to-end detection and forensics method based on deep learning called EDNet (Encryption Detection Network). To improve the feature extraction performance, multiscale feature extraction mechanism is used. It utilizes different activation functions to capture different feature responses. Followed, the residual learning mechanism combined with a pooling layer can further enhance the presentation of the features to retain richer information of input data. Last, we use the linear classification layer to calculate the encryption probability of the input sample. The contributions of this paper are as follows:

- 1) The proposed method is the first approach for the detection and forensics of encryption behavior of storage file and network transmission data based on deep learning. It has great reference value for other similar tasks in this field.
- 2) Different from manually extracting features, a multiscale feature extraction mechanism with multiple activation functions and residual learning layers are combined in the EDNet to enhance the ability of feature expression and improve network performance. The proposed EDNet can not only detect different types of data encrypted with different methods but also have a good performance on mixed data. The detection accuracy of EDNet on different types of mixed data is higher than 99%. At the same time, EDNet achieves state-of-the-art performance on storage file and also has a good performance on network transmission data.

The remaining of this paper is organized as follows: In Section II we summarize the relevant research works on encryption detection. In Section III we describe the EDNet model in detail. In Section IV, we explain the experimental

settings and discuss the results. The conclusion is in Section V.

II. RELATED WORK

The traditional approaches of encrypted data detection are to extract and learn the data features which can correctly determine whether the data is encrypted. Jozwiak *et al.* [2] detect encryption by calculating the entropy value of the data. Thurner *et al.* [3] test the stored data using the test statistics proposed by Knuth [8], and set the confidence level. If the value of the test statistical feature is higher than the confidence level, the data will be considered as encrypted data. However, this method has limited performance on compressed data.

Feature selection is an important step in the traditional encryption detection process. Anderson and McGrew [9] have found that feature engineering played a decisive role in identifying malware traffic. Feature extraction [10]–[12] has a much greater impact on the performance of encryption detection. Meng *et al.* [5] propose an encrypted traffic identification method based on randomness estimation. First, 188-dimension randomness features are produced by randomness test and realized dimension reduction by sparse logistical regression based on 1-norm regularization. Then, the encrypted traffic can be identified using the Extreme Learning Machine (ELM). However, the feature dimension selection procedure in this method has a great influence on the result. The method in [4] conducts a randomness test on the test data using the 15 test items published by NIST SP800-22standard [13]. The test first extracts a total of 188-dimensional features and then selects features utilized the greedy algorithm. Finally, the Support Vector Data Description (SVDD) algorithm [14] is used to train and test the model with these selected features. The feature selection operation of the greedy algorithm is a very tedious work that requires at least 375 times of feature selections and a maximum of 17766 times in extreme cases. Thus, the model requires extensive computational time and resources in the training and testing phase. According to our survey, there are few related works for the detection and forensics of encrypted data.

Although the detection and forensic methods of encrypted data have not been fully studied, as a kind of encrypted data detection, the identification of encrypted network traffic has attracted a lot of attention, and many works in this aspect have been proposed recently. Dorfinger *et al.* [15] propose a real-time detection method of Skype encrypted traffic based on the entropy estimation method, but the performance of this method is poor when the encrypted data traffic is small. Moreover, the detection performance of encrypted and unencrypted compressed traffic is not discussed. Anderson and McGrew [16] concentrate on identifying malware communication in encrypted leveraging communication flow context information. Nonetheless, this paper focuses more on the extraction and identification of malicious traffic characteristics, rather than general traffic encryption

detection and identification. The methods in [16], [17] propose a port-based encrypted traffic identification method and believe that current major encryption protocols often have a fixed default port which can help to identify the encryption type. Works [18], [19] consider that encryption protocol data has specific content functions, and propose recognition methods based on content signature recognition. At the same time, the purpose of encryption protocol recognition is achieved by matching content signatures. Some researchers [20]–[23] propose identification methods based on flowing features and believe that the encryption protocol needs to establish a connection to complete version negotiation and key exchange before transmitting encrypted data. At the stage of establishing a connection, the interactive protocol package has a relatively fixed format, content, and specific stream characteristics so that encryption identification can be performed according to these stream characteristics. Unfortunately, the above identification methods based on the port, content signature, and traffic characteristics, can only be implemented for specific encryption protocol identification and the details of the encryption protocol must be known. In fact, in an open network environment, the emergence of private encryption protocols makes it difficult to identify these encrypted network traffic.

Traditional identification methods of encrypted information are mostly based on feature engineering technology. Recently, technologies based on deep learning have continuously produced promising results [25]. Moreover, it is believed that deep learning-based methods are highly desirable approaches for encrypted network traffic classification since it automatically extracts and selects features through training [26]. Wang *et al.* [27] propose an encrypted traffic data classifier based on three deep learning models (multi-layer perceptron MLP, stacked autoencoder [28], and CNN), achieving distributed application perception through classifying the data traffic in smart home networks. Zeng *et al.* [26] propose encrypted traffic classification and intrusion detection using deep learning. The paper uses CNN for spatial feature learning, LSTM [29] for time-domain feature learning, SAE for coding feature learning, and finally combines these three aspects of features to enhance the understanding of the original input data. Ran *et al.* [30] first propose the application of 3D CNN networks for traffic classification. All these deep learning based methods for encrypted traffic classification have achieved good results. Unfortunately, the study applying deep learning to encrypted data detection and forensics has not yet been proposed.

