applied sciences

MDPI

Article

A Comprehensive Review of Tunnel Detection on Multilayer Protocols: From Traditional to Machine Learning Approaches

Zhonghang Sui, Hui Shu, Fei Kang *, Yuyao Huang and Guoyu Huo

State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450000, China
* Correspondence: kfminnie@163.com

Abstract: Tunnels, a key technology of traffic obfuscation, are increasingly being used to evade censorship. While providing convenience to users, tunnel technology poses a hidden danger to cybersecurity due to its concealment and camouflage capabilities. In contrast to previous studies of encrypted traffic detection, we perform the first measurement study of tunnel traffic and its unique characteristics and focus on the challenges and solutions in detecting tunnel traffic among traditional and machine learning techniques. This study covers an almost twenty-year research period from 2003 to 2022. First, we present the concepts of two types of tunnels, broad and narrow tunnels, respectively, as well as a framework for major tunnel applications, such as Tor (the second-generation onion router), proxy, VPN, and their relationships. Second, we analyze state-of-the-art methods from traditional to machine learning applications to systematize tunnel traffic detection, including HTTP, HTTPS, DNS, SSH, TCP, ICMP and IPSec. A quantitative evaluation is presented with five crucial indicators applied to the detection methods and reviews. We further discuss the research work based on datasets, feature engineering, and challenges that have are solved, partly solved and unsolved. Finally, by providing open questions and the potential directions, we hope to inspire future work in this area.

Keywords: cyber security; tunnel detection; network traffic; multilayer protocols; machine learning

check for updates

Citation: Sui, Z.; Shu, H.; Kang, F.; Huang, Y.; Huo, G. A Comprehensive Review of Tunnel Detection on Multilayer Protocols: From Traditional to Machine Learning Approaches. *Appl. Sci.* **2023**, *13*, 1974. https://doi.org/10.3390/app13031974

Academic Editor: Andrea Prati

Received: 22 December 2022
Revised: 29 January 2023
Accepted: 30 January 2023
Published: 3 February 2023

1. Introduction

Tunnel technology offers numerous benefits, such as breaking firewall restrictions, forcing data to a specified address, hiding private network addresses, providing secure encrypted channels, etc. Tunnel technology is widely used [1], with applications including proxy services, remote access, intrusion control, traffic stealing, etc.

By installing tunnel clients, users can indirectly forward data packets by bypassing IP blocks by using the proxy service, which mimics socks5. Remote access means building a negotiatory channel between the private network and the company network and then accessing secret data or applications. Intrusion control mainly includes establishing C&C communication [2], stealing private data, realizing long-term control, guarding, etc. Among the specific methods used are malicious Trojan, Botnet, ransomware, and advanced persistent threat (APT) [3] techniques, etc. In traffic stealing, web traffic is encapsulated in free DNS [4], ICMP, and other data packets to deceive the network billing system. Tunnel technology is advancing, its threats are increasing with time, and the identification and classification of tunnel traffic in a network are especially critical.

Challenges

In order to detect and classify tunnel traffic, we must find discriminative features between normal traffic and tunnel traffic. Recently, in spite of dramatic progress having been made in detecting encrypted traffic, the detection of tunnel remains difficult (or is inaccurate). Actually, tunnel traffic is often encrypted, and so transferring the source problem into the target problem is beneficial. Due to its concealment and camouflage, which are inherently unique, it poses the following challenges to current researchers. These issues are different from those of encrypted traffic detection [5].

We must identify whether it is tunnel traffic. A network communication protocol is easy to identify; however, identifying whether it is encapsulated as a tunneling protocol is much more difficult. Additionally, most traffic is encrypted, and the encryption algorithms obscure plaintext features, thereby reducing the randomness of the original message to a great extent. Furthermore, some software can mask tunnel traffic characteristics as normal traffic characteristics by morphing and padding packets, which makes identification more difficult.

We must identify a passenger protocol for the tunnel. To avoid censorship, tunnel traffic may be mixed between multiple passenger protocols, such as FTP-DNS, DNS-HTTPS, TELNET-HTTP, and SMTP-TLS. To identify each passenger protocol in the specific tunnel, it is necessary to have an accurate command of multilayer protocols so that lower traffic heterogeneity within the same tunnel renders the tasks more complex.

