applied sciences

MDPI

Article

FF-MR: A DoH-Encrypted DNS Covert Channel Detection Method Based on Feature Fusion

Yongjie Wang [1,2], Chuanxin Shen [1,2,*], Dongdong Hou [1,2], Xinli Xiong [1,2] and Yang Li [1,2]

[1] College of Electronic Engineering, National University of Defense Technology, Hefei 230037, China
[2] Anhui Province Key Laboratory of Cyberspace Security Situation Awareness and Evaluation, Hefei 230037, China
* Correspondence: shenchuanxin@nudt.edu.cn

Abstract: In this paper, in order to accurately detect Domain Name System (DNS) covert channels based on DNS over HTTPS (DoH) encryption and to solve the problems of weak single-feature differentiation and poor performance in the existing detection methods, we have designed a DoH-encrypted DNS covert channel detection method based on features fusion, called FF-MR. FF-MR is based on a Multi-Head Attention and Residual Neural Network. It fuses session statistical features with multi-channel session byte sequence features. Some important features that play a key role in the detection task are screened out of the fused features through the calculation of the Multi-Head Attention mechanism. Finally, a Multi-Layer Perceptron (MLP) is used to detect encrypted DNS covert channels. By considering both global and focused features, the main idea of FF-MR is that the degree of correlation between each feature and all other features is expressed as an attention weight. Thus, features are re-represented as the result of the weighted fusion of all features using the Multi-Head Attention mechanism. Focusing on certain important features according to the distribution of attention weights improves the detection performance. While detecting the traffic in encrypted DNS covert channels, FF-MR can also accurately identify encrypted traffic generated by the three DNS covert channel tools. Experiments on the CIRA-CIC-DoHBrw-2020 dataset show that the macro-averaging recall and precision of the FF-MR method reach 99.73% and 99.72%, respectively, and the macro-averaging F1-Score reached 0.9978, which is up to 4.56% higher than the existing methods compared in the paper. FF-MR achieves at most an 11.32% improvement in macro-averaging F1-Score in identifying three encrypted DNS covert channels, indicating that FF-MR has a strong ability to detect and identify DoH-encrypted DNS covert channels.

Keywords: DNS over HTTPS; DNS covert channel; features fusion; multi-head attention mechanism

check for updates

Citation: Wang, Y.; Shen, C.; Hou, D.; Xiong, X.; Li, Y. FF-MR: A DoH-Encrypted DNS Covert Channel Detection Method Based on Feature Fusion. *Appl. Sci.* **2022**, *12*, 12644. https://doi.org/10.3390/app122412644

Academic Editors: Guang Cheng, Shui Yu and Yongning Tang

Received: 25 October 2022
Accepted: 6 December 2022
Published: 9 December 2022

1. Introduction

As a critical infrastructure of the Internet, the DNS protocol plays an important role in the translation between domain names and IP addresses. However, DNS requests and responses are transmitted in the form of plaintext, which means that anyone can intercept and view the network access behavior of users between the host and local DNS server, which is detrimental to the protection of user privacy and network security [1]. Therefore, the technology to protect the security of DNS requests comes into being. At present, the three protocols listed in the Internet standardization document include Domain Name System Security Extensions (DNSSEC), DNS over TLS (DoT), and DNS over HTTPS (DoH). DNSSEC mainly uses digital signature technology to protect the integrity and authenticity of the DNS response, but the communication process is still transparent to attackers. DoT and DoH both use TLS encryption, the difference between them being that the former uses the dedicated port 853, while the latter uses the HTTPS standard port 443, i.e., DoH transfers DNS messages using HTTPS streams. Today, companies, such as Google, Cloudflare, and Alibaba, offer DoH nodes, and Google's Chrome browser natively supports DoH. In February 2020, Mozilla Firefox began enabling DoH by default for US users, and DNS requests from Firefox were encrypted by DoH and forwarded to Cloudflare or [2]. DoH has played an effective role in protecting customers' privacy and network security and has thus developed rapidly and is more widely used in practical applications.

With the development of 5G technology, Internet of Things (IoT) systems are gaining momentum. While they result in greater convenience, their use presents many security challenges, e.g., data privacy, unstable network connections, and possible botnets [3]. The DNS system, as the underlying network, affects the reliability and security of the IoT [4]. DNS technology meets the availability and transparency requirements for deploying the IoT, a large number of heterogeneous devices can use DNS for network access, and DNS encryption technology better protects the DNS data privacy of users. However, the risks and opportunities are the same: the encryption of DoH and the interactivity of end-to-end devices in the IoT (without human involvement) may jointly lead to attacks that are less likely to be detected, not to mention the emergence of large-scale botnets and DDoS attacks on public IoT services [5].

Therefore, while DoH enhances security, it also provides new opportunities for attackers. In July 2019, Netlab, the cyber threat search division of Qihoo 360, released a report that malware named Godlua used DoH to obtain domains and use them as communication channels for Command and Control (C&C) [6]. In May 2020, Kaspersky also discovered that Iran's APT group OilRig had weaponized DoH and applied it to actual network data theft activities [7]. General, non-encrypted DNS covert channels are transmitted in plaintext via port 53 in the C&C stage of Advanced Persistent Threats (APT), while a DoH uses encrypted transmission via port 443, which is indistinguishable from general HTTPS traffic for network administrators. An attacker can thus hide and encrypt the DNS covert channel via DoH to conduct malicious cyber attacks.

In this paper, we propose a DoH-encrypted DNS covert channel detection method called FF-MR that aims to improve detection performance and solve the problems of weak single-feature differentiation and poor performance in existing research. In summary, the contributions of our paper are three-fold:

- We summarize and analyze the threat scenario of DoH-encrypted DNS covert channels in the C&C stage, clarify its communication principle, and provide support for the research of detection methods.
- We propose a DoH-encrypted DNS covert channel detection method (FF-MR) based on feature fusion. FF-MR takes the session as a representation of encrypted DNS covert channel traffic, fuses statistical features with byte sequence features extracted by Residual Neural Networks, and focuses on important features through a Multi-Head Attention mechanism to detect and identify three encrypted DNS covert channels.
- We conduct comprehensive experiments to evaluate the performance of FF-MR by comparing it with other encrypted DNS covert channel detection methods. We establish four baselines to measure the improvements achieved by the detection model in FF-MR and verify the validity of the model. Finally, recommended values for the hyperparameters are identified using a parameter sensitivity experiment.

The rest of the paper is organized as follows: Section 2 introduces related work on DoH-encrypted DNS covert channel detection. Section 3 summarizes and analyzes the command control process of DoH-encrypted DNS covert channel and introduces the research and application of Multi-Head Attention. In Section 4, we present the design details of FF-MR. In Section 5, we experimentally evaluate the performances of FF-MR on the publicly available dataset CIRA-CIC-DoHBrw-2020. Section 6 concludes this paper.

2. Related Work

In this paper, we mainly present the existing research on DoH-encrypted DNS covert channel detection. Most previous studies have used statistical features and the CIRA-CIC-DoHBrw-2020 dataset in their experiments. Detection is basically performed in two layers: the first layer classifies non_DoH (HTTPS traffic) with DoH traffic, and the second layer classifies DoH traffic into normal DoH and malicious DoH traffic, i.e., DoH-encrypted DNS covert channel traffic, as shown in Table 1.

Table 1. Research on DoH-encrypted DNS covert channel detection and identification.

Research Category	Publication Year	Author	Features/Neural Network Input	Method
Detection	2020	Banadaki et al. [8]	Statistical Features	LGBM, Random Forest
	2020	MontazeriShatoori et al. [9]	Statistical Features	Random Forest, Naive Bayes, SVM, LSTM
	2021	Al-Fawa'reh [10]	Statistical Features	Bi-RNN
	2022	Nguyen et al. [11]	Statistical Features	Transformer
	2022	Zhan et al. [12]	Statistical Features +TLS fingerprint	Decision tree, Random Forest, Logistic Regression
Detection and Identification	2021	Mitsuhashi et al. [13]	Statistical Features	LGBM, XGBoost
	2022	Zebin et al. [14]	Statistical Features	Stacked Random Forest

Banadaki et al. [8] performed a statistical analysis of DoH traffic and extracted a total of 34 classes of statistical features, including IPs and ports, and used machine learning algorithms such as LGBM and XGBoost to perform two-level classification. However, the source IPs and destination IPs, which are directly related to the data itself, were used as the basis for classification, and the resulting experimental results were obviously not objective. MontazeriShatoori et al. [9] proposed arranging the captured packets in temporal order. A set of consecutive packets in the same direction within a certain time threshold is called a packet cluster. In this study, 28 classes of statistical features were extracted, and traditional machine learning algorithms, including Random Forest (RF), Naive Bayes (NB), Support Vector Machines (SVM), and Long Short-Term Memory (LSTM) Neural Networks, were used to distinguish non_DoH from DoH and normal DoH from malicious DoH traffic. Nguyen et al. [11] proposed a two-layer classification of DoH based on a Transformer containing a four-layer encoder and a six-layer decoder using statistical features as input. They also used an ELK stack architecture, which included four modules: Elasticsearch, Logstash, Kibana, and Beats. Finally, a Security Operation Center (SOC) system enabled the monitoring and detection of malicious DoH traffic for enterprise-level networks.

For binary classification, Al-Fawa'reh [10] combined statistical feature analysis with a Bidirectional Recurrent Neural Network (Bi-RNN) to achieve the detection of DoH-encrypted DNS covert channels. Zhan et al. [12] established a TLS fingerprint whitelist based on the information from the TLS handshake stage, where TLS fingerprints not on the whitelist will be considered as suspicious DoH traffic; in addition, normal DoH and encrypted DNS covert channels are classified according to the statistical features of the DoH flow. The difference between Al-Fawa'reh's work and other studies is that the attack scenarios are simulated; data on the location (latency), number, sending interval (rate), and packet (domain name) length of different DoH servers are generated; and a large number of adversarial and evaluation experiments are conducted to verify the effectiveness of the method using three machine learning models.

The above studies only investigate the detection of encrypted DNS covert channels, but there is less work related to identification of DoH-encrypted DNS covert channels in the literature. Zebin et al. [14] proposed an interpretable machine learning approach using ten-fold cross-validation to triple classify HTTPS, normal DoH, and DoH-encrypted DNS covert channel traffic by stacking RF-based classifiers and, finally, test the performance of the model in identifying encrypted DNS covert channels. The model could only achieve an accuracy of about 92%. Mitsuhashi et al. [13] chose three machine learning algorithms, XGBoost, LightGBM, and CatBoost, to implement a three-stage detection of HTTPS and DoH, normal DoH, and DoH-encrypted DNS covert channel traffic, respectively, followed by the classification and identification of the encrypted DNS covert channels.

By summarizing the existing research on DoH-encrypted DNS covert channel detection, we identify the following limitations:

- There are few works related to DoH-encrypted DNS covert channel detection and identification in existing studies, and the performance in this area still needs to be improved;
- Most existing studies use statistical features as the basis for detection, which makes it easy for attackers to evade detection by using single features;
- The role of byte sequence features and combining multiple features is ignored, thus failing to meet the requirements of encrypted DNS covert channel detection.

In summary, there is a lack of work related to DoH-encrypted DNS covert channel identification in existing studies, and the detection performance still needs to be improved. Therefore, in this paper, we focus on the detection and identification of DoH-encrypted DNS covert channels and calculate the correlation between statistical features and session byte sequence features globally through Multi-Head Attention to obtain weighted fusion features as a basis for detection and identification. As the correlation between global features is extracted and the key features are highlighted, detection and identification performance have been further improved.

3. Background

We elaborate and analyze the mechanism of the DoH-encrypted DNS covert channel in Section 3.1 and formally describe the principle of the Multi-Head Attention mechanism in Section 3.2.

3.1. DoH-Encrypted DNS Covert Channel

In this paper, we focus on DoH-encrypted DNS covert channels and domain name resolution using DoH in two cases: one is to use browsers such as Google, Firefox, etc., that support the DoH protocol, where all DNS traffic is directly encapsulated into a TLS encrypted HTTP message and sent to DoH servers, which are then forwarded to domain name servers on the Internet for resolution; the second is to use hosts that do not support the DoH protocol by building a local DoH proxy for forwarding (available proxy tools include QuantumultX, Surge, Loon, etc.). The host forwards all network requests to the local DoH proxy, the proxy will forward the DoH traffic to the Internet DoH server, and, finally, the DoH server will perform domain name resolution.