III. EDNet FOR ENCRYPTION DETECTION AND FORENSICS

We proposed a novel deep learning-based method for detection and forensics of encryption behavior called EDNet. In this section, we will illustrate the details of our proposed network. First, the original encrypted binary data will be converted into a two-dimensional matrix which can be viewed as a grayscale image. Second, to extract representative

features from encrypted data, the multiscale feature extraction mechanism is used to enhance the performance of feature extraction. Then, to ensure the depth and learnability of the network, the residual learning and pooling layers are combined to improve the performance of the proposed network. Last, EDNet uses a linear classifier to get the final label of the input data.

A. OVERVIEW

The proposed EDNet framework is shown in Fig. 1. As can be seen that the new framework includes four processing stages: the data pre-processing (highlights in red), the multiscale feature extraction (highlights in green), the residual learning (highlights in yellow), and the linear classifier (highlights in blue). The number of kernels in convolutional layers in each layer of “Inception architecture” is shown in Table 1.

TABLE 1. The number of kernels in each convolutional layer of the “Inception Structure” in the EDNet model.

| Type name | activation | 1x1 | 3x3 | 3x3 | 5x5 | 5x5 | pooling |
|------------|------------|-----|-----|-----|-----|-----|---------|
| InceptionR | Relu | 64 | 32 | 64 | 16 | 32 | 32 |
| InceptionT | Tanh | 64 | 32 | 64 | 16 | 32 | 32 |
| InceptionA | Relu | 128 | 128 | 192 | 32 | 96 | 64 |

The original data to be detected from encrypted files and the network traffic is a binary sequence composed of 0 and 1. Moreover, the data lengths of different sources are different. Therefore, the data to be detected cannot meet the input requirements of a two-dimensional CNN network. In other words, the deep learning framework cannot directly accept raw data as input data. To overcome the input problem, the data pre-processing is used to convert the original input data into a handleable format. To accomplish this goal, EDNet transforms every 8 bytes of binary data into the corresponding decimal numbers and gather them into an asymmetric two-dimensional matrix. In this manner, the data pre-processing phase extracts part of data from the files or the network traffic to be detected, then convert them into 2-D matrixes as the input of deep learning framework which can be seen as a grayscale image as well. For example, when the length of the input data is 16 bytes, each byte will be converted into the corresponding pixel value fixing in a grayscale image with size 4×4 .

Extracting distinguishable features from encrypted data is the focus of traditional detection methods. In the EDNet, a multiscale feature extraction mechanism is realized by the Group 1 layer which contains a convolution operation and Inception modules. After that, a reconstructed feature vector will be obtained as the input of the next neural network layer. Group 1 starts from a 3×3 convolution filter, followed by Batch normalization [31], Relu [32] activation function, and average pooling. To extract rich spatial details and enhance the feature representation, EDNet feeds the pooling output into two parallel Inception structures [33] (displayed as InceptionR and InceptionT in Fig. 1). The difference between