We must identify applications and behavior in tunnel traffic. Further, there are several specific granularities used to identify different applications and their behaviors in tunnel techniques. These include ongoing applications, service benign/malicious behaviors of the same application, webpages, and content parameters related to the same webpage, etc. A tunnel's traffic is extremely homogeneous, and the application traffic in that tunnel is the same quintuple, so determining its beginning and ending times is difficult, and tunnel noise also makes fine-grained identification difficult.

Contributions

To clarify the search status of solving the problems and challenges mentioned above, we have made two contributions.

- The first thing we have done is focus on three challenges of tunnel traffic detection from the perspective of protocol classification within the TCP/IP protocol stack, which is different from the closely related topics of encrypted traffic detection in detail. A new method of tunnel traffic identification and classification is presented based on the combination of traditional and machine learning detection methods within seven protocols: HTTP, HTTPS, DNS, SSH, TCP, ICMP and IPSec. We use the thesis database (scopus database, occasionally assisted by web of science) and use a certain keyword (such as HTTP/HTTPS/DNS/TCP/ICMP/IPSec tunnel traffic detection, traffic detection, etc.) to search and determine a large range of initial related literature review papers (LRPs) [6]. The simplest way is to filter the paper′ date (we have been looking for papers for nearly 20 years), publishing platform, field and so on with the filter that comes with the database, including journal article and conference paper. By browsing abstracts, keywords and the objective statements of pieces of literature, we can determine their relevance. Then, the results of rough reading, based on inductive coding and of intensive reading, based on co-occurrence analysis, are presented. The papers are selected based on three main criteria: whether they (i) provide new techniques or ideas on tunnel detection, (ii) i they have a high degree of completion and reproducibility, and (iii) if they have an ability to solve three challenges above.
- Second, five evaluation indices are used to evaluate the three typical tunnel detection methods from over the past ten years using AHP (analytical hierarchy process). Additionally, we also conclude by exploring and analyzing the direction of tunnel identification and detection. Figure 1 summarizes our work.