As shown in Figure 1, in the C&C stage of an APT attack, the data carrier in the DoH-encrypted DNS covert channel does not use the DNS covert channel directly, but rather, the attack is implemented by encapsulating it into a TLS-encrypted HTTP message. Firstly, the victim host sends a DoH request containing the DNS covert channel domain name, *updata.tunnel.com*, through a local DoH proxy or directly to the DoH server. *Updata* refer to sensitive information leaked from the victim host or command requests sent to the attacker. Secondly, the DoH server parses the DNS request and performs an iterative query, which is eventually forwarded to a disguised authoritative domain name server controlled by the attacker, i.e., C&C Server. Finally, the attacker obtains the *updata* sent by the victim host through the C&C server. The attacker also issues commands, i.e., *downdata*, through a disguised authoritative domain name server and delivers *downdata* to the victim host by DNS response and DoH response.

In general, the DoH-encrypted DNS covert communication between the attacker and the victim host is similar to the scenario of non-encrypted DNS covert communication: the principles of building DNS covert communication are the same. As shown in Figure 1, both *updata* and *downdata* are iteratively queried, and the disguised authoritative domain name server is used as the C&C server to relay between the attacker and the victim host.

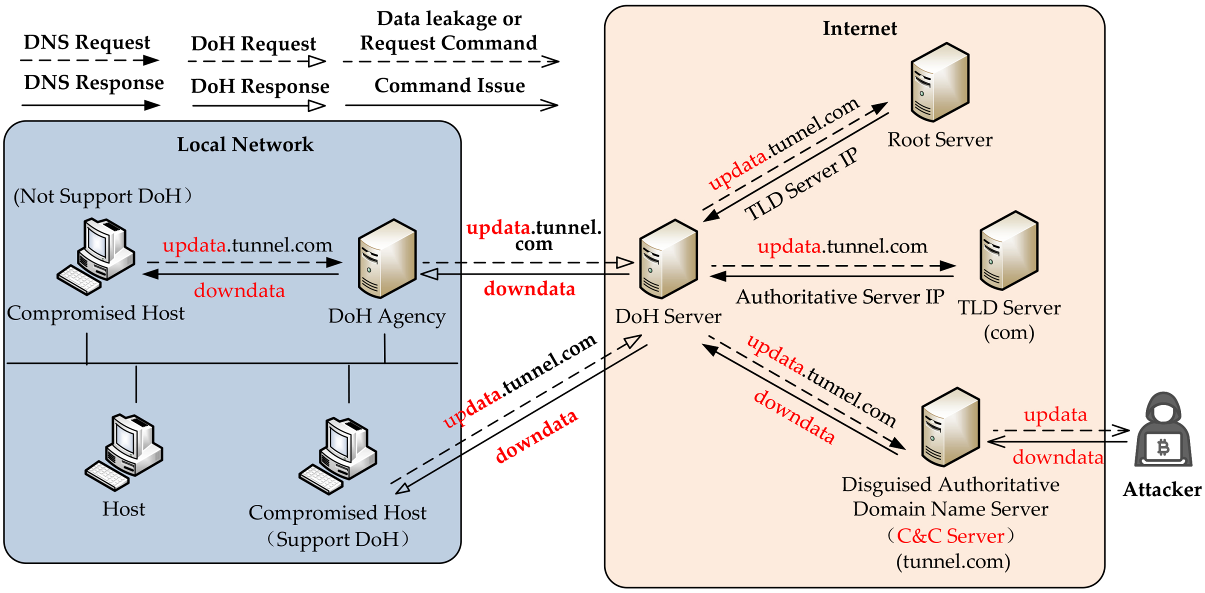


Figure 1. Data leakage and command control process based on DoH-encrypted DNS covert channels.

The difference between a non-encrypted DNS covert channel and a DoH-encrypted DNS covert channel is reflected in two aspects. First, as a data carrier for leakage and command and control, DNS covert channels in DoH traffic are encrypted, making it impossible to apply existing deep packet inspection techniques. Second, DNS covert channels are hidden in HTTPS traffic, and the DoH server acts as a local DNS server to forward DNS traffic, which also makes it impossible for the local network administrator to find the malicious activities of the victim through DNS. At the same time, the victim host reduces the frequency of DNS requests, reducing the suspicion of malicious activities. The above two characteristics bring a greater challenge for DoH-encrypted DNS covert channel detection.

3.2. Multi-Head Attention Mechanism

Attention mechanisms were first proposed in the field of image processing [15]. In 2014, the Google mind team combined an RNN and an attention mechanism and applied it to an image classification task [16]; this was then further developed and expanded by researchers. In different fields, many different attention mechanisms have evolved, including basic attention mechanisms, such as Soft Attention, Hard Attention, Self-Attention, etc., and combined attention mechanisms, such as Co-Attention, Attention-over-Attention, Multi-Head Attention [17]. Although the above attention mechanisms are different, the basic principles of implementation are similar. This section provides a brief overview of the Multi-Head Attention mechanism by summarizing the general implementation principles of the attention mechanism.

In essence, the attention mechanism can be summarized as filtering out important and noteworthy information by computing the weight distribution of attention within or among data. It usually contains three variables, Query, Key, and Value ($\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}$), which represent the data encoding using target data, source data encoding, and content data encoding, respectively. The calculation process can be divided into two steps: one is to calculate the similarity between the target data $\boldsymbol{Q}$ and the source data $\boldsymbol{K}$, and the other is to calculate the new data representation $\boldsymbol{V'}$ based on the similarity and $\boldsymbol{V}$:

$$\boldsymbol{e} = g(f(\boldsymbol{Q}, \boldsymbol{K})) \tag{1}$$

$$\boldsymbol{V'} = m(\boldsymbol{e}, \boldsymbol{V}) \tag{2}$$

where the similarity between $\boldsymbol{Q}$ and $\boldsymbol{K}$ is calculated by the energy function f [18] and the distribution function g to obtain the weight distribution of attention $\boldsymbol{e}$. Later, using the transformation function m, the new data representation $\boldsymbol{V}'$ is obtained by multiplying $\boldsymbol{e}$ with $\boldsymbol{V}$. Usually, the distribution function g is chosen to be normalized by *softmax*, while the transformation function m is a weighted summation. The usual energy functions f include additive and dot product functions [19], which are calculated as follows:

$$f(\boldsymbol{Q},\boldsymbol{K}) = \boldsymbol{v}^{\boldsymbol{T}}\,\mathrm{act}(\boldsymbol{W_1K} + \boldsymbol{W_2Q} + \boldsymbol{b}) \tag{3}$$

$$f(\boldsymbol{Q},\boldsymbol{K}) = \boldsymbol{Q}^{\boldsymbol{T}}\boldsymbol{K} \tag{4}$$

where act is the nonlinear activation function, such as tanh and ReLU, etc.; $\boldsymbol{v}^{\boldsymbol{T}}$ is the parameter vector; $\boldsymbol{b}$ is the neuron bias; and $\boldsymbol{W_1}$ and $\boldsymbol{W_2}$ are the weight matrices.

The Self-Attention mechanism was proposed in 2017 by Vaswani et al. [20] for computing the correlation between words within a sentence to extract syntactic and semantic features. The difference compared to the general attention mechanism is that $\boldsymbol{Q} = \boldsymbol{K} = \boldsymbol{V}$; that is, it only focuses on the interdependencies between elements within the data.

The Multi-Head Attention mechanism belongs to a kind of combined attention mechanism, which can greatly improve the data fitting ability and enrich the feature representation by combining multiple attention mechanisms $\boldsymbol{head_i}$ and jointly extracting information from different representation subspaces:

$$\begin{gathered}\mathrm{MultiHead}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \mathrm{Concat}(\boldsymbol{head_1},\ldots,\boldsymbol{head_h})\boldsymbol{W^o} \\ where\ \boldsymbol{head_i} = m(g(f(\boldsymbol{Q},\boldsymbol{K})),\boldsymbol{V})\end{gathered} \tag{5}$$

It should be noted that the Multi-Head Attention mechanism used in our paper is combining multiple Self-Attention mechanisms.

Nowadays, attention mechanisms are widely used in natural language processing [21], as well as in autonomous driving [22] and human–computer interaction [23], among others. After a major breakthrough by the Google mind team using attention mechanisms in image processing, researchers have also used them in natural language processing. Bahdanau et al. [24] first used the attention mechanism to solve the word alignment problem of indeterminate-length sentences in machine translation. Furthermore, attention mechanisms are gradually being taken advantage of in applications in the field of cyberspace security [25,26]. In their study of abnormal traffic and encrypted malicious traffic detection, Jiang et al. [27] used LSTM with CNN to extract spatio-temporal features of packets on the CICAndMal2017 dataset and further used a Multi-Head Attention mechanism to extract sequence features of multiple packets in a session. Wang et al. [28] deployed a single-layer Self-Attention mechanism on the CIC-IDS-2017 dataset to learn the correlation and dependency within statistical features in order to detect abnormal and attack traffic. Dong et al. [29] added convolutional operations between multi-layer Self-Attention mechanisms to improve performance over models such as GoogLeNet and ResNet-50 on the NSL-KDD dataset. For encrypted traffic classification, Lin et al. [30] proposed the ET-BERT (Encrypted Traffic Bidirectional Encoder Representations from Transformer) model. Based on the BERT model [31], traffic is converted to tokens for pre-training. They proposed two fine-tuning strategies, packet-level fine-tuning for single-packet classification and stream-level fine-tuning for single-stream classification, and verified the robustness and generalization ability of the model on five encrypted traffic datasets and the TLS1.3 dataset.

4. Method Design

In this section, we design a DoH-encrypted DNS covert channel detection method named FF-MR based on features fusion. FF-MR, including a Multi-Head Attention mechanism and a Residual Neural Network, fuses statistical features and byte sequence features. Its framework is shown in Figure 2.

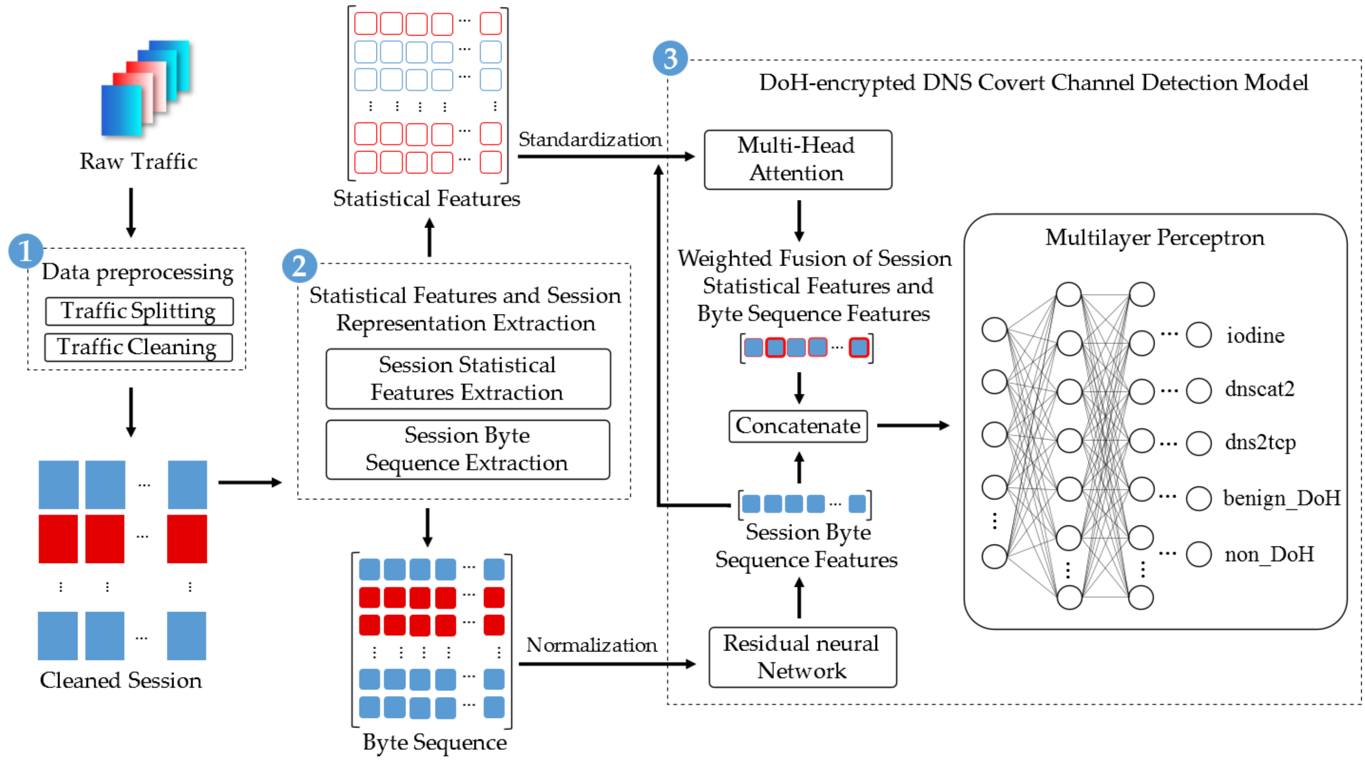

Raw Traffic
1
Data preprocessing
Traffic Splitting
Traffic Cleaning
Cleaned Session
2
Statistical Features and Session Representation Extraction
Session Statistical Features Extraction
Session Byte Sequence Extraction
Statistical Features
Byte Sequence
Standardization
Normalization
3
DoH-encrypted DNS Covert Channel Detection Model
Multi-Head Attention
Weighted Fusion of Session Statistical Features and Byte Sequence Features
Concatenate
Session Byte Sequence Features
Residual neural Network
Multilayer Perceptron
iodine
dnscat2
dns2tcp
benign_DoH
non_DoH

Figure 2. Framework of FF-MR.