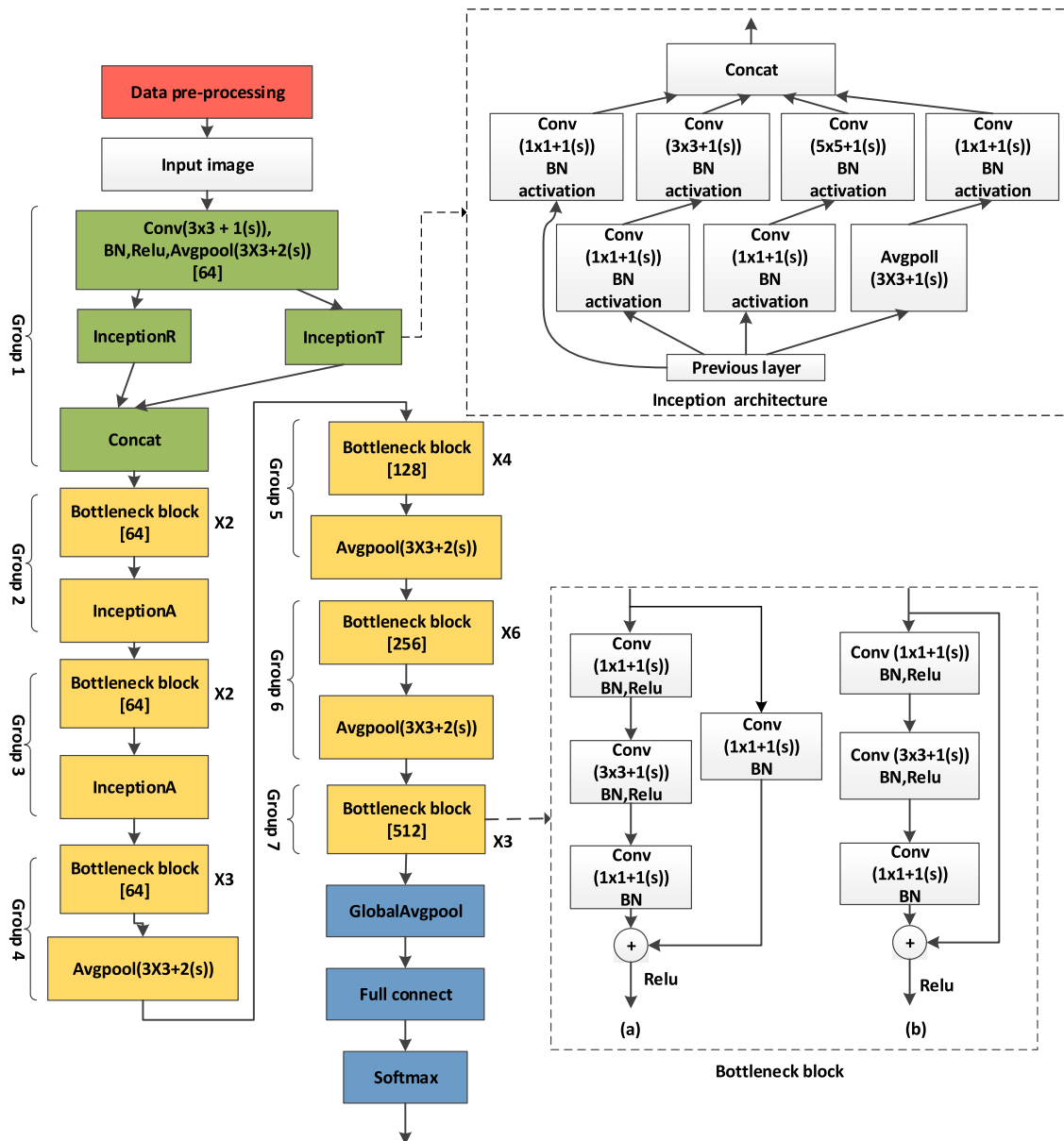


FIGURE 1. The architecture of the proposed EDNet. Conv($a \times a + x_1(s)$) indicates that the kernel size in the convolution layer is $a \times a$, and the stride is x_1 . Avgpool($b \times b + x_2(s)$) indicates that the window size in the average pooling layer is $b \times b$, and the stride is x_2 . Batch normalization is abbreviated as BN. The number “[n]” indicates the number of kernels in the convolution layer. The number “ $\times n$ ” outside the “Bottleneck block” indicates the numbers of Bottleneck blocks stacked.

InceptionR and InceptionT is that they use different activation functions. Since different activation functions respond differently to input data, different feature structures can be retained by two parallel Inception layers.

Next, the residual learning mechanism is introduced into the EDNet to expand the depth of the network while ensuring the learning ability of the network. The feature maps outputted from the multiscale feature extraction layer are concatenated and pass through Group 2-6 to complete the residual learning. Finally, the global averaging pooling layer in Group 7 merges each spatial map into a single element (for example, when the input is assumed to be a grayscale 224×224 image, 1024 feature maps of size 14×14 are

reduced to a 1024-D feature vector by computing the statistical average of each 14×14 feature map).

Last, the linear classifier module is followed to realize the identification of encrypted data. First, the output of the Group1-7 flows through a global average pooling and a full connection layer. Then a softmax function is used to calculate the probability of whether the input data is encrypted.

B. MULTISCALE FEATURE EXTRACTION

As described above, the quality of feature extraction has a great influence on the detection results. The EDNet utilizes a multiscale feature extraction mechanism to enhance the performance of feature representation.

The multiscale feature extraction mechanism includes multiple convolution modules and different activation functions. Specifically, a form of horizontal multiscale convolutional kernels is constructed firstly, so that multiple convolutional kernels of different sizes in the same layer can simultaneously extract features of different scales. Then use different activation functions to capture different feature responses. As shown in Fig. 1 the first half of the EDNet (Group1-Group3) focuses on extracting rich multiscale features and the last half (Group4-Group7) extracts more complex high-level features by multiple convolutional groups.

Activation functions are an important part of the multiscale feature extraction mechanism. Different activation functions have different statistical modeling characteristics, such as the Tanh function, its mathematical expression is shown in (1) below.

$$\text{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (1)$$

Because of its saturated regions, it is possible to limit the range of data values and prevent the subsequent layers from modeling large values. It is generally believed that large values are sparse and their statistical properties are not significant. In [34], a hybrid of Tanh and Relu activations is employed. Tanh is used as the activations in the previous Group1 and Group2 layers, and Relu is used as the activations in the deeper layer. The performance is better than using Relu alone. The mathematical expression of the Relu activation is shown in (2) below.

$$\text{Relu}(x) = \begin{cases} x, & x > 0 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

To learn the different multiscale features of encrypted data, we adopt multiple activations module in Group1, which include InceptionR and InceptionT, using Relu, Tanh function, respectively. And concatenating the resulting feature maps for the subsequent Group 2.

It is expected that each kind of activation unit in the diverse activation module has different responses to the encrypted data. At the same time, it can be seen from Fig. 1 that the multiple activations module is not used in the deeper groups. On one hand, this can avoid increasing the number of parameters in the convolutional layer, on the other hand, since both Tanh and Sigmoid function have saturated regions, the backward propagation of the gradient is difficult when the network is deeper, because of the vanishing gradient phenomenon [35]. So, the multiple activations module is only used in the shallow layer and we choose Relu as the activations for the subsequent convolutional layers.