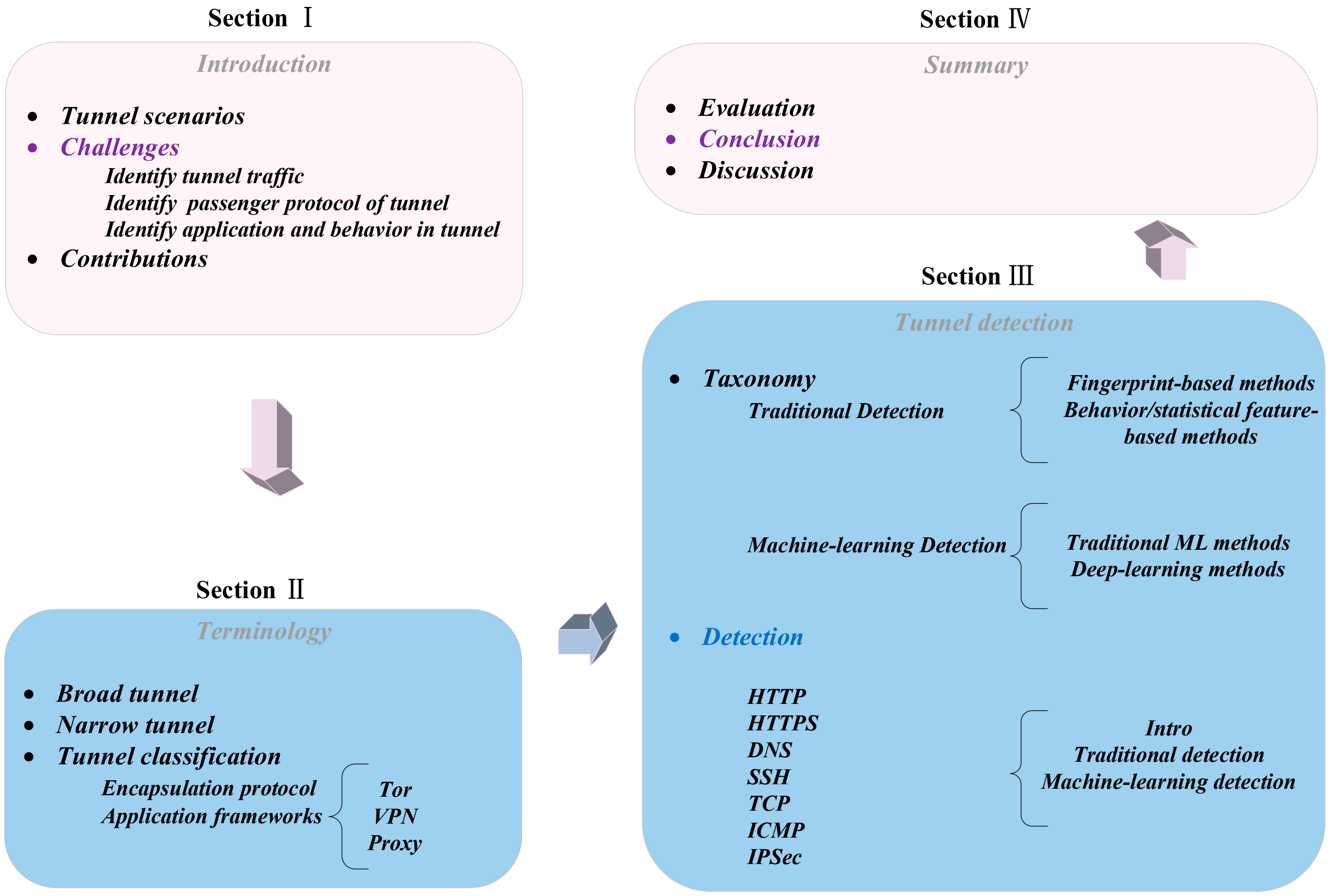


Figure 1. Our review of tunnel detection process.

Terminology

We first introduce the broad and narrow concepts of tunnel technology. We then arrange them as protocols and mainstream application frameworks, including anonymous networks, proxies, and VPN technology.

1.1. Broad & Narrow Tunnel

Broad Tunnel. Simmons [7] proposed a general tunnel as a classic prisoner communication model in 1983 and described a network tunnel in detail. As shown in Figure 2, the OSI computer network model is based on a classic prisoner communication model. In this model, Alice and Bob communicate through two networked machines. Despite the fact that their communication data appears to be sent through an open communication channel, they can create a private tunnel that is only visible to them. This generalized tunnel is formally defined as $\|A \circ B\|$. For any kind of tunnel $t_i, \forall t_i \in \|A \circ B\|, i \in R$, there is $t_i = \begin{cases} \|A \circ B\|_T \\ \|A \circ B\|_S \end{cases}$.

$\|A \circ B\|_S$ represents the hidden tunnel of network storage. As the most commonly used tunnel, this tunnel is also the tunnel most discussed in our paper because it changes the header fields and payloads of various network protocols to hide information. Both communication parties will only deal with keywords and optional parameters with this tunnel, specified ahead of time, and any keywords or optional parameters not specified in the protocol will be automatically discarded. It is also beneficial for covert communication to hide more information in the message payload.

$\|A \circ B\|_T$ is a network time covert tunnel which encodes covert information based on a control time interval. For example, data packets' delivery times are directly changed, but data packet timestamps are changed indirectly. In this tunnel, data packet arrival time depends on network stability. As time jitter, packet disorder, and loss often lead to decoding failure, we are not focusing on these tunnels.