FF-MR is mainly divided into three parts: data preprocessing, statistical features and session representation extraction, and a DoH-encrypted DNS covert channel detection model based on Multi-Head Attention and Residual Neural Network. The proposed method achieves HTTPS (i.e., non_DoH), normal DoH (i.e., benign_DoH), and three kinds of malicious DoH traffic (i.e., iodine, dnscat2, and dns2tcp) in five categories. Iodine, dnscat2, and dns2tcp are three kinds of malicious DoH traffic generated by encrypted DNS covert channel tools, that is, DoH-encrypted DNS covert channel traffic.

Firstly, the data preprocessing module splits and reorganizes the raw pcap file into sessions, and then, we clean these sessions for filtering and anonymization.

Secondly, session representation and statistical features are extracted, and after standardization and normalization, they are used as the input of the DoH-encrypted DNS covert channel detection model.

Finally, in the DoH-encrypted DNS covert channel detection model based on the Multi-Head Attention and Residual Neural Network (MHA-Resnet), byte sequence features are extracted by the Residual Neural Network, and the Multi-Head Attention mechanism calculates the weighted fusion of session statistical features and byte sequence features so that the distinction between features of different traffic sources is more pronounced. Moreover, the classification of four kinds of DoH and HTTPS traffic is achieved by a Multilayer Perceptron (MLP) to detect and identify DoH-encrypted DNS covert channels.

4.1. Data Preprocessing

Data preprocessing is divided into two steps—traffic splitting and traffic cleaning—in order to obtain traffic representation suitable for the detection model and to remove any invalid data mixed in with the original traffic that could reduce the classification performance of the model.

4.1.1. Traffic Splitting

We split the original pcap file into multiple temporally contiguous sets of packets according to certain rules. The five-tuple ($tuple$) contained in each packet is comprised of source IP address $srcIP$, destination IP address $dstIP$, source port $srcPort$, destination port $dstPort$, and transport layer protocol type $protocol$. The ith packet, p_i, can be defined by the start transmission time $time_i$, five-tuple $tuple_i$, and payload $payload_i$ as follows:

$$tuple = (srcIP, dstIP, srcPort, dstPort, protocol) \tag{6}$$

$$p_i = (time_i, tuple_i, payload_i). \tag{7}$$

$Flow$ is the set of packets with the same $tuple$, and all packets in a $flow$ have the same origin and destination and are independent of $time$ and $payload$. $Session$ refers to a bidirectional flow. Packets in a session have the same $tuple$ or $tuple'$, where $tuple'$ has $srcIP, dstIP$, $\{srcPort, dstPort\}$ and is the opposite of that in $tuple$. Therefore, even though the packets do not have exactly the same five-tuple, they are still considered to be the same session. $Flow$ and $session$ are expressed as:

$$flow = \{p_1, p_2, \ldots, p_N\}, tuple_1 = tuple_2 = \ldots = tuple_N \tag{8}$$

$$\begin{gathered} session = \{p_1, p'_1, p_2, p'_2, \ldots, p_N, p'_M\} \\ where\ tuple_1 = tuple_2 = \ldots = tuple_N, tuple'_1 = tuple'_2 = \ldots = tuple'_M. \end{gathered} \tag{9}$$

The raw pcap file is split into sessions using the SplitCap tool, which can optionally split the file by flow. In addition, the tool can also choose to keep all the data of the protocol layers or only the data above the transport layer. Since we need to extract session statistical features, the result of splitting the traffic is to retain all of the session's protocol layers.

4.1.2. Traffic Cleaning

We sort, filter, and anonymize the sessions. Firstly, sessions after traffic splitting are classified and sorted into five categories according to detection results in Figure 2, namely, iodine [32], dnscat2 [33], dns2tcp [34], benign_DoH, and non_DoH, where iodine, dnscat2, and dns2tcp represent malicious_DoH sessions for different types of DoH-encrypted DNS covert channels. Benign_DoH refers to normal DoH sessions, in which packets are encrypted DNS packets without DNS covert channels, and non_DoH refers to HTTPS sessions.

Secondly, it is necessary to filter out sessions with too little data because of the uneven session size. The main principle of filtering is to remove sessions with fewer packets than min_window_size because the raw pcap files corresponding to sessions may be incomplete, and the TLS handshake information needed for model classification may be missing, which will greatly reduce the performance of model classification. Furthermore, since the input of the detection model named MHA-Resnet is of a fixed length, the session length, in bytes, needs to be unified. However, if the number of packets in the session is small, it will result in a small number of session bytes; thus, the extracted byte length will be insufficient, and a large amount of zero-padded byte data will be generated when the length is unified, which may also affect the performance of the model's classification. Since the TCP connection and TSL handshake are generally completed before the sixth packet of the session, the value of min_window_size is taken as six in this paper.

Finally, the $tuple$ of packets in a session is either the same or opposite, and the classification is directly influenced by the $tuple$, resulting in classification exclusively according to the $tuple$ rather than the features of the session, which greatly affects the detection and identification performance of MHA-Resnet. Therefore, the session needs to be anonymized. Specifically, the port, IP, and MAC address of each packet in the session are overwritten with all zeros. In this way, the impact of specific fields on classification can be minimized.

4.2. Statistical Features and Session Representation Extraction

After preprocessing, we extract the statistical features and session representation as input to the detection model in two steps, as described in Sections 4.2.1 and 4.2.2.

4.2.1. Session Representation Extraction

To make up for the fact that only using the statistical features is insufficient to detect DoH-encrypted DNS covert channels, we intercept a string of bytes from a session after traffic cleaning to use as input in the detection model. After analyzing the CIRA-CIC-DoHBrw-2020 dataset for HTTPS and DoH traffic, there is a certain difference in packet size between the two in the TCP connection stage, which is mainly reflected in the optional fields of the TCP headers. For example, in order to avoid serial number wrapping, most DoH traffic contains a TSval field for reliable transmission. In addition, because DoH traffic needs to query the domain name, the response time is longer, and time-related fields, such as TSval and Timestamps in the TCP header, reflect this communication delay, which can also be used as TCP transmission features to distinguish HTTPS and DoH traffic.

The distinction between normal DoH and malicious DoH traffic, i.e., DoH-encrypted DNS covert channel, is mainly reflected in the TLS handshake stage, where the communicating parties negotiate the plaintext information, such as the TLS version, extension, cipher suite, certificate, and elliptic curve type used for encryption and decryption. To a certain extent, the plaintext information reflects the trustworthiness of the encrypted session. Due to the lack of security and formality guarantees for malicious DoH traffic, the above plaintext information is different from normal DoH traffic; for example, malicious traffic is more likely to use a lower version of the encryption algorithm. Normal encrypted traffic mostly uses Extended Validation SSL Certificates (EV SSL) and other highest trust level certificates [35]. In related studies, the certificate information and Client Hello message have also been verified to ensure a good degree of differentiation [36,37]. For different forms of malicious DoH traffic, the TLS handshake information negotiated by the encrypted DNS covert channel, such as cipher suite and elliptic curve type, etc., is not consistent, so this information can be used as an effective feature to detect and identify encrypted malicious DNS covert channels.

In summary, the packet size during the TCP connection stage, the timestamp field of the TCP packet, and the non-encrypted messages in the TLS handshake stage all reflect the characteristics of communication behavior of different types of traffic. Therefore, instead of focusing on data below the network layer, we concatenate the traffic data in the TCP layer with the traffic data in the TLS layer of each packet, extracting the first n bytes as the session representation. The number of bytes n is used as the hyperparameter of the detection model, and 512, 1024, 2048, 4096, and 8192 bytes are selected in the experiment in Section 5.3. According to the experimental results, we choose 1024 bytes as the session representation for the input n of the DoH-encrypted DNS covert channel detection model. The byte vector $\boldsymbol{X_i}$ of the ith session after normalization can be expressed as:

$$\boldsymbol{X_i} = \left[x_1^i, x_2^i, \ldots, x_k^i, \ldots, x_n^i\right] \tag{10}$$

where x_k^i is the kth byte of the ith session.

4.2.2. Session Statistical Features Extraction

We extract a total of 29 sessions statistical features of five categories, as shown in Table 2. The session duration, number of bytes, packet length, packet time, and request/response time difference are counted. We calculate the mean, median, mode, variance, standard deviation, coefficient of variation, skew from median, and skew from mode for three features, except session duration and number of bytes. At the same time, the rate of session bytes sent and received are calculated because the above five categories of features characterize the characteristics of DoH-encrypted DNS covert channel traffic. For example, DoH-encrypted DNS covert channels contain TCP traffic with covert transmission, which requires more data to be sent, so the values of session duration, number of bytes, and packet length are larger. Compared with the normal DoH and HTTPS traffic, DOH-encrypted DNS covert channel traffic usually has a lower cache hit rate, which leads to a higher latency, a higher frequency of sending packets, and a larger time difference between request and response.

After experimental verification, these features can better represent the difference between the DoH-encrypted DNS covert channel and normal DoH and HTTPS traffic. Because of the different magnitudes, numerical values vary greatly in the statistical features, and it is therefore necessary to be standardized to maintain numerical sensitivity. The standardized statistical features can be expressed as:

$$\boldsymbol{S_i} = \left[s_1^i, s_2^i, \ldots, s_k^i, \ldots, s_{29}^i\right] \tag{11}$$

where $\boldsymbol{S_i}$ is the statistical feature vector of the ith session, and s_k^i is the kth statistical feature value of the ith session.

Table 2. Session statistical features.

Category	Number	Feature
Duration	1	Session duration
Number of bytes	2	Number of session bytes sent
	3	Rate of session bytes sent
	4	Number of session bytes received
	5	Rate of session bytes received
Packet length	6	Mean Packet Length
	7	Median Packet Length
	8	Mode Packet Length
	9	Variance of Packet Length
	10	Standard Deviation of Packet Length
	11	Coefficient of Variation of Packet Length
	12	Skew from median Packet Length
	13	Skew from mode Packet Length
Packet time	14	Mean Packet Time
	15	Median Packet Time
	16	Mode Packet Time
	17	Variance of Packet Time
	18	Standard Deviation of Packet Time
	19	Coefficient of Variation of Packet Time
	20	Skew from median Packet Time
	21	Skew from mode Packet Time
Request/response time difference	22	Mean Request/response time difference
	23	Median Request/response time difference
	24	Mode Request/response time difference
	25	Variance of Request/response time difference
	26	Standard Deviation of Request/response time difference
	27	Coefficient of Variation of Request/response time difference
	28	Skew from median Request/response time difference
	29	Skew from mode Request/response time difference

We have improved the statistical feature extraction tool DoHMeter [38]. The original DoHMeter tool extracts statistical features in the unit of time-divided streams, and the improved DoHMeter extracts statistical features in session (bidirectional stream) units. Compared with the standard tool, the features extracted by the improved DoHMeter are more complete, sufficient, and more effective for detection using experimental verification.

4.3. Model Development and Architecture

The MHA-Resnet architecture includes three parts: the extraction of session byte sequence features, the weighted fusion of session statistical features and byte sequence features, and session classification. The model is based on the Residual Neural Network.