To intuitively understand the design of multiple activations module in Group 1, Fig. 2 shows the heat feature maps generated by a 224×224 train image through InceptionR, InceptionT, and the data distribution of the feature map. It can be seen from Fig. 2, with different activations, each kind of activation unit has different responses to the input, so that

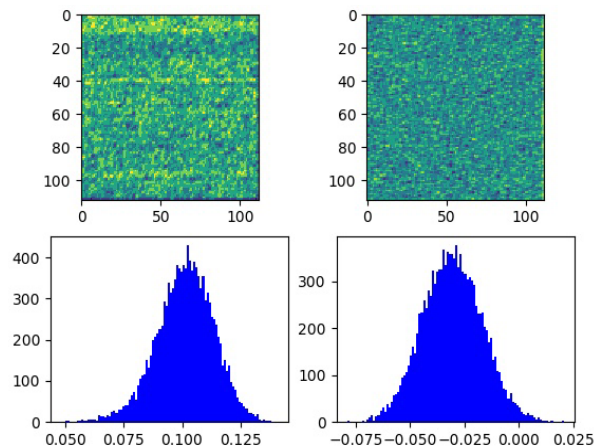


FIGURE 2. Feature maps and distribution. The first row is the heat feature maps generated by a 224×224 train image through InceptionR and InceptionT. The second row is the data distribution of the corresponding feature map.

different features can be obtained. The mean value activated by Relu is around 0.1, while by Tanh it is around -0.03 .

C. RESIDUAL LEARNING

Generally speaking, the deep network has a stronger ability to extract features. In other words, more representative features can be learned as the network deepens. To this end, aiming at increasing the depth of the network, EDNet employs a deep residual learning strategy to capture the important global contextual information. Specifically, a “Bottleneck block” architecture with the shortcut is used. As illustrated in Fig. 1, Group 2-7 consist of Bottleneck block and Inception. Moreover, the average pooling is utilized for dimensionality reduction.

The “Bottleneck block” architecture in Group 2-7 is shown at the bottom of Fig. 1. As can be seen from the figure, each residual function is constructed by a stack of 3 convolutional layers. The three layers are 1×1 , 3×3 , and 1×1 convolutions, where the number of the first 1×1 kernels and the followed 3×3 kernels are shown in the figure. Moreover, the expansion of the “Bottleneck block” is set to 2, which means after the calculation of the residual function, the number of output dimensions will be expanded twice as the input dimensions. The identity shortcuts can be directly used when the input and output are of the same dimensions (shown as structure (b) in Fig.1). Otherwise, we consider using 1×1 convolutions to match the dimensions (shown as structure (a) in Fig.1).

Next, an average pooling layer follows the residual function. In Image classification, image is classified according to image objects, and may only be related to some local regions, but encryption detection is related to the content of the entire image. Convolution with a stride of 2 will weaken some local features while strengthening a certain local feature. On the contrary, average pooling can better preserve the feature correlation by averaging the adjacent pixels, thus, the average pooling layer is used in EDNet.

IV. EXPERIMENT

In this section, we will introduce the experimental dataset, experimental settings, and training methods. Moreover, we also expand on performance analysis and comparison of the EDNet.

A. DATASETS AND PLATFORM ENVIRONMENTS

In this paper, to evaluate the performance of the proposed method, we establish two datasets including dataset1 (DS1) and dataset2 (DS2). The DS1 consists of 1335133 grayscale images of size 224×224 , which are generated from 353 randomly downloaded videos. Then, we use the DS1 to pre-train our model. As analyzed previously in the introduction section, locally stored data and network traffic data are the two encryption detection scenarios. The DS2 is divided accordingly into these two types. For locally stored data, we first download 12000 audio, 12000 video, and 10000 text files whose size is more than 1MB from the Internet, covering a variety of data formats, as shown in Table 2. For each file, we take a data piece in the middle of a certain length (e.g. 49KB) and convert it into a two-dimensional gray image through the data pre-processing module. Note that only one such image will be extracted from each file and all images will finally be collected to form the original video, audio, text dataset.

TABLE 2. The Stored data sets.

| Data types | | Number of data per type |
|------------|------|-------------------------|
| Video | MP4 | 3000 |
| | AVI | 3000 |
| | MOV | 3000 |
| | MKV | 3000 |
| Audio | MP3 | 3000 |
| | WMA | 3000 |
| | AAC | 3000 |
| | WAV | 3000 |
| Text | TXT | 3000 |
| | WORD | 3000 |
| | PDF | 3000 |
| | PPT | 3000 |
| | HTML | 3000 |

For network traffic data, we divide it into internet surfing traffic and FTP transmission traffic. Within the internet surfing traffic, the encrypted data stream is generated by the encryption proxy software VPN, while the unencrypted traffic is the network traffic that is generated while browsing through a browser. Within the FTP transmission traffic, the non-encrypted traffic is captured when the above-mentioned audio, video, and text files are transmitted through FTP, while the encrypted traffic is captured when the AES encrypted audio, video, and text files are transmitted. All the above-mentioned network traffic data are captured by wire shark software. Similarly, for each of these four traffic types, we select 32000 segments of the fixed length of 49KB

and convert them into 224×224 size 2D grayscale to get the final Internet traffic dataset, which is shown in Table 3.