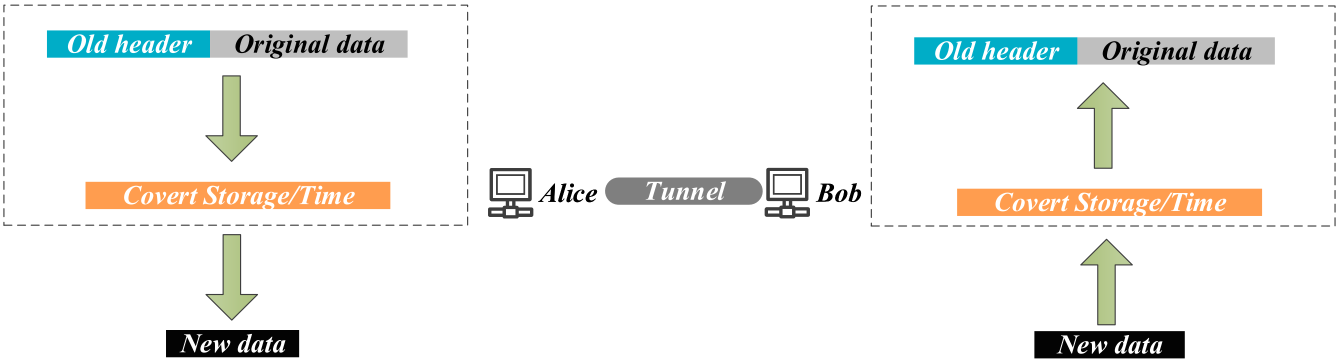


Figure 2. Broad Tunnel.

Narrow Tunnel. In a narrow tunnel, the passenger protocol is encapsulated in a payload and transmitted to the channel. This is purely responsible for the protocol encapsulation and decapsulation of data packets like the edge of the network topology. Our paper formally defines this narrow tunnel as $|A \circ B|$ in Figure 3. Moreover, the encryption of $|A \circ B|$ is more prevalent.

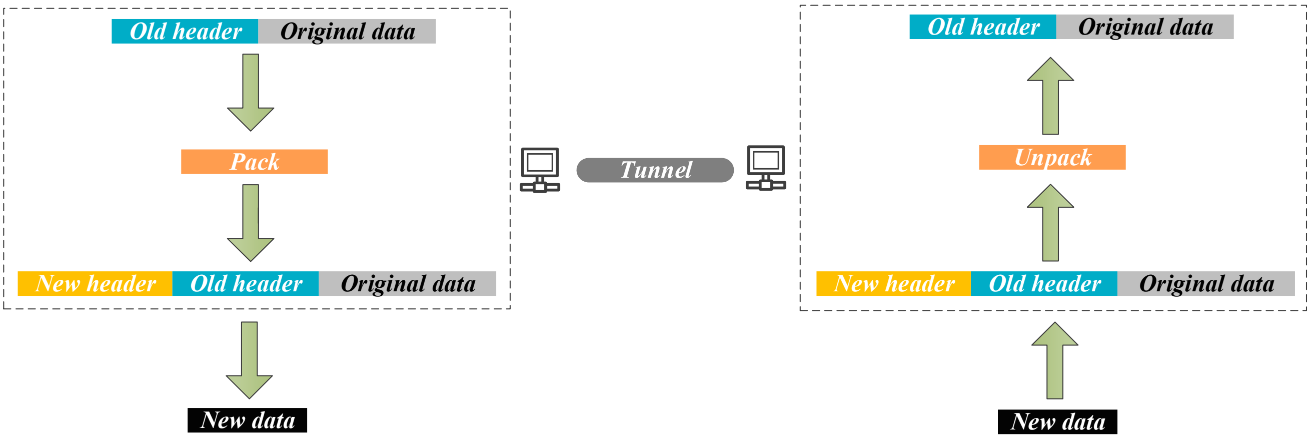


Figure 3. Narrow tunnel.

1.2. Classification

1.2.1. Encapsulation Protocol

Using the five-layer model of the TCP/IP protocol stack in Table 1, we propose a method of dividing tunnel traffic from protocol classification at each layer.

Table 1. Tunnels classified by TCP/IP protocol stack.