The Multi-Head Attention mechanism is used to globally weight and fuse the features to improve the detection performance of the model. The structure of MHA-Resnet is shown in Figure 3. In order to learn the different communication behaviors of five types of traffic and the mode of TLS encrypted connection, session statistical features and byte data are taken as the input of the model. Byte sequence features, including TCP transmission features, TLS handshake features, and local patterns of DoH-encrypted DNS covert channels, are extracted by the multiple one-dimensional convolutional layers (Conv1D) of the Residual Neural Network, which are then concatenated with session statistical features. The attention weight distribution between all features is calculated by the Multi-Head Attention mechanism, which re-represents the features as the result of a weighted fusion. The output vector of the model is the probability that a session is judged as each of the five types of traffic (i.e., iodine, dnscat2, dns2tcp, benign_DoH, non_DoH), and the label of the maximum probability in the output vector is taken as the classification result.

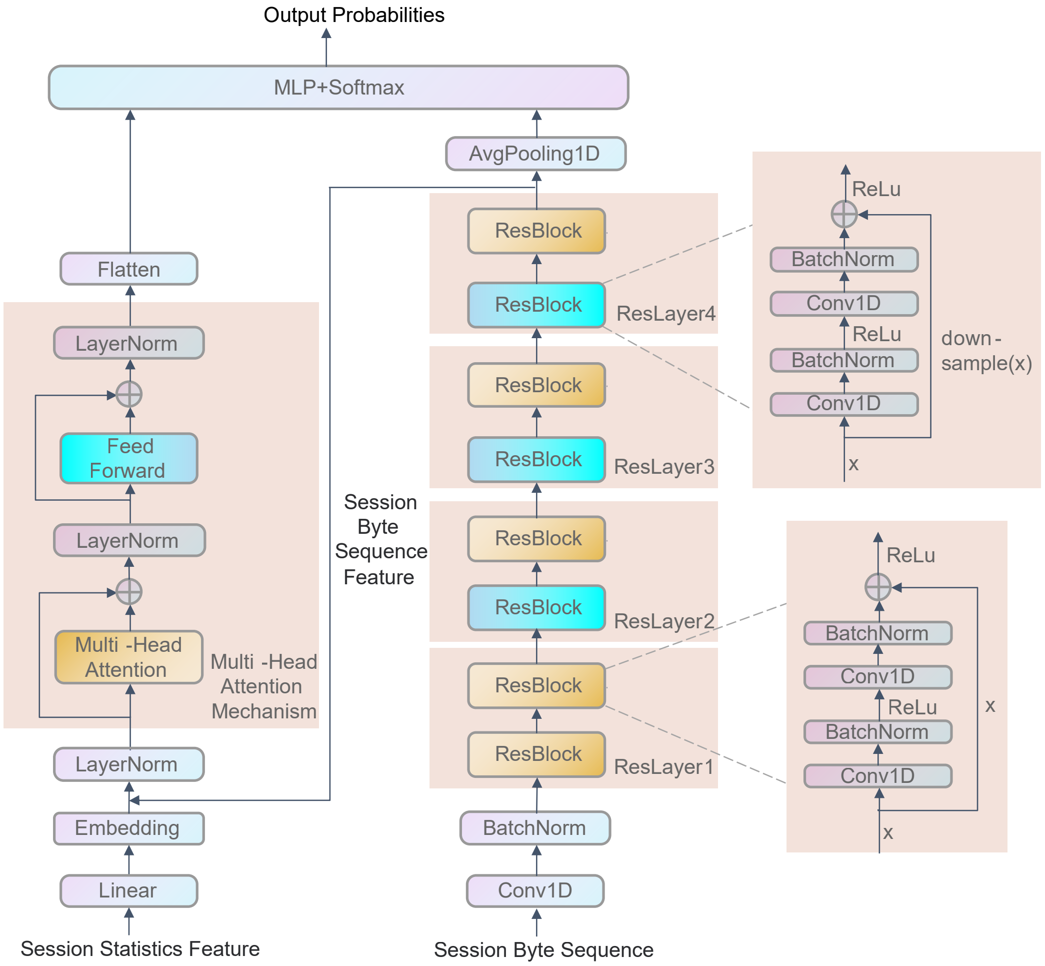


Figure 3. Structure of MHA-Resnet.

The model contains the design of residual connection in both of the above neural networks and achieves the purpose of adaptively adjusting the number of network layers according to the task needs by generating constant mapping of the redundant network layers when it is unnecessary. This mitigates to some extent the negative impact of the deep neural network degradation problem on the model performance [39].

4.3.1. Session Byte Sequence Features Extraction

We adopt a Residual Neural Network to extract the session byte sequence features, treating the bytes as words in Natural Language Processing (NLP) tasks and the session byte sequences as sentences, obtaining the contextual associations of the bytes in the session through Conv1D and then extracting the combined byte information, including fields and messages. Moreover, shallow convolution is used to obtain the contextual association of bytes inside the TCP header and TLS handshake message, at which time the combined byte information is at the field level, corresponding to the packet size in the TCP connection stage, timestamp, and TLS certificate mentioned in Section 4.2, i.e., TCP transmission features and TLS handshake features. Deep convolution is used to obtain the contextual association of the TCP and TLS messages in the session when the combined byte information is at the message level, corresponding to the extraction of the correlation between adjacent TCP and TLS messages, i.e., the local pattern of the DoH-encrypted DNS covert channel during transmission.

As shown in Figure 3, in order to process network traffic data with a one-dimensional sequence structure, the Residual Neural Network is based on one-dimensional convolution, and the main body consists of four residual layers (ResLayer). Each residual layer consists of two residual blocks (ResBlock), and two sets of one-dimensional convolutional layers (Conv1D) with batch normalization layers (BatchNorm) are used in the basic residual blocks. The difference between the different residual layers is that the first residual block in the last three residual layers adds to the downsampling operation, and the structure of the downsampling is shown in Figure 4.

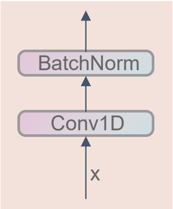


Figure 4. Downsample.

Firstly, we ascend dimensionality of the first n bytes $\boldsymbol{X_i}$ of the ith session after a layer of one-dimensional convolution using the batch normalization operation. Specifically, the multi-channel feature matrix $\boldsymbol{X_i'}$ is obtained by multiple convolution kernels, where multiple convolution kernels represent multiple feature extractors, indicating the extraction of different convolutional features with adjacent bytes. Batch normalization is mainly used to solve the problem of gradient disappearance or explosion in deep neural networks.

Secondly, four residual layers extract convolutional features of adjacent fields or messages. Afterwards, the preliminary convolution operation of $\boldsymbol{X_i'}$ is performed in the first residual layer to obtain the output with the same dimension as the input and the next three residual layers to extract the sequence features under different step lengths using the downsampling operation. The downsampling operation can be performed by decreasing the size of the convolution kernel and increasing the step length in order to extract sequence features with multiple fields while ensuring the same tensor dimensionality for residual concatenation. The calculation of the multi-channel session's byte sequence features $\boldsymbol{X_i''}$ is as follows:

$$\boldsymbol{X_i'} = \text{BatchNorm}(\text{Conv1D}(\boldsymbol{X_i})) \tag{12}$$

$$\boldsymbol{X_i''} = \text{Reslayer4}(\text{Reslayer3}(\text{Reslayer2}(\text{Reslayer1}(\boldsymbol{X_i'})))) \tag{13}$$

Finally, the multi-channel session byte sequence features $\boldsymbol{X_i''}$ are input into the neural network composed of the Multi-Head Attention mechanism for weighted feature fusion. The main significance of the multi-channel features is to fully characterize the different TCP transmission features, TLS handshake features, and local transmission patterns of the DoH-encrypted DNS covert channels extracted by the multi-convolutional kernel and multi-step and then correlate them with statistical features, thus filtering out unnecessary features. On the other hand, after the global one-dimensional averaging pooling operation

(Avgpooling1D) is performed, the features in each channel are averaged to simplify computation. Then, we obtain the session byte sequence feature vector $\boldsymbol{Res\_X_i}$ extracted by the Residual Neural Network:

$$\boldsymbol{Res\_X_i} = \mathrm{AvgPool1D}\left(\boldsymbol{X_i''}\right) \tag{14}$$

4.3.2. Weighted Fusion of Session Statistical Features and Byte Sequence Features

In existing studies, the byte sequence features or statistical features alone are not enough to detect and identify encrypted DNS covert channels. Multifaceted features, such as session statistical features, TCP transmission features, TLS handshake features, and encrypted DNS covert channel transmission patterns of the same category of traffic, are not independent but have some correlation and are uniformly related to the behavior patterns of normal or malicious DoH traffic. Therefore, on the basis of the Residual Neural Network extracting byte sequence features, we also adopt the Multi-Head Attention mechanism in MHA-Resnet. The Self-Attention mechanism in Multi-Head Attention treats the distance between any two features as one and can obtain the global correlation between features, with the purpose of focusing on important features and ignoring redundant and useless features by assigning weights. Specifically, the Self-Attention mechanism expresses the global correlation and dependency between the above multi-faceted features as an attention weight matrix: the stronger the correlation between two features, the larger the weight. The more important a feature is, the more strongly correlated it is with multiple other features. The fusion features with greater distinction for detection are obtained by the weighted summation of all features. This method, which considers both global and focused features, solves the problem of long-distance dependency in RNN, highlights important features and their mapping relationships in the overall features using the correlations between multi-faceted features, and further improves detection performance.

The interpretability of the global correlation of features extracted by the Self-Attention mechanism can be visualized as Figure 5a of the attention distribution in machine translation. The solid line indicates the referential and correlation relationship between words. For example, the words related to the word "it" through the learning of the Self-Attention mechanism include "The", "cat", "street", "it", and punctuation "."; the strongest correlation is the word "cat", which is consistent with the semantics of the sentence. In this case, the correlation is a semantic-grammatical feature. Similarly, applying the Self-Attention mechanism to DoH traffic, the global correlation of features can be considered as the connection between multi-faceted features embodied in activities and behaviors of encrypted DNS covert channels, i.e., the correlation relationship between the session statistical features and byte sequence features within and among each other.

The Multi-Head Attention mechanism integrates multiple Self-Attention mechanisms to improve the robustness and generalization of MHA-Resnet by learning the features of different representation subspaces. As shown in Figure 5b, the distribution of attention learned by another Self-Attention mechanism is different from Figure 5a. Here, the word "it" has a strong correlation with "street".

As shown in Figure 6a,b, the TCP transmission features extracted from session byte sequence are correlated with Mean Packet Length, Mean Packet Time, and Mean Request/response time differences, while different attention mechanisms will produce different degrees of correlation. The TCP transmission features in Figure 6a are strongly correlated with the Mean Request/response time difference, while the TCP transmission features in Figure 6b are strongly correlated with the Mean Packet Length. Therefore, the Multi-Head Attention mechanism can represent the relationship between traffic features in multiple dimensions, thereby preventing overfitting.

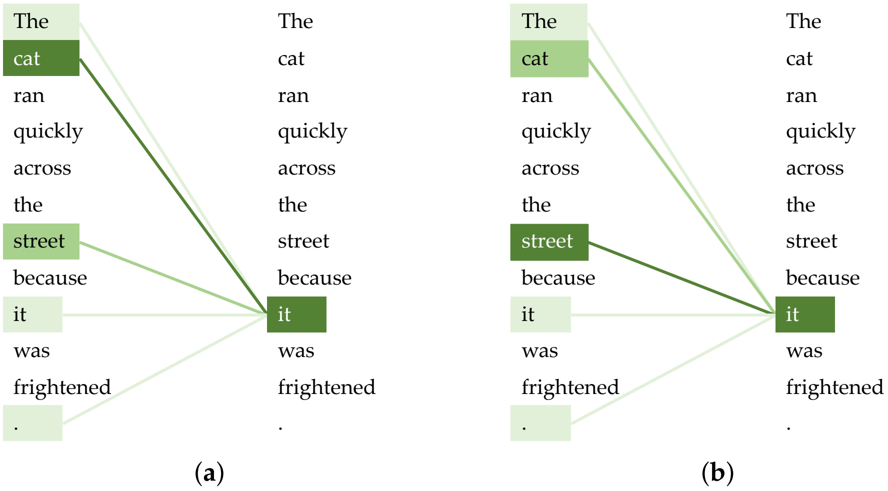


Figure 5. Interpretability of distribution of attention in machine translation: (**a**) a distribution of attention; (**b**) another distribution of attention.