TABLE 3. Internet traffic data sets.

| Data types | | Image number |
|--------------------------|-------------------|--------------|
| Web Browsing Traffic | VPN encrypted | 32000 |
| | Unencrypted | 32000 |
| FTP Transmission Traffic | FTP encrypted | 32000 |
| | FTP non-encrypted | 32000 |

All the experiments in this paper are based on the PyTorch framework. The hardware platform is a server equipped with an NVIDIA RTX 2080TI graphics card. This GPU has 11GB on-chip memory and its single-precision floating-point computing capability can reach 13.45 TFLOPS.

B. EDNet TRAINING

Transfer learning [36] is widely used in many deep learning tasks. It can further improve the learning capability of a deep learning model. To get better initial parameters for our task, the EDNet model is pre-trained with the DS1. Then, we fine-tune it by applying the pre-trained model into our encrypted data detection task. Moreover, to prevent overfitting, we adopt the RandorizionalFlip data enhancement strategy to enrich our DS2.

In the pre-training stage, we select Adamax [37] as the optimizer with a minibatch of 16. The weight of filters is initialized by He Kaiming initializer [38] and regularized with L2 regularization of $2e-4$. The fully connected layer is initialized by a standard normal distribution. We divide the DS1 into the training, validation, and testing set with a ratio of 7.5:1:1.5. A validation operation will be performed on a validation set every 50000 iterations. Moreover, to avoid the model falling into local optimum, a learning decay strategy is applied when the accuracy on the validation dataset no longer rises.

After the pre-trained model is obtained, we will train our model on DS2, which is divided into three parts: training set, validation set, and test set. The proposed method needs large amounts of data for training. And the training process is time-consuming. During the fine-tuning, the number of the epoch is limited to 100, and validation is conducted on the validation set every 1000 iterations. The initial learning rate is set to 0.0001 and reduced to 0.00005 after 20 epochs.

C. ABLATION STUDY

A wider network can capture richer spatial information in the early stage of the network. The EDNet uses Inception to increase the width of the network. To be specific, in Group1, the Inception is used to widen the network with different activation functions, which can extract multiscale features. Furthermore, InceptionA in Group2 and Group3 can further enhance the representation of features. Besides, the residual learning mechanism avoids the problem of gradient disappearance or gradient explosion while the network deepens.

To analyze the impact of different function modules, we conduct a series of ablation studies with different network architecture. 1) Using a 3×3 convolutional layer replaces the Inception module in the Group1, named “No_inception”. 2) Only use the InceptionR in the Gtoup1 called “only InceptionR”. 2) Only use the InceptionT in the Group1 called “only InceptionT”. 3) Using InceptionR and InceptionT without pooling operation in the Group 4-7, termed “no pooling”. The comparison results of different network architectures are illustrated in Table 4.

TABLE 4. Detection accuracy of each model on video test dataset.

| Model | Accuracy(test) |
|-----------------|----------------|
| EDNet | 98.83% |
| only InceptionR | 98.70% |
| only InceptionT | 98.77% |
| no_pooling | 98.52% |
| no_inception | 98.62% |

From Table 4, it is obvious that the EDNet has the highest accuracy on the test video datasets. Besides, without the pooling layer results in a terrible decrease in performance. Which means the average pooling layer can extract the representative features from the upper input. Moreover, compared with the network without the inception module, the EDNet improve the accuracy of the network by 0.21%. This means the employment of the multiscale feature extraction mechanism plays a positive role in improving the results.

D. PERFORMANCE ANALYSIS

In this section, the length of data pieces in all experiments is set to 49KB except Experiment 2, and the corresponding image size is 224×224 . Note that all stored data are encrypted with AED except experiment 1.

According to the previously stated relevant researches in Section II, works [2]–[5] are most closely related to research in this paper. References [2] and [4] are relatively common methods for detecting encrypted data, [3] is a specific detection method for stored data, and [5] is an encrypted network traffic detection method. Therefore, when analyzing the performance of our model, we compare our EDNet model with the methods [2]–[4] on stored data, and with methods [2], [4], [5] on network traffic. The corresponding experimental settings are set following the conditions of the best result in these methods.

1) THE DETECTING PERFORMANCE OF DIFFERENT ENCRYPTION ALGORITHMS ON DIFFERENT TYPES OF DATA

Since the random statistical properties of encrypted data are affected by the encryption algorithm, to evaluate the robustness of the EDNet model, we choose AES and DES to encrypt different types of data. The experimental results are shown in Table 5.