TCP/IP Protocol Stack	Types of Tunnels According to Protocol
data link layer	L2F tunnel, PPTP tunnel, L2TP tunnel
network layer	**ICMP tunnel, IPSec tunnel**, GRE tunnel, VTP tunnel
transport layer	**TCP tunnel**, UDP tunnel
application layer	**HTTP tunnel, HTTPS tunnel, DNS tunnel, SSH tunnel**

1.2.2. Tunnel Application Frameworks

Anonymous Networks

An anonymous network protects the privacy of users' communication in an open network setting. Due to its enhanced security, deployment, availability, and flexibility, Tor, a typical application of a P2P [8,9] structure, is widely used by anonymous users. In this scheme, data packets are forwarded at the entry node and bundled into nested layers of encrypted packets of similar size at the entry node. After routing and decrypting these data packets, the data are sent to a destination node.

Proxy

Proxy is a typical representative of $|A \circ B|$, transferring data between client and server. Client requests are sent to proxy servers, which access the targeted website instead of the client, load its content, and then transmit the loaded content back to the client. As a result of Chrome browser's data compression proxy technology, web page loading speeds can be increased, which reduces bandwidth consumption.

VPN

In VPN, the initial plaintext data (including the data to be transmitted, source/destination IP) are encrypted by the VPN client and then a signature is attached. To re-package the data packet, a new data header is added (including the new IP address, the VPN device's security information, and some initialization parameters) and it is sent via a "specific path" over the public network. When data arrives at VPN server, they are unsealed and decrypted after checking signature.

Relationships

Tor and VPN differ in how they guarantee data anonymity. As a result of the fixed IP address between the VPN client and the VPN server during VPN communication, both client and server can tell which real IP address is sending which data packet to what destination address. However, Tor does not rely on a single server. Rather, it chooses a random relay node and it updates more frequently than VPN. Thus, all nodes in the path cannot obtain the information completely. As a result, Tor can be considered a distributed VPN network if only its start and exit nodes are considered.

Whether to secure data transmission is the difference between a proxy and a VPN. The final effect of using a user VPN or proxy server is the same, i.e., hiding one's real IP from the website or server of interest. Proxy servers are suitable for users who wish to access websites blocked by their region but do not want to conceal their operations on the Internet. When users need to visit a website for a long time or perform sensitive operations, such as sending personal information over open Wi-Fi, VPNs are typically used.

2. Taxonomy

From a traffic fingerprint perspective, plaintext tunnel traffic may contain a specific string. Statistically, even encrypted tunnel traffic has different temporal and spatial characteristics than normal traffic. As shown in Table 2, traditional methods of tunnel traffic detection and current popular methods based on machine learning are summarized in this section.

2.1. Traditional Detection

As tunnel traffic becomes increasingly difficult to identify using the port [10] alone as the criterion, fingerprint-based methods as well as behavior feature (statistical feature) methods will become more popular for unencrypted tunnel traffic identification [11].

In a data packet, fingerprints are non-random features, such as keywords found in DNS request fields QNAME and RDATA, as well as headers like Host, Connection, and Content-type in HTTP requests. A variety of fingerprint-based detection technologies are available, including threshold matching and MD5 matching. In these simple judgment methods, we can quickly and accurately identify when the database capacity is adequate, otherwise, we are highly likely to make mistakes. In text-like data analysis, N-gram word frequency models (also known as unigrams and bigrams) and implicit Markov models

(HMMs) are important. Combined with the Markov property, we can identify tunnel traffic by predicting the value of normal traffic based on a specific string in a fixed field.

As a representation method, behavior features are detected by using the data element information of statistical data packets/streams (hereinafter called statistical features). Statistics (average, maximum, minimum, variance, etc.) are used in probability statistics, statistical distributions (Poisson distribution, Zipf distribution, etc.), and cluster analyses. An abnormal user behavior model can be observed from data analysis by comparing its threshold with a normal behavior model.

2.2. Machine Learning Detection

Since tunnel technology is densely cultured and complex, fingerprint-based methods have required maintaining a huge database with limited patterns, and statistical feature methods have not been able to deal with traffic camouflage dynamically in recent years. Tunnel traffic detection has gained new vitality with the advent of machine learning (ML).