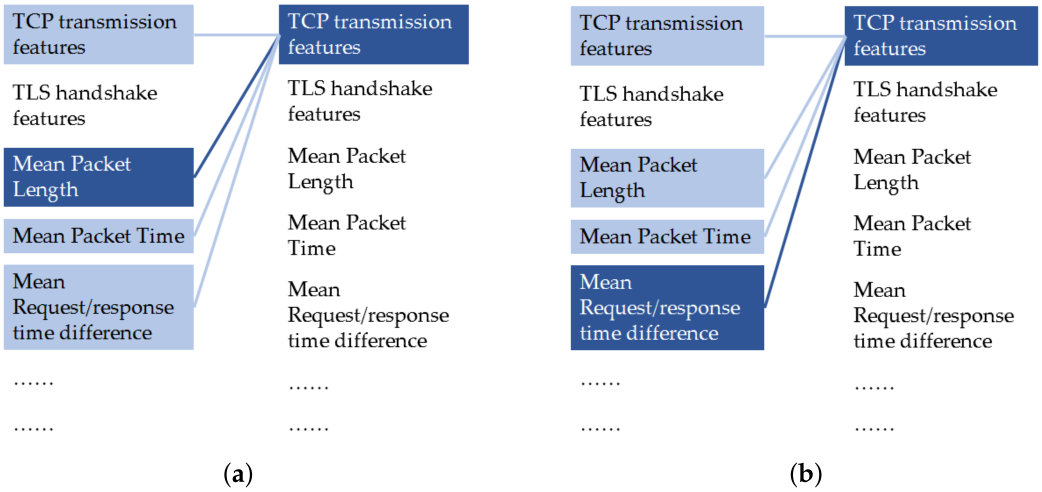


Figure 6. Interpretability of distribution of attention for features of traffic: (**a**) a distribution of attention; (**b**) another distribution of attention.

The computation of the Multi-Head Attention mechanism is divided into three steps. Firstly, to improve the nonlinear expression of the network, the statistical features $\boldsymbol{S_i}$ are input to a fully connected layer linear with the Sigmoid activation function and transformed into a two-dimensional word vector matrix through word embedding. It is concatenated with a multi-channel byte sequence feature matrix $\boldsymbol{X_i''}$ extracted by the Residual Neural Network. Meanwhile, we adopt the LayerNorm normalization for the sake of faster convergence and consistent data distribution and then obtain the input $\boldsymbol{U_i}$ of the Multi-Head Attention layer:

$$\boldsymbol{U_i} = \mathrm{LayerNorm}\big(\mathrm{Concat}\big(\mathrm{Embedding}(\mathrm{Linear}(\boldsymbol{S_i})), \boldsymbol{X_i''}\big)\big). \tag{15}$$

Secondly, Figure 7a [20] shows the weighted fusion of the session statistical features and byte sequence features using the scaled dot-product self-attention, which is more efficient compared to other Self-Attention mechanisms [40]. The scaled dot-product self-attention mechanism is implemented through $\boldsymbol{Q}$ (Query), $\boldsymbol{K}$ (Key), and $\boldsymbol{V}$ (Value), which are linear transformations of $\boldsymbol{U_i}$. Essentially, the attention weight distribution of *boldsymbolV* is determined by computing the similarity (multiplication) between $\boldsymbol{Q}$ and $\boldsymbol{K}$, where a larger weight indicates that a feature is more relevant to another feature and vice versa. Thus, the statistical features $\boldsymbol{S_i}$ and the multi-channel byte sequence features $\boldsymbol{X_i''}$ are computed by a single-scaled dot-product self-attention mechanism to obtain the feature matrix of weighted fusion $\boldsymbol{attention_i}$:

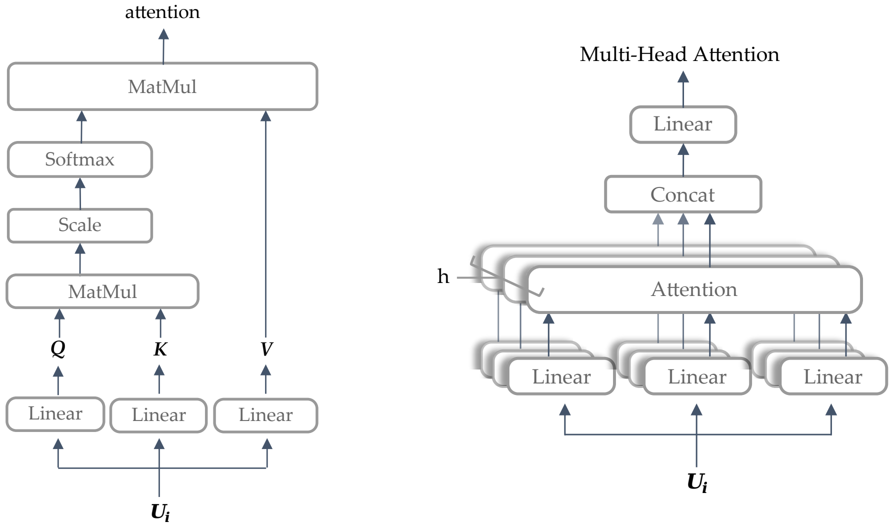
$$\boldsymbol{Q} = \boldsymbol{U_i W_i^Q} \tag{16}$$

$$\boldsymbol{K} = \boldsymbol{U_i W_i^K} \tag{17}$$

$$\boldsymbol{V} = \boldsymbol{U_i W_i^V} \tag{18}$$

$$\boldsymbol{attention_i} = \text{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{QK^T}}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{19}$$

where d_k is the dimension of $\boldsymbol{K}$, whose main function is to avoid the inner product of $\boldsymbol{QK^T}$ being too large, and softmax normalizes the feature matrix of weighted fusion $\boldsymbol{attention_i}$.



(**a**) (**b**)

Figure 7. Attention mechanism in this paper: (**a**) weighted fusion of features calculated by scaled dot-product self-attention; (**b**) weighted fusion of features in different representation subspaces calculated by Multi-Head Attention.

The singular Self-Attention is concatenated to obtain the Multi-Head Attention mechanism, as shown in Figure 7b [20]. When calculating $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ in the h heads Self-Attention mechanism, $boldsymbolV$, the initialized $\boldsymbol{W_i^Q}$, $\boldsymbol{W_i^K}$ are different, as is the similarity of $\boldsymbol{Q}$ to $\boldsymbol{K}$; therefore, the feature matrix of weighted fusion $\boldsymbol{attention_i}$ also differs. MHA-Resnet combines the features of different representation subspaces by Concat. After linear transformation $\boldsymbol{W^O}$, $\boldsymbol{attention_i}$ is dimensionally reduced to MultiHead$(\boldsymbol{U_i})$ of the same dimension as the input $\boldsymbol{U_i}$. After the residual connection and LayerNorm normalization, we obtain $\boldsymbol{U_i'}$:

$$\begin{gathered}\text{Mu ltiHead}(\boldsymbol{U_i}) = \text{Concat}(\boldsymbol{attention_1}, \ldots, \boldsymbol{attention_h})\boldsymbol{W^O} \\ where\ \boldsymbol{attention_i} = \text{Attention}\left(\boldsymbol{U_i W_i^Q}, \boldsymbol{U_i W_i^K}, \boldsymbol{U_i W_i^V}\right)\end{gathered} \tag{20}$$

$$\boldsymbol{U_i'} = \text{LayerNorm}(\boldsymbol{U_i} + \text{MultiHead}(\boldsymbol{U_i})). \tag{21}$$

Finally, we obtain the weighted fusion $\boldsymbol{MHA\_U_i}$ of session statistical features and multi-channel byte sequence features through the nonlinear transformation in the forward feedback layer (FeedForward) and residual concatenation and smoothing (Flatten):

$$\boldsymbol{MHA\_U_i} = \text{Flatten}\left(\boldsymbol{U_i'} + \text{FeedForward}\left(\boldsymbol{U_i'}\right)\right). \tag{22}$$

The main reason for using smoothing instead of a maximum or average pooling is to reduce information loss so that different features in the multi-channel can be used for classification.

4.3.3. Session Classification

We combine the weighted fusion of features $\boldsymbol{MHA\_U_i}$ with the byte sequence features $\boldsymbol{Res\_X_i}$, classified by MLP with softmax. Specifically, $\boldsymbol{MHA\_U_i}$ and $\boldsymbol{Res\_X_i}$ are concatenated and fed into three fully connected layers with ReLu activation, while the probability vector predicted as each class of traffic, and the label corresponding to the maximum probability is taken as the predicted label $y\_predict_i$ for the ith session:

$$y\_predict_i = \text{softmax}(\text{MLP}(\text{Concat}(\boldsymbol{MHA\_U_i}, \boldsymbol{Res\_X_i}))) \tag{23}$$

where we add dropout between the fully connected layers to prevent overfitting in MLP.

5. Experimental Evaluation

This section is divided into five parts. Section 5.1 describes the dataset and performance metrics. Section 5.2 shows the hyperparameter settings in MHA-Resnet. We evaluate the performance of FF-MR in Section 5.3. We verify the effectiveness of the model MHA-Resnet in Section 5.4, and in Section 5.5, we implement the parameter sensitivity experiments.

5.1. Dataset and Performance Metrics

The CIRA-CIC-DoHBrw-2020 dataset [9] comes from the Canadian Institute for Cybersecurity Research, and the data preprocessing results are shown in Table 3. DoH traffic is generated using two browsers, Google Chrome and Mozilla Firefox, and three DNS covert channel tools, including iodine, dnscat2, and dns2tcp, through four DoH servers, including AdGuard, Cloudflare, Google DNS, and Quad9. The dataset contains three categories, namely, non_DoH, benign_DoH, and malicious_DoH, respectively, representing HTTPS traffic, normal DoH traffic, and malicious DoH, i.e., DoH-encrypted DNS covert channel traffic. The first two are generated by browsers using the HTTPS and DoH protocols, respectively, to access the top 10,000 domains on the Alexa website, while encrypted DNS covert channel traffic is generated by DNS covert channel tools, which can send DNS requests using TLS-encrypted HTTPS requests to special DoH servers.

Table 3. CIRA-CIC-DoHBrw-2020 dataset and preprocessing results.

Category	Browsers\Tools	Number of Flows	Number of Sessions	Number of Sessions after Preprocessing
malicious_DoH	iodine	46,613	12,368	12,367
	dnscat2	35,622	10,298	10,298
	dns2tcp	167,515	121,897	121,738
benign_DoH	Google Chrome Mozilla Firefox	19,807	27,940	26,238
non_DoH	Google Chrome Mozilla Firefox	897,493	492,171	485,654

FF-MR not only detects encrypted DNS covert channels, i.e., malicious_DoH from HTTPS and normal DoH traffic, but it also identifies traffic generated by three DNS covert channel tools. As shown in Table 3, the magnitude of the preprocessed data is still at the level of hundreds of thousands, indicating that the dataset is sufficient. The division ratio of the training set, validation set, and test set is 6:2:2.

The CIRA-CIC-DoHBrw-2020 dataset is imbalanced, so the commonly used performance metrics such as Accuracy are not applicable. We adopt three performance metrics:

Precision, Recall, and F1-Score for evaluation in five categories. The comprehensive performance metrics are macro-averaging:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{24}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{25}$$

$$\text{F1-Score} = \frac{2\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{26}$$

$$\text{Macro\_ P} = \frac{1}{n}\sum_{i=1}^{n}\text{Precision}_i \tag{27}$$

$$\text{Macro\_ R} = \frac{1}{n}\sum_{i=1}^{n}\text{Recall}\ _i \tag{28}$$

$$\text{Macro\_ F1} = \frac{1}{n}\sum_{i=1}^{n}\text{F1-Score}_i \tag{29}$$

where $n = 5$, true positive (TP) means the model predicts the target traffic and the actual case is the target traffic, true negative (TN) means the model predicts the non-target traffic and the actual case is also the non-target traffic, false positive (FP) means the model predicts the target traffic and the actual case is the non-target traffic, and false negative (FN), in contrast, means the model predicts the non-target traffic, and the actual case is the target traffic.

The experimental environment is a 12th Gen Intel (R) Core (TM) i7-12700K @4.70GHz, 64GB RAM, and 2×NVIDIA RTX 3090 GPUs. The proposed architecture is developed on Ubuntu 20.04 LTS based on Python 3.9.7, Pytorch 1.11.0, CUDA Toolkit 11.3, and cuDNN8200, and codes are run with GPU acceleration.