TABLE 5. Encryption detection accuracy of current methods and EDNet.

| Encryption algorithm | Data type | paper[2] | paper[3] | pape[4] | EDNet |
|----------------------|-----------|----------|----------|-------------|---------------|
| AES | Video | 50.80% | 82.58% | 94.3% | 98.83% |
| | Audio | 68.95% | 93.94% | 99.96% | 100% |
| | Text | 90.02% | 92.03% | 100% | 100% |
| | Mean | 69.92% | 89.52% | 98.09% | 99.61% |
| DES | Video | 50.67% | 82.03% | 92.31% | 98.75% |
| | Audio | 62.35% | 81.25% | 91.48% | 99.90% |
| | Text | 88.76% | 86.9% | 99.93% | 99.98% |
| | Mean | 67.26% | 83.39% | 94.57% | 99.54% |

As illustrated in Table 5, the encryption detection of video data is more difficult than other data. For the video dataset, the EDNet model offered improvements of 48.03%, 16.25%, and 4.53% compared with the methods in paper [2], [3], and [4] respectively when encrypted with AES. Besides a similar value of 48.08%, 16.72%, 6.44% are offered when encrypted with DES. Methods in [2]–[4] are essentially based on the detection of randomness testing value to judge whether the data is encrypted. The randomness of data varies when encrypted with different algorithms. Thus, different encryption algorithms would result in a great impact on detection accuracy. Compared with AES encrypted data, DES-encrypted data has poor randomness, so the encryption detection accuracy of DES-encrypted data in works [2]–[4] is poor. In contrast, the EDNet showed better performance on both encryption algorithms because it can automatically learn the representative feature for encrypted data. Last, from the mean value, the EDNet has the highest accuracy on the dataset generated by different encryption algorithms. Moreover, the superiority of the EDNet model shows its inherently good robustness.

2) EVALUATE THE IMPACT OF DATA LENGTH ON DETECTING PERFORMANCE

The length of the data can greatly affect the randomness test value. At the same time, the different image resolutions generated by different data lengths can affect the performance of the neural network. Therefore, we conduct experiments on three different lengths of all datasets to evaluate the effect of data length on existing methods and the proposed EDNet. The experimental results are shown in Table 6.

It is easy to be seen that generally speaking, the detection accuracy gets higher as the data length gets longer or the image size gets larger. Despite the EDNet has the highest mean accuracy on the datasets of all different lengths, its performance on a few datasets is slightly behind the comparison algorithms. Note that when the data length is 5.06 KB, the detection accuracy of EDNet on the video and audio datasets is 0.08% and 0.02% lower than that of paper [4], respectively. What’s more, a lower accuracy on the text dataset can also be seen when the data length is 10.06 KB.

TABLE 6. Detection accuracy of current methods and EDNet on different data length.

| Data length (image size) | Method | Video | Audio | Text | Mean |
|--------------------------|----------|---------------|---------------|---------------|---------------|
| 5.06KB (72x72) | paper[2] | 50.98% | 68.40% | 90.00% | 69.79% |
| | paper[3] | 75.50% | 90.58% | 91.75% | 85.94% |
| | paper[4] | 84.58% | 99.96% | 99.30% | 94.61% |
| | EDNet | 84.50% | 99.94% | 99.40% | 94.61% |
| 10.16KB (102x102) | paper[2] | 50.80% | 68.55% | 90.02% | 69.79% |
| | paper[3] | 75.00% | 91.63% | 91.78% | 86.14% |
| | paper[4] | 86.10% | 100% | 99.68% | 95.26% |
| | EDNet | 91.21% | 100% | 99.65% | 96.95% |
| 49KB (224x224) | paper[2] | 50.80% | 68.95% | 90.02% | 69.92% |
| | paper[3] | 82.58% | 93.94% | 92.03% | 89.52% |
| | paper[4] | 94.30% | 99.96% | 100% | 98.09% |
| | EDNet | 98.81% | 100% | 100% | 99.60% |

The reason behind this is that, in the experiment, the parameter in the paper [4] method is selected to get optimal performance for each dataset, while the parameter of EDNet model is just a global optimal one, which is used on all dataset.

When the data length is 5.06 KB, the detection accuracy of the EDNet model is 33.52%, 9%, and -0.08% higher than that of the paper [2], [3], and [4] respectively. on the video dataset, 31.54%, 9.36%, and 0.02% on the audio dataset and 9.4%, 7.65%, and 0.1% on the text dataset. In general, the EDNet model is superior in detection performance regardless of the data length.

3) EVALUATE THE VERSATILITY OF EDNet MODEL FOR VARIOUS DATA TYPES

To prove the versatility of the EDNet model to various data types, we conducted blind detection experiments by mixing video, audio, and text datasets in different combinations. The experimental results are shown in Table 7.

TABLE 7. Accuracy of blind detecting with current methods and EDNet.

| Data types | paper[2] | paper[3] | pape[4] | EDNet |
|------------------------------|----------|----------|---------|---------------|
| Mix of video, audio and text | 68.73% | 89.68% | 97.57% | 99.17% |
| Mix of video and audio | 59.87% | 88.26% | 96.57% | 99.18% |
| Mix of video and text | 68.62% | 86.88% | 97.83% | 99.11% |
| Mix of audio and text | 78.53% | 93.07% | 99.98% | 99.99% |
| Mean | 68.94% | 89.47% | 97.99% | 99.36% |

The experimental results demonstrate that the EDNet method performs better than the comparison algorithm in different types of mixed data. Furthermore, from the mean point

of view, compared with the sub-optimal method, the accuracy of the EDNet is improved by nearly 2%. Meanwhile, it can be seen from the experimental results that the detection of video data is more difficult, and the detection accuracy of different types of mixed datasets which contain video data is lower than that of datasets without video data. Overall, the accuracy of the EDNet method on different types of datasets can reach more than 99%, which is a satisfactory result.