There are three types of ML: supervised, unsupervised, and semi-supervised. In supervised learning, accuracy is measured by constantly comparing prediction results to training data. Supervised learning includes decision trees such as C4.5, RF (random forest), SVM (support vector machine), etc. Additionally, its training process requires a great deal of manual labels and selecting features is highly dependent on expert knowledge. Unsupervised learning, also known as clustering, distinguishes between categories based on unlabeled data in large datasets. In reality, this method is simpler, but its accuracy is not high due to the large amount of unlabeled raw tunnel traffic in real networks. Moreover, semi-supervised learning, which combines advantages from both supervised and unsupervised learning, is more accurate than unsupervised learning, and manual involvement is reduced to some extent.

Since deep neural networks do not extract features in the training process, which can automatically learn high-dimensional abstract data (called E2E), they are extremely popular for the classification of tunnel traffic on large scales. RNN (recurrent neural network), LSTM (short- and long-term memory networks), etc., can handle the temporal features of tunnel traffic well, CNN (convolutional neural networks) can handle the spatial features of tunnel traffic well, and the GAN (generative adversarial network) can be used to alleviate class imbalance in datasets. A deeper and more complex network model structure also means that parameter adjustment is more time-consuming.

Table 2. Traditional and ML models of tunnel detection.

Traditional		**ML**		
Fingerprint	**Statistic**	**Supervised**	**Unsupervised**	**Semi-Supervised**
■Match [12–24]	🟦Markov [25,26]	🟨SVM [16,21,27–33]	🟧CAE [34]	🟧VAE [35]
■N-gram [28,36,37]	🟦MaMPF [38]	🟨Decision Tree (C4.5/5.0) [29,32,33,39–43]	🟧OLDBSC [44]	🟧Bi-GRU [35]
■MD5[24]	🟦KNN [40,43,45,46]	🟨RF [29,37,43,47,48]		
■Degree Distribution [49]	🟦Naïve Bayes [17,27,32,48]	🟨LSTM [50,51]		
	🟦MNB [52]	🟨Bi-LSTM [53,54]		
	🟦*k*-means [55–57]	🟨CNN [34,50,58,59]		
	🟦TF-IDF [60]	🟨Linear Regression [29]		
	🟦GP [42]	🟨MLP [51]		
		🟨AdaBoost [42,61,62]		
		🟨RIPPER [61]		
		🟨RBFN [43,57]		

3. Encapsulation Protocol

In this section, tunnel traffic detection methods are systematically classified to focus on the above three challenges according to the five-layer model of the TCP/IP protocol stack, including HTTP, HTTPS, DNS, SSH, TCP, ICMP and IPSec.

3.1. HTTP

Intro

The existing forms of $\|A \circ B\|_S$ include a URL-based HTTP tunnel and an HTTP header field-based HTTP tunnel. The optional segments of HTTP messages can be altered and confused. For example, headers such as Host, Connection, Content-Type, etc., in HTTP requests can be filled out and confused. Attackers use the referer field to construct covert tunnels since it has a fixed function in normal communication. Common tunnel tools include Neo-reGeorg, Frp, HTTPTunnel, etc.

3.1.1. Traditional Detection

Fingerprint

The goal of most papers (in Table 3) is to determine whether the traffic is tunnel traffic or not, and so signature-based fingerprint detection is of particular importance among them. For the detection of suspicious strings in the payload, Dharmapurikar [12] generated hash values for strings using the same hash function. In order to identify tunnel traffic, it must maintain a large database and have a high false negative rate. The model can detect 10,000 predefined fine strings, but has a high false negative rate. In Wang et al. [36], two innovations were presented: Wang's N-gram(N = 1) word frequency model calculates the average frequency of ASCII characters as one of tunnel traffic's characteristics. First, the model is made more robust by combining it with Mahalanobis to realize unsupervised learning. Second, "Z-String" derived from byte distribution is used as a signature to represent payload, which can be stored quickly in the real-time distributed detection system. Almost a 100% detection rate and about a 0.1% false alarm rate are achieved for HTTP tunnel traffic in DARPA'99 [63].