5.2. Hyperparameter Settings

To ensure the objectivity and validity of the method, we performed ten experiments with MHA-Resnet on the CIRA-CIC-DoHBrw-2020 dataset and averaged the final experimental results in Section 5.3.

In terms of hyperparameter setting, MHA-Resnet was trained with a cross-entropy loss function for evaluation and an Adam optimizer for optimization. The number of training epochs was set to 100. We adopted a dynamic learning rate, where the initial learning rate was set to 0.0001 and decayed by 0.1 times every 20 rounds. The structural parameters of MHA-Resnet are shown in Table 4.

5.3. Performance Evaluation

We visualized and analyzed the features by using t-SNE feature dimensionality reduction in Section 5.3.1. The detection performance of FF-MR was evaluated by comparing it with state-of-the-art methods in Section 5.3.2.

5.3.1. t-SNE Feature Dimensionality Reduction and Visual Analysis

Figure 8 shows the experimental results of the the normalized confusion matrix. We mainly focus on the detection results of encrypted DNS covert channels. As shown in the matrix, there was small amount of confusion in the classification of encrypted DNS covert channel traffic, i.e., malicious_DoH traffic generated by iodine, dnscat2, and dns2tcp. To visually analyze the classification results, the test data were saved with the feature vectors learned by the MHA-Resnet before applying softmax, and we randomly selected 500 samples for each type of traffic, which were then reduced to two dimensions by t-SNE [41], as shown in Figure 9.

Table 4. Structural parameters in MHA-Resnet.

Substructure	Layer	Operation	Input	Output
Residual neural Network	Conv1D	One-dimensional convolution	1*1024	32*1024
	ResLayer1	One-dimensional convolution*4	32*1024	32*1024
	ResLayer2	One-dimensional convolution*4	32*1024	64*512
	ResLayer 3	One-dimensional convolution*4	64*512	128*256
	ResLayer 4	One-dimensional convolution*4	128*256	256*128
	AvgPooling1D	Global average pooling	256*128	256*1
Multi-Head Attention mechanism	Linear	linear transformation + Sigmoid	29*1	14*1
	Embedding	word embedding	14*1	14*128
	Multi-Head Attention	calculate the attention weight matrix	(256 + 14)*128	(256 + 14)*128
	Feed Forward	linear transformation + ReLU+ linear transformation	(256 + 14)*128	(256+14)*128
	Flatten	Flatten the weighted fusion feature matrix	(256 + 14)*128	34,560*1
MLP+softmax	Linear	linear transformation + ReLU	34,560 + 256	200
	Linear	linear transformation	200	30
	Linear	linear transformation + softmax	30	5

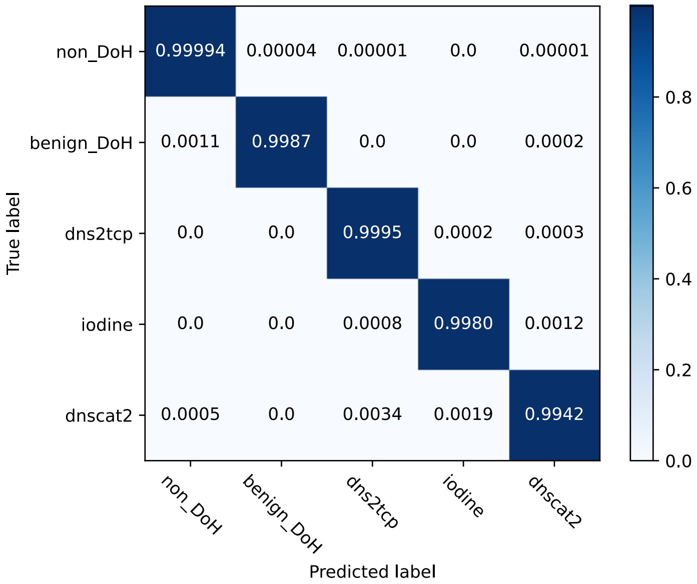


Figure 8. Normalized confusion matrix.

The same category of traffic is aggregated into a cluster in Figure 9, and the distinction between traffic in different categories is obvious, in which there is a small amount of confusion for the identification of the three encrypted DNS covert channels due to the effect of data imbalance, verifying the experimental results of Figure 8. As shown in Table 3, after data preprocessing, the encrypted DNS covert channel traffic generated by the iodine and dnscat2 tools was much smaller than other forms of traffic, resulting in the inability to learn the features to identify the above two categories of traffic. In general, this finding verifies that the FF-MR has a good feature extraction ability and performs well in detection and identification.

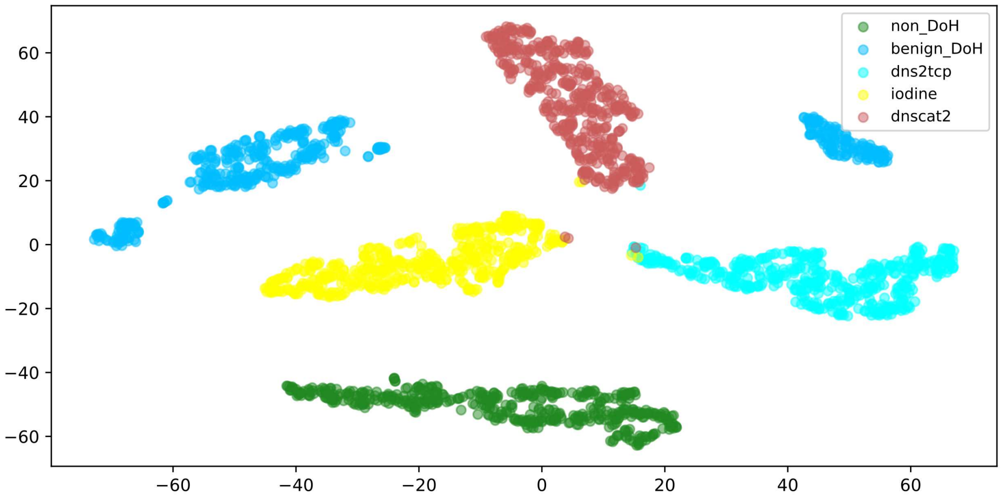


Figure 9. Visualization of t-SNE dimension reduction in traffic features.

5.3.2. Results and Evaluation

Baselines. To measure the improvements achieved by FF-MR, we reproduce four baselines:

- LightGBM [13] is a framework for implementing the Gradient Boosting Decision Tree (GBDT) algorithm, which supports efficient parallel training and has a faster training speed, lower memory consumption, and better accuracy. The method takes the statistical features of flow as input and outputs one of the five labels as a prediction;
- RF [8] is based on the Random Forest classifier, which adopts the same input and output as LightGBM [13]. The above two methods use the same statistical features as in Table 2;
- HAST-II [42] takes the first 4096 bytes of the session as input and combines CNN with LSTM to learn the spatial features and temporal features of the session bytes, respectively; it then uses a softmax classifier to perform five classifications on the spatio-temporal features;
- The input of CENTIME [43] is the same as ours: both use the statistical features and the first *n* bytes of the session. The difference is that they use the self-encoder to reconstruct the statistical features and the residual neural network with the same structure as ours to extract the byte sequence features and then concatenate the two inputs to the fully connected network for classification.

Results. As shown in Table 5, FF-MR achieved scores of 99.72%, 99.73%, and 0.9978 on Macro_P, Macro_R, and Macro_F1, respectively, demonstrating that the detection performance is better than the other four methods both in macro-averaging and metrics of identification of each category. Next, we will present a detailed comparative analysis of the experimental results though Table 5 and Figure 10.

Evaluation. As shown in Figure 10, comparing the results of LightGBM and RF, HAST-II, and LightGBM and RF, which use statistical features alone, it can be seen that these methods have higher recall and their overall performance is similar; HAST-II, which uses session bytes as input, has a higher precision. Although the results of the above three methods reflect the different advantages of the two features, the macro-averaging metrics are poor, and their F1-scores are only about 0.96. In contrast, the detection performance of FF-MR and CENTIME, which combine the two features, is better than the other three methods.

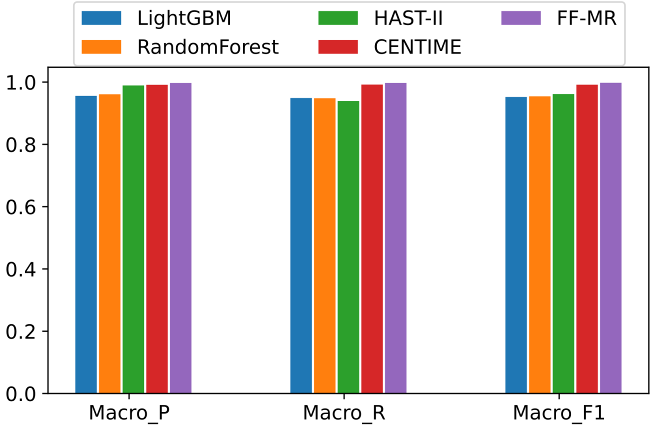


Figure 10. FF-MR vs. other methods on macro-averaging Metrics.

As shown in Table 5, in terms of the metric Macro_F1, FF-MR improves over LightGBM and RF by 4.56% and 4.35%, respectively, and 3.62% over HAST-II. The results of FF-MR in the five classifications are also significantly higher than the other three methods, especially in the identification of encrypted DNS covert channel traffic generated by the iodine and dnscat2 tools. FF-MR improves F1-Score when using iodine and dnscat2 by 6.06% and 8.22% over LightGBM, by 5.81% and 8.12% over RF, and by 5.67% and 11.32% over HAST-II, respectively, indicating the important role of the combined use of features for encrypted DNS covert channel identification.

Table 5. FF-MR vs. other methods.

	Metrics	LightGBM [13]	RF [8]	HAST-II [42]	CENTIME [43]	FF-MR
Macro-averaging	Macro_P	0.9558	0.9609	0.9892	0.9913	**0.9972**
	Macro_R	0.9489	0.9482	0.9391	0.992	**0.9973**
	Macro_F1	0.9522	0.9543	0.9616	0.9916	**0.9978**
iodine	Precision	0.9234	0.9319	0.9743	0.9773	**0.994**
	Recall	0.9458	0.942	0.905	0.9766	**0.9935**
	F1-Score	0.9345	0.937	0.9384	0.977	**0.9951**
dnscat2	Precision	0.9157	0.913	**0.9957**	0.9864	0.9939
	Recall	0.9107	0.9154	0.7919	0.9874	**0.9942**
	F1-Score	0.9132	0.9142	0.8822	0.9869	**0.9954**
dns2tcp	Precision	0.9911	0.9926	0.9761	0.9931	**0.9993**
	Recall	0.986	0.9881	**0.9998**	0.9982	0.9995
	F1-Score	0.9885	0.9903	0.9878	0.9956	**0.9995**
benign_DoH	Precision	0.9513	0.9697	**0.9999**	0.9994	0.999
	Recall	0.9033	0.896	0.999	0.9992	**0.9995**
	F1-Score	0.9267	0.9314	**0.9994**	0.9993	0.9992
non_DoH	Precision	0.9977	0.9975	0.9999	**1.0000**	0.9999
	Recall	0.9987	0.9994	**1.0000**	0.9987	0.9999
	F1-Score	0.9982	0.9985	**0.9999**	0.9993	**0.9999**

Our analysis is that LightGBM and RF belong to traditional machine learning algorithms, which are variations of decision tree algorithms. Therefore, we can infer that the decision tree algorithm does not achieve accurate classification and that decision tree integration algorithms cannot simply improve the performance of detection by optimizing the node splitting algorithm, i.e., XGBoost in LightGBM, or by increasing the number of decision trees (RF). The reason is that non_DoH, benign_DoH, and the three kinds of malicious DoH traffic are closer in the hyperplane, and multiple decision trees and the shallow neural network in HAST-II cannot divide them in a nonlinear way. However, the deep neural network used in FF-MR is able to achieve this distinction by training multiple layers of weights, thus greatly improving the performance of detection and identification.