4) EVALUATE THE DETECTING PERFORMANCE OF EDNet MODEL ON ENCRYPTED INTERNET TRAFFIC

The above experiments prove the effectiveness of the EDNet model for storing data. This experiment focuses on the applicability of the EDNet model to data generated during network transmission.

The test set used in this section includes web browsing traffic and FTP traffic. In the web browsing traffic, the encrypted data flow is generated by the encrypted VPN, and the unencrypted traffic is the network traffic generated by the normal use of the browser. In FTP traffic, non-encrypted traffic is traffic captured when audio, video, and text files are transferred via FTP. Encrypted traffic is the traffic captured when audio, video, and text files are encrypted using AES encryption and transmitted.

The experimental results are described in Table 8. It is indicated that even for network traffic, the EDNet model still shows a better performance than the existing methods. Especially on FTP data, the accuracy of the EDNet is significantly better than the comparison algorithms. Furthermore, in the mean value, the accuracy of the new method is also in a leading position. This means that in the tested mixed network traffic, EDNet is stronger than other methods in expressing representative features. Although EDNet is slightly weaker than methods in paper [4], [5] in the accuracy of network traffic detection, the difference is very small and acceptable. In summary, EDNet not only improves the detection accuracy of stored files but also has better performance on network transmission Data.

TABLE 8. Detection accuracy of current methods and EDNet on internet traffic.

| Data types | paper[2] | paper[4] | pape[5] | EDNet |
|----------------------|----------|----------|---------------|---------------|
| Web browsing traffic | 68.40% | 99.83% | 99.99% | 99.96% |
| FTP traffic | 52.91% | 95.11% | 92.23% | 98.80% |
| Mean | 60.66% | 97.47% | 96.11% | 99.38% |

V. CONCLUSION

In this paper, a deep learning-based method for the detection and forensics of encryption behavior of storage file and network transmission data is proposed. The newly proposed method is called EDNet, which first maps the data into a two-dimensional gray image to solve the input problem of

encrypted data in the neural network. Then, EDNet applies different activation functions that respond differently to different upper-layer inputs to complete the multiscale feature extraction operation. Then the residual learning mechanism and the average pooling layer further improve the learning ability of the new network. Finally, use the linear classification layer to assign labels to the input data.

The effectiveness of network architecture has been proved by the ablation study. Further, the experiment indicated that the detection accuracy of EDNet on different lengths of data and mixed encrypted data is outstanding compare to the competitive methods, and the detection performance under different encryption algorithms is also very superior. Moreover, EDNet has also achieved satisfactory results in detecting whether the network traffic data is encrypted. In future work, we will consider designing an encryption detection network by combining design ideas of other great classification networks such as DenseNet [39].