It is also common for tunnel traffic to be characterized by detecting the header of HTTP packets. If the user agent is filled, or if they are filled by an unknown browser, this method can also be used. Using Bortolameotti's [13] model, top-level and second-level domain names, constant header fields, and language fields in Host are extracted, and a module is added for dynamically updating the fingerprint database, making it more flexible. During private datasets, the model achieves a false alarm rate of 0.9%, with a detection accuracy of 97.7% on average.

Statistic Feature

As well as identifying tunnel traffic, statistical flow characteristics also describe how tunnel traffic is classified from applications and behaviors. The temporal and spatial features of a data stream are often determined by the packet size and the inter-arrival time (IAT) between successive packets. Besides these factors, various pieces of statistical information on packet length and packet intervals, such as the estimation of a round-trip time, the size of the TCP segment, and the total number of retransmissions, and simple statistical knowledge such as average, median, maximum, minimum and variance are used by researchers to create models. To determine whether traffic is tunnel traffic, Li [64] utilized hierarchical clustering technology and a scoring mechanism model to establish a normal behavior clustering model by comparing the characteristics of normal HTTP traffic, including packet/stream time intervals. Despite being able to achieve an accuracy of more than 93.9%, this model is heavily influenced by the long HTTP traffic that transmits large files. As a measure of the randomness of data, the entropy criterion is often used to distinguish tunnel traffic, and the entropy value in tunnel traffic differs from that in normal traffic. Nasseralfoghara [14] used the entropy of message exchange traffic. The HTTP protocol request text should contain the address of the requested page, the required method, and the parameters for that method. If its entropy value exceeds its predetermined threshold, it is classified as HTTP tunnel traffic. However, if the unintentional increase in noise changes the entropy, it incorrectly classifies the traffic as HTTP tunnel traffic.

The outbound bandwidth usage in tunnel traffic will differ from normal HTTP requests. In addition to counting outbound bandwidth usage, long streams account for a substantial share of tunnel traffic, which is important. Piraisoody [27] introduced the concept of "occupancy rate" as a key element in the classification process, and proposed the definition of "stream groups". HTTP tunnel applications such as audio, video, and file transfer software are classified based on parameters such as download rate, bytes per stream, bytes per packet, and "occupancy rate," with a high accuracy rate of at least 70% in private datasets.

3.1.2. Machine Learning

To avoid the problem that the exponential growth of feature dimensions makes the accuracy of the N-gram model lower when n > 2, Perdisci [28] used a feature clustering algorithm to reduce the dimensions in different feature spaces and multiple single-class SVM classifiers to vote in different feature spaces, making the model extremely accurate at detecting HTTP tunnel traffic. Thus, the author achieved an FPR of 10^{-5} in DARPA'99 data [63]. The URL and payload string were extracted using a deep neural network model built with Bi-LSTM (bi-directional long and short-term memory). In private datasets, experiments [53] have shown that the accuracy of identifying tunnel traffic can reach 90%. With CNN and LSTM, Wong [50] developed a deep learning model for detecting URL-based HTTP tunnels. In privately balanced datasets, the best model averages 95% accuracy. However, the model it cannot handle seriously unbalanced datasets.

He [54] combined classical classifiers with bagging, boosting, etc., with classical classifiers such as C4.5. The packet payload size, variance, packet count, data stream length, and IAT of adjacent packets can be used to identify whether the traffic is HTTP tunnel traffic.

It is also possible to classify passenger protocols in tunnel traffic using detection methods based on ML. Based on private datasets, Ding's [39] method can identify tunnel traffic and distinguish SMTP-HTTP and P2P-HTTP passenger protocols with 95% accuracy. What is remarkable is that the model is 100% accurate at identifying gray pigeon traffic. According to Wang [16], it is necessary to analyze the second-order statistical correlations between consecutive packets, including the packet distribution difference in HTTP sessions and the entropy in N-RPP and N-RMI distances. An SVM classifier can distinguish tunnel traffic from non-tunnel traffic and identify FTP-HTTP, SMTP-HTTP, and POP3-HTTP passenger protocols. HTTP tunnel traffic is recognized with an average accuracy of 82.5% on private datasets.