Comparing FF-MR with CENTIME, although both use statistical features and byte sequence features and have similar performance, FF-MR is better than CENTIME in identifying the encrypted DNS covert channel traffic generated by iodine and dnscat2 tools due to the difference in the structure of the two models, improving F1-Score from 0.977 and 0.9869 to 0.9951 and 0.9954, respectively. The main reason for such improvement is that FF-MR is a weighted fusion of session statistical features and multi-channel byte sequence features though a Multi-Head Attention mechanism, instead of simply concatenating the two features in CENTIME. The drawback of CENTIME is that it does not mine the correlations between two features or give weighted attention to important features; thus, the performance of identifying a specific encrypted DNS covert channel is poor. The above results show that the Multi-Head Attention mechanism using the weighted fusion of features plays an important role in accurately identifying an encrypted DNS covert channel with smaller samples.

5.4. Validation of Effectiveness

We verify the effectiveness of MHA-Resnet from three aspects: first, we compare and validate the effect of one-dimensional and two-dimensional convolution on the model's classification performance; second, we assess the improvement of the model's classification performance using statistical features; third, the role of the Multi-Head Attention mechanism is comparatively verified on the CIRA-CIC-DoHBrw-2020 dataset.

Therefore, baseline models selected in this section include 1D-CNN, 2D-CNN, 1D-Resnet, and 2D-Resnet. The 1D-CNN and 2D-CNN both contain two convolutional layers in series, which are classified by fully connected networks; the difference between them is that the former is a one-dimensional convolution, and the latter is a two-dimensional convolution. The structural settings of 1D-Resnet and 2D-Resnet are the same as that of the Residual Neural Network in MHA-Resnet, and similarly, the difference is in the dimensions of convolution. The hyperparameters and other settings of the five models are the same, and the training losses are shown in Figure 11, which shows that all models have reached convergence without overlearning after 100 epochs. MHA-Resnet converges around the 20th epoch, while the other four models converge at around the 30th epoch, indicating that MHA-Resnet is more efficient in training.

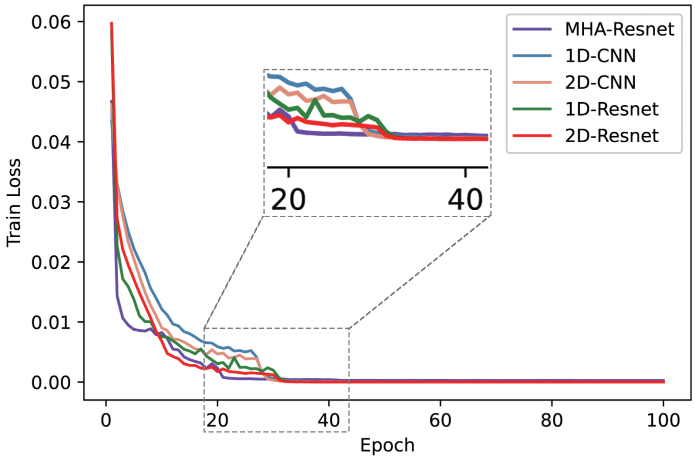


Figure 11. Training Loss.

As shown in Figure 12, the classification performance of the baseline models is compared with that of MHA-Resnet on macro-averaging metrics. MHA-Resnet has the best detection and identification performance, as shown in Table 6, showing an improvement of 9.03%, 8.52%, 1.42%, and 4.62% on Macro_F1 metrics over the baseline models, respectively.

The next best model in detection performance is 1D-Resnet, with all three metrics scoring around 0.98, while the Macro_F1 of 2D-Resnet with the same structure only reaches around 0.95, verifying that the one-dimensional convolution is more useful for processing network traffic. However, the classification performance of 1D-CNN and 2D-CNN is similar, and the Macro_F1 of the two models is only around 0.9, which illustrates the effectiveness of the Residual Neural Network in the MHA-Resnet.

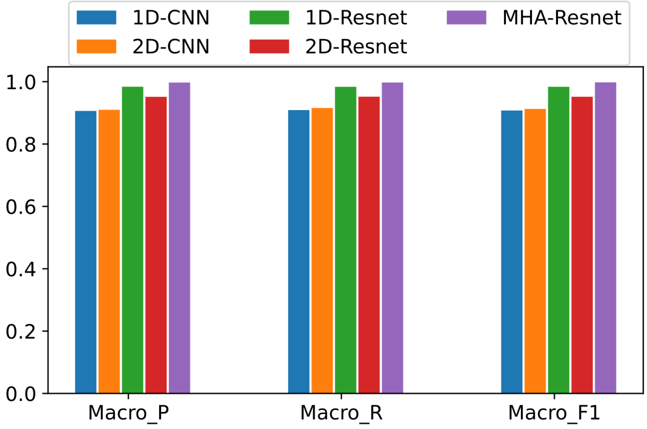


Figure 12. MHA-Resnet vs. Baseline Models on Macro-averaging Metrics.

The results of detection and identification are shown in Table 6. Iodine, dnscat2, and dns2tcp belong to malicious_DoH. The classification performance of non_DoH and benign_DoH is more desirable due to the different protocols; the former is HTTPS, while other categories of traffic are DoH. On the other hand, because benign_DoH is generated by browsers, while malicious_DoH is generated by three DNS covert channel tools, a large difference in the plaintext information occurs (e.g., TLS certificates, TLS cipher suites). However, the identification results of three types of encrypted DNS covert channels in malicious_DoH vary greatly among different models; especially, the traffic generated by iodine and dnscat2 tools are more difficult to identify. The F1-Scores of MHA-Resnet reach 0.9951 and 0.9954 for the identification of these two types of encrypted DNS covert channels, respectively. These scores are much higher than the other baseline models', improving by 14.88% and 28.34% over 1D-CNN, 14.16% and 26.58% over 2D-CNN, 2.67% and 4.05% over 1D-Resnet, and 7.26% and 14.61% over 2D-Resnet, respectively.

Table 6. MHA-Resnet vs. Baseline Models.

	Metrics	1D-CNN	2D-CNN	1D-Resnet	2D-Resnet	MHA-Resnet
Macro-averaging	Macro_P	0.9061	0.9098	0.9838	0.9514	**0.9972**
	Macro_R	0.9089	0.9157	0.9835	0.9519	**0.9973**
	Macro_F1	0.9075	0.9126	0.9836	0.9516	**0.9978**
iodine	Precision	0.847	0.8557	0.9696	0.9262	**0.994**
	Recall	0.8456	0.8513	0.9673	0.9188	**0.9935**
	F1-Score	0.8463	0.8535	0.9684	0.9225	**0.9951**
dnscat2	Precision	0.7007	0.7081	0.954	0.8424	**0.9939**
	Recall	0.7238	0.7524	0.9558	0.8563	**0.9942**
	F1-Score	0.712	0.7296	0.9549	0.8493	**0.9954**
dns2tcp	Precision	0.9851	0.9874	0.9959	0.9899	**0.9993**
	Recall	0.9825	0.9825	0.9959	0.9892	**0.9995**
	F1-Score	0.9838	0.985	0.9959	0.9896	**0.9995**

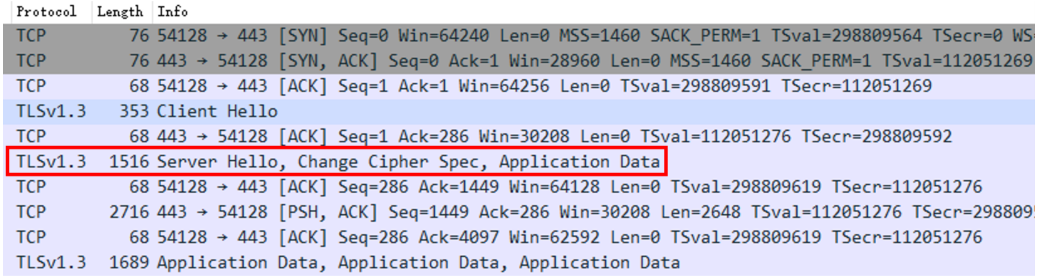
Table 6. *Cont.*

	Metrics	1D-CNN	2D-CNN	1D-Resnet	2D-Resnet	MHA-Resnet
benign_DoH	Precision	0.9983	0.9981	**0.9994**	0.9989	0.999
	Recall	0.9929	0.9926	0.9987	0.9952	**0.9995**
	F1-Score	0.9956	0.9953	0.999	0.997	**0.9992**
non_DoH	Precision	0.9996	0.9996	**0.9999**	0.9997	**0.9999**
	Recall	0.9999	0.9999	**1.0000**	0.9999	0.9999
	F1-Score	0.9997	0.9997	**0.9999**	0.9998	**0.9999**

Comparing 1D-Resnet and MHA-Resnet reveals that both contain one-dimensional Residual Neural Networks with the same structure; however, the difference is that statistical features are added to MHA-Resnet, and important features are highlighted using the Multi-Head Attention mechanism that not only enrich the training information but also enhance the representation ability of the model. This is the reason why the MHA-Resnet performs better in classification. The above comparative analysis verifies that statistical features are an important factor in improving the classification performance of a model when applied to the CIRA-CIC-DoHBrw-2020 dataset. It also verifies that the Multi-Head Attention mechanism improves the detection performance of the model by fusing both features.

5.5. Parameter Sensitivity Experiments

FF-MR extracts the first n bytes above the TCP layer as the session representation, aiming to extract the TCP transmission features, TLS handshake features, and the local patterns of encrypted DNS covert channel during transmission, as shown in Figure 13. In addition, the TLS messages after Server Hello are encrypted. Therefore, the principle for selecting n is that the byte sequence of length n should include at least the Client Hello and Server Hello, which are plaintext messages in the TLS handshake stage, with as many TCP messages as possible on top of that.



```
Protocol  Length Info
TCP           76 54128 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=298809564 TSecr=0 WS
TCP           76 443 → 54128 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=112051269
TCP           68 54128 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=298809591 TSecr=112051269
TLSv1.3      353 Client Hello
TCP           68 443 → 54128 [ACK] Seq=1 Ack=286 Win=30208 Len=0 TSval=112051276 TSecr=298809592
TLSv1.3     1516 Server Hello, Change Cipher Spec, Application Data
TCP           68 54128 → 443 [ACK] Seq=286 Ack=1449 Win=64128 Len=0 TSval=298809619 TSecr=112051276
TCP         2716 443 → 54128 [PSH, ACK] Seq=1449 Ack=286 Win=30208 Len=2648 TSval=112051276 TSecr=298809
TCP           68 54128 → 443 [ACK] Seq=286 Ack=4097 Win=62592 Len=0 TSval=298809619 TSecr=112051276
TLSv1.3     1689 Application Data, Application Data, Application Data
```

Figure 13. TLS handshake and encrypted messages transmission.

We count the TCP and TLS layer bytes in the messages before Server Hello in a session. The distribution is shown in Figure 14, and the size of the bytes is mainly within 5000. According to the previous research on byte selection in the field of deep neural network traffic detection and identification, the number of bytes is usually taken to the power of two. Therefore, within 5000 bytes, n is selected as 512, 1024, 2048, and 4096 bytes. In addition, we also select 8192 bytes in order to minimize information loss. The results are shown in Figure 15.

As shown in Figure 15a,b, the results with byte sizes less than 4096 are close because 512 bytes already contain the Client Hello message, which can achieve high F1-Score, precision, and recall. When $n > 4096$, the overall performance decreases significantly due to the excessive zero padding at a uniform length of 8192 bytes. Figure 15b focuses on the identification of the encrypted DNS covert channel traffic generated by iodine and dnscat2. The use of n =1024 gives the best results; therefore, 1024 is chosen as the value of n.

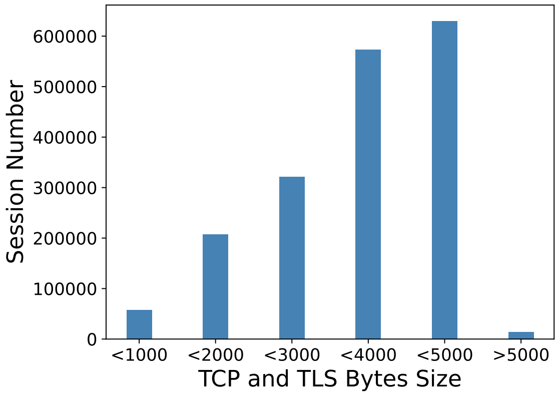


Figure 14. Distribution of TCP and TLS layer bytes.