REFERENCES

- [1] Y. Pan, "Research on network database encryption technology," in *Proc. IEEE 3rd Int. Conf. Commun. Softw. Netw. (ICCSN)*, May 2011, pp. 690–693.
- [2] I. Jozwiak, M. Kedziora, and A. Melinska, "Theoretical and practical aspects of encrypted containers detection—Digital forensics approach," in *Dependable Computer Systems (Advances in Intelligent and Soft Computing)*, vol. 97. Berlin, Germany: Springer, 2011, pp. 75–85.
- [3] S. Thurner, M. Grun, S. Schmitt, and H. Baier, "Improving the detection of encrypted data on storage devices," in *Proc. 9th Int. Conf. IT Secur. Incident Manage. IT Forensics*, May 2015, pp. 26–39.
- [4] J. Meng, Y. Zhou, and Z. Pan, "Detection of encrypted data based on support vector data description," in *Proc. Int. Conf. Adv. Cloud Big Data*, Dec. 2013, pp. 187–191.
- [5] J. Meng, L. Yang, Y. Zhou, and Z. Pan, "Encrypted traffic identification based on sparse logistical regression and extreme learning machine," in *Proceedings of ELM-2014*. Cham, Switzerland: Springer, 2015, pp. 61–70.
- [6] S. Rezaei and X. Liu, "Deep learning for encrypted traffic classification: An overview," *IEEE Commun. Mag.*, vol. 57, no. 5, pp. 76–81, May 2019.
- [7] E. Hesamifard, H. Takabi, and M. Ghasemi, "Deep neural networks classification over encrypted data," in *Proc. 9th ACM Conf. Data Appl. Secur. Privacy*, Mar. 2019, pp. 97–108.
- [8] D. E. Knuth, "Son of seminumerical algorithms," *ACM SIGSAM Bull.*, vol. 9, no. 4, pp. 10–11, Nov. 1975.
- [9] B. Anderson and D. McGrew, "Machine learning for encrypted malware traffic classification: Accounting for noisy labels and non-stationarity," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 1723–1732.
- [10] F. Luo, L. Zhang, B. Du, and L. Zhang, "Dimensionality reduction with enhanced hybrid-graph discriminant learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 8, pp. 5336–5353, Aug. 2020.
- [11] L. Zhang, Q. Zhang, L. Zhang, D. Tao, X. Huang, and B. Du, "Ensemble manifold regularized sparse low-rank approximation for multiview feature embedding," *Pattern Recognit.*, vol. 48, no. 10, pp. 3102–3112, Oct. 2015.
- [12] Z. Liu, Z. Lai, W. Ou, K. Zhang, and R. Zheng, "Structured optimal graph based sparse feature extraction for semi-supervised learning," *Signal Process.*, vol. 170, May 2020, Art. no. 107456.
- [13] J. Sobotík and V. Plátěnka, "A statistical test suite for random and pseudorandom number generators for cryptographic application," in *Proc. 2nd Conf. Secur. Protection Inf. (SPI)*, 2003, pp. 161–168.
- [14] A. Banerjee, P. Burlina, and R. Meth, "Fast hyperspectral anomaly detection via SVDD," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, vol. 4, Oct. 2007, pp. IV-101–IV-104.
- [15] P. Dorfinger, G. Panholzer, B. Trammell, and T. Pepe, "Entropy-based traffic filtering to support real-time skype detection," in *Proc. 6th Int. Wireless Commun. Mobile Comput. Conf. ZZZ (IWCMC)*, 2010, pp. 747–751.
- [16] B. Anderson and D. McGrew, "Identifying encrypted malware traffic with contextual flow data," in *Proc. ACM Workshop Artif. Intell. Secur. (ALSec)*, 2016, pp. 35–46.
- [17] R. Alshammari and N. Zincir-Heywood, "Generalization of signatures for SSH encrypted traffic identification," in *Proc. IEEE Symp. Comput. Intell. Cyber Secur.*, Nashville, TN, USA, Mar. 2009, pp. 167–174.
- [18] T. Karagiannis, A. Broido, M. Faloutsos, and K. Claffy, "Transport layer identification of P2P traffic," in *Proc. 4th ACM SIGCOMM Conf. Internet Meas. (IMC)*, 2004, pp. 121–134.
- [19] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: Multilevel traffic classification in the dark," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun. (SIGCOMM)*, 2005, p. 229.
- [20] H.-J. Kang, M.-S. Kim, and J. W. Hong, "Streaming media and multimedia conferencing traffic analysis using payload examination," *ETRI J.*, vol. 26, no. 3, pp. 203–217, Jun. 2004.
- [21] A. McGregor, M. Hall, P. Lorier, and J. Brunskill, "Flow clustering using machine learning techniques," in *Proc. Int. Workshop Passive Act. Netw. Meas.*, 2004, pp. 205–214.
- [22] L. Bernaille, R. Teixeira, and K. Salamatian, "Early application identification," in *Proc. ACM CoNEXT Conf. (CoNEXT)*, 2006, p. 1.
- [23] C. Bacquet, K. Gumus, D. Tizer, A. N. Zincir-Heywood, and M. I. Heywood, "A comparison of unsupervised learning techniques for encrypted traffic identification," *J. Inf. Assurance Secur.*, vol. 5, no. 1, pp. 464–472, 2010.
- [24] G. Maiolini, A. Baiocchi, A. Iacovazzi, and A. Rizzi, "Real time identification of SSH encrypted application flows by using cluster analysis techniques," in *Proc. Int. Conf. Res. Netw.*, in Lecture Notes in Computer Science, vol. 5550, 2009, pp. 182–194.
- [25] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
- [26] Y. Zeng, H. Gu, W. Wei, and Y. Guo, "Deep-Full-Range: A deep learning based network encrypted traffic classification and intrusion detection framework," *IEEE Access*, vol. 7, pp. 45182–45190, 2019.
- [27] P. Wang, F. Ye, X. Chen, and Y. Qian, "Datanet: Deep learning based encrypted network traffic classification in sdn home gateway," *IEEE Access*, vol. 6, pp. 55380–55391, 2018.
- [28] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, A. P. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, 2010.
- [29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [30] J. Ran, Y. Chen, and S. Li, "Three-dimensional convolutional neural network based traffic classification for wireless communications," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Nov. 2018, pp. 624–627.
- [31] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: <http://arxiv.org/abs/1502.03167>
- [32] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Int. Conf. Mach. Learn.*, 2010, pp. 807–814.
- [33] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
- [34] G. Xu, H.-Z. Wu, and Y.-Q. Shi, "Structural design of convolutional neural networks for steganalysis," *IEEE Signal Process. Lett.*, vol. 23, no. 5, pp. 708–712, May 2016.
- [35] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.
- [36] S. Jialin Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.
- [37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [38] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.
- [39] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.



research interests include machine learning, multimedia signal processing, and information forensics.

SONGBIN LI received the Ph.D. degree from the Institute of Acoustics, Chinese Academy of Sciences, Beijing, China, in 2010. He was a Postdoctoral Fellow and a Visiting Professor with Tsinghua University and the University of Southern California. He has been a Professor with the Institute of Acoustics, Chinese Academy of Sciences, since 2018. He has been the Principle Investigator of several projects of the National Natural Science Foundation of China. His current



PENG LIU received the B.S. degree in communication engineering from Hainan University, in 2011, and the Ph.D. degree from the Institute of Acoustics, Chinese Academy of Sciences, Beijing, China, in 2016. He has been an Associate Professor with the Chinese Academy of Sciences, since 2018. His current research interests include information forensics, multimedia signal processing, and information forensics.

...