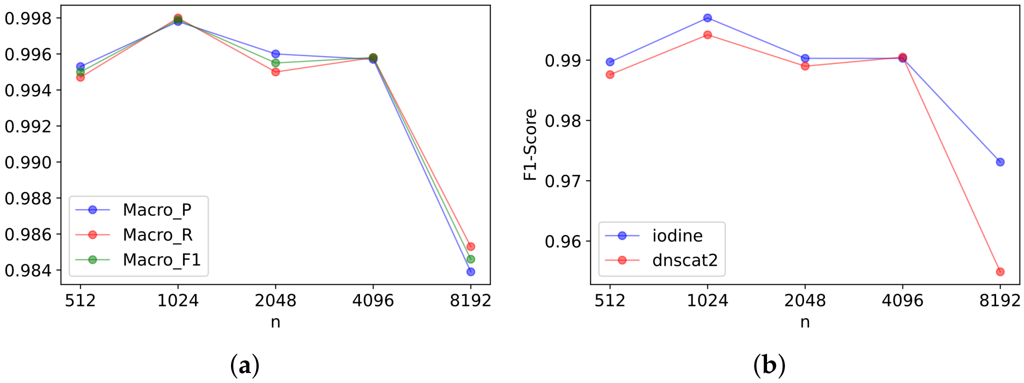


Figure 15. Comparison of results under different bytes size n: (**a**) comparison of macro-averaging under different bytes size n; (**b**) Comparison of F1-Score under different bytes size n in identification of iodine and dnscat2.

6. Conclusions

In this paper, we propose a DoH-encrypted DNS covert channel detection method based on feature fusion called FF-MR to solve the problem of weak single-feature differentiation in existing research. FF-MR extracts TCP transmission features, TLS handshake features, and the local transmission patterns of DoH-encrypted DNS covert channels from session byte sequences using a Residual Neural Network, calculates global correlations with statistical features using a Multi-Head Attention mechanism, and finally, performs weighted fusion. After multiple iterations of the neural network, important features will be given higher weights, which plays a key role in classification. The proposed method's results on the CIC-DoHBrw-2020 dataset show that its macro-averaging precision and recall reach 99.72% and 99.73%, respectively, and its macro-averaging F1-Score is able to reach 0.9978. Compared with existing methods discussed in this paper, FF-MR achieves at most a 4.56% improvement in macro-averaging F1-Score. Moreover, FF-MR demonstrates a better F1-Score than the methods discussed in this paper when identifying two encrypted DNS covert channels, iodine and dns2cat, improving from the highest scores of 0.977 and 0.9869 for other methods to 0.9951 and 0.9954, respectively. The effectiveness of the MHA-Resnet model used in FF-MR is verified from three aspects by comparison with baseline models, and finally, we implemented parameter sensitivity experiments to determine the value of the byte sequence length n. However, due to the complex structure of the model, the real-time performance is poor. Thus, we will take into account the accuracy and real-time performance in future research.

Author Contributions: Conceptualization, C.S., Y.W. and D.H.; methodology, C.S. and Y.W.; validation, C.S., Y.W. and D.H.; formal analysis, C.S., Y.W., X.X. and Y.L.; investigation, D.H.; resources, C.S. and Y.W.; data curation, C.S., Y.W. and X.X.; writing—original draft preparation, C.S.; writing—review and editing, C.S., Y.W., X.X. and Y.L.; visualization, C.S. and D.H.; supervision, C.S., D.H. and Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The CIRA-CIC-DoHBrw-2020 dataset that supports the findings of this study is openly available in DNS over HTTPS (CIRA-CIC-DoHBrw2020) at https://www.unb.ca/cic/datasets/index.html, (accessed on 10 June 2022).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DNS	Domain Name System
DoH	DNS over HTTPS
FF-MR	Feature Fusion based on Multi-Head Attention and Residual Neural Network
MLP	Multilayer Perceptron
DNSSEC	Domain Name System Security Extensions
DoT	DNS over TLS
IoT	Internet of Things
C&C	Command and Control
APT	Advanced Persistent Threat
RF	Random Forest
NB	Naive Bayes
SVM	Support Vector Machines
LSTM	Long Short-Term Memory
ELK	Elasticsearch, Logstash, Kibana
SOC	Security Operation Center
Bi-RNN	Bidirectional Recurrent Neural Network
ET-BERT	Encrypted Traffic Bidirectional Encoder Representations from Transformer
EV SSL	Extended Validation SSL Certificate
Conv1D	One-dimensional Convolutional layer
NLP	Natural Language Processing
ResLayer	Residual Layer
ResBlock	Residual block
BatchNorm	Batch Normalization layer
Avgpooling1D	One-dimensional Averaging pooling
TP	True Positives
TN	True Negatives
FP	False Positives
FN	False Negatives
GBDT	Gradient Boosting Decision Tree

References

1. Meng, D.; Zou, F. DNS Privacy Protection Security Analysis. *Commun. Technol.* **2020**, *53*, 5.
2. Cloudflare. *Dns Over tls Vs. dns Over https | Secure dns*. Technical Report. 2021. Available online: https://www.cloudflare-cn.com/learning/dns/dns-over-tls/ (accessed on 10 June 2022).
3. Bures, M.; Klima, M.; Rechtberger, V.; Ahmed, B.S.; Hindy, H.; Bellekens, X. Review of specific features and challenges in the current internet of things systems impacting their security and reliability. In *World Conference on Information Systems and Technologies*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 546–556.

4. Mahmoud, R.; Yousuf, T.; Aloul, F.; Zualkernan, I. Internet of things (iot) security: Current status, challenges and prospective measures. In Proceedings of the 2015 10th International Conference for Internet Technology and Secured Transactions (ICITST), London, UK, 14–16 December 2015; pp. 336–341.
5. Hesselman, C.; Kaeo, M.; Chapin, L.; Claffy, K.; Seiden, M.; McPherson, D.; Piscitello, D.; McConachie, A.; April, T.; Latour, J.; et al. The dns in iot: Opportunities, risks, and challenges. *IEEE Internet Comput.* **2020**, *24*, 23–32. [CrossRef]
6. Network Security Research Lab at 360. An Analysis of Godlua Backdoor. Technical Report. 2019. Available online: https://blog.netlab.360.com/an-analysis-of-godlua-backdoor-en/ (accessed on 10 June 2022).
7. Cyber Security Review. Iranian Hacker Group Becomes First Known Apt to Weaponize Dns-Over-Https (Doh). Technical Report. 2020. Available online: https://www.cybersecurity-review.com/news-august-2020/iranian-hacker-group-becomes-first-known-apt-to-weaponize-dns-over-https-doh/ (accessed on 10 June 2022).
8. Banadaki, Y.M.; Robert, S. Detecting malicious dns over https traffic in domain name system using machine learning classifiers. *J. Comput. Sci. Appl.* **2020**, *8*, 46–55.
9. Montazerishatoori, M.; Davidson, L.; Kaur, G.; Lashkari, A.H. Detection of doh tunnels using time-series classification of encrypted traffic. In Proceedings of the 2020 IEEE International Conference on Dependable, Autonomic and Secure Computing, International Conference on Pervasive Intelligence and Computing, International Conference on Cloud and Big Data Computing, International Conference on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Calgary, AB, Canada, 17–22 August 2020; pp. 63–70.
10. Al-Fawa'reh, M.; Ashi, Z.; Jafar, M.T. Detecting malicious dns queries over encrypted tunnels using statistical analysis and bi-directional recurrent neural networks. *Karbala Int. J. Mod. Sci.* **2021**, *7*, 4. [CrossRef]
11. Nguyen, T.A.; Park, M. Doh tunneling detection system for enterprise network using deep learning technique. *Appl. Sci.* **2022**, *12*, 2416. [CrossRef]
12. Zhan, M.; Li, Y.; Yu, G.; Li, B.; Wang, W. Detecting dns over https based data exfiltration. *Comput. Netw.* **2022**, *209*, 108919. [CrossRef]
13. Mitsuhashi, R.; Satoh, A.; Jin, Y.; Iida, K.; Takahiro, S.; Takai, Y. Identifying malicious dns tunnel tools from doh traffic using hierarchical machine learning classification. In *International Conference on Information Security*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 238–256.
14. Zebin, T.; Rezvy, S.; Luo, Y. An explainable ai-based intrusion detection system for dns over https (doh) attacks. *IEEE Trans. Inf. Forensics Secur.* **2022**, *17*, 2339–2349. [CrossRef]
15. Ren, H.; Wang, X. Review of attention mechanism. *J. Comput. Appl.* **2021**, *41*, 6.
16. Mnih, V.; Heess, N.; Graves, A.; Kavukcuoglu, K. Recurrent models of visual attention. *Adv. Neural Inf. Process. Syst.* **2014**, *3*, 2204–2212.
17. Zhu, Z.; Rao, Y.; Wu, Y.; Qi, J.; Zhang, Y. Research Progress of Attention Mechanism in Deep Learning. *J. Chin. Inf. Process.* **2019**, *33*, 11.
18. Zhao, S.; Zhang, Z. Attention-via-attention neural machine translation. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
19. Britz, D.; Goldie, A.; Luong, M.T.; Le, Q. Massive exploration of neural machine translation architectures. *arXiv* **2017**, arXiv:1703.03906.
20. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762
21. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Amodei, D. Language models are few-shot learners. *arXiv* **2020**, arXiv:2005.14165.
22. Ming, Y.; Meng, X.; Fan, C.; Yu, H. Deep learning for monocular depth estimation: A review. *Neurocomputing* **2021**, *438*, 14–33.
23. Wang, Y.; Dong, X.; Li, G.; Dong, J.; Yu, H. Cascade regression-based face frontalization for dynamic facial expression analysis. *Cogn. Comput.* **2022**, *14*, 1571–1584. [CrossRef]
24. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
25. Liu, S.; Zhang, X. Intrusion Detection System Based on Dual Attention. *Netinfo Secur.* **2022**, *in press*.
26. Zhang, G.; Yan, F.; Zhang, D.; Liu, X. Insider Threat Detection Model Based on LSTM-Attention. *Netinfo Secur.* **2022**, *in press*.
27. Jiang, T.; Yin, W.; Cai, B.; Zhang, K. Encrypted malicious traffic identification based on hierarchical spatiotemporal feature and Multi-Head attention. *Comput. Eng.* **2021**, *47*, 101–108.
28. Wang, H.; Wei, T.; Huangfu, Y.; Li, L.; Shen, F. Enabling Self-Attention based multi-feature anomaly detection and classification of network traffic. *J. East China Norm. Univ. (Nat. Sci.)* **2021**, *in press*.
29. Wang, R.; Ren, H.; Dong, W.; Li, H.; Sun, X. Network traffic anomaly detection model based on stacked convolution attention. *Comput. Eng.* 2022, *in press*.
30. Lin, X.; Xiong, G.; Gou, G.; Li, Z.; Shi, J.; Yu, J. Et-bert: A contextualized datagram representation with pre-training transformers for encrypted traffic classification. *Proc. ACM Web Conf.* **2022**, *2022*, 633–642.
31. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
32. Ekman, E. *iodine*; Technical Report; lwIP Developers: New York, NY, USA, 2014.
33. Ron. *dnscat2*; Technical Report; SkullSecurity: New York, NY, USA, 2014.
34. Dembour, O. *dns2tcp*; Technical Report; SkullSecurity: New York, NY, USA, 2017.

35. Huo, Y.; Zhao, F. Analysis of Encrypted Malicious Traffic Detection Based on Stacking and Multi-feature Fusion. *Comput. Eng.* **2022**, 142–148.
36. Torroledo, I.; Camacho, L.D.; Bahnsen, A.C. Hunting malicious tls certificates with deep neural networks. In Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security, Toronto, Canada, 15–19 October 2018; Association for Computing Machinery: New York, NY, USA.
37. Pai, K.C.; Mitra, S.; Madhusoodhana, C.S. Novel tls signature extraction for malware detection. In Proceedings of the 2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, India, 2–4 July 2020.
38. Lashkari, A.H. *Dohlyzer*; Technical Report; York University: York, UK, 2020.
39. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
40. Niu, Z.; Zhong, G.; Yu, H. A review on the attention mechanism of deep learning. *Neurocomputing* **2021**, *452*, 48–62. [CrossRef]
41. Van Der Maaten, L.; Hinton, G. Visualizing data using t-sne. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
42. Wang, W.; Sheng, Y.; Wang, J.; Zeng, X.; Ye, X.; Huang, Y.; Zhu, M. Hast-ids: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. *IEEE Access* **2018**, *6*, 1792–1806. [CrossRef]
43. Wang, M.; Zheng, K.; Ning, X.; Yang, Y.; Wang, X. Centime: A direct comprehensive traffic features extraction for encrypted traffic classification. In Proceedings of the 2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS), Chengdu, China, 23–26 April 2021; pp. 490–498.