3.2. HTTPS

Intro

HTTPS protocol adds an SSL/TLS layer between the transport layer and HTTP protocol, supporting encryption, authentication and integrity checking, and which is mainly used to realize secure HTTP data transmission. As opposed to HTTP tunnel traffic detection, HTTPS traffic is encrypted, and the hidden message content makes fixed string matches and character statistics ineffective. This section emphasizes the spatial–temporal characteristics of plaintext packets as well as their interaction during SSL/TLS handshakes. There are two tunnel modes of HTTPS, the first one is $|A \circ B|$, and the second one is VPN, i.e., a typical HTTPS tunnel, which is an encryption $|A \circ B|$. We also separately summarize the methods of ML used to detect VPN traffic. Tools that can be used to build HTTPS tunnels include abptts, Shadowsocks Obfs, reGeorg, etc.

Table 3. Summary of HTTP tunnel detection. T represents traditional method. ML represents machine learning method. I represents detecting with coarse content. II represents detecting in a specific location.

Ref.	Year	Datasets	T/ML	Fingerprint	Statistic	Flow	Packet	Spatial	Temporal	I	II	Model
Dharmapurikar [12]	2003	Private	T	√		√		√		Payload	Strings	Match
Wang [36]	2004	Public	T	√			√	√		Payload	Strings	N-gram + Mahalanobis
Perdisci [28]	2009	Public	ML	√			√	√		Payload	Strings	N-gram + SVM
Ding [39]	2011	Private	T		√	√	√	√	√	Payload	Size + Duration	C4.5
Wang [16]	2013	Private	ML		√	√		√	√	Payload	Size + Duration	Match + SVM
Piraisoody [27]	2013	Private	T		√	√	√	√	√	Payload	Size + Duration + occupancy	Naïve Bayes + SVM
Li [64]	2014	Private	T		√	√	√	√	√	Payload	Size + Duration	Hierarchical Clustering + Scoring Mechanism
Bortolameotti [13]	2017	Private	T	√			√	√		Head	Host + Language + User-Agent	Match
Yu [53]	2018	Private	ML		√	√			√	Payload	Strings	Bi-LSTM
He [54]	2019	Private	ML		√	√			√	Payload	Duration	Bi-LSTM
Wong [50]	2019	Private	ML		√	√			√	Field	URL	CNN + LSTM
Nasseralfoghara [14]	2020	Private	T		√		√	√		Head	Request Method + URL	Match

3.2.1. Traditional Detection

Fingerprint

For $|A \circ B|$, an SSL/TLS handshake fingerprint (including supported cipher suites, TLS extensions, etc.) is mostly focused on detecting tunnel traffic in HTTP header (as shown in Table 4). According to Durumeric et al. [15], certificate authority-based detection can be improved by examining the trust relationships between root/intermediate authorities and leaf certificates extracted by the Web server.

Statistic Feature

Besides identifying tunnel traffic, classifying the applications in tunnel based on features such as the length of certificate packets, IAT, and an SSL/TLS handshake session is popular. In Herrmann's [52] model, traffic features were only determined by the normalized frequency distribution of IP packet sizes in the stream using Multinomial Naïve Bayes (MNB). As a result of TF-IDF transformation, HTTPS tunnel traffic can be identified with 97% accuracy in private datasets. Korczyński [25] labeled SSL/TLS handshake state, established first-order homogeneous Markov fingerprints and judged 12 applications within HTTPS tunnel traffic based on their convergence toward the normal traffic state. In private datasets, the accuracy rate is above 97%, but two disadvantages occur with this method. First, IP can only be resolved into a domain name model through DNS, and second, the efficiency and accuracy of this method will decrease as packet numbers increase.

In order to increase the diversity and multi-attribute quality of traffic characteristics, Sun [17] used a hybrid method in the first byte of handshake-based messages to identify HTTPS tunnel traffic. With fine granularity, Liu [38] proposed using Markov Probability Fingerprints (MaMPF) to identify tunnel traffic applications. MaMPF can calculate the power law distribution and relative probability of all 18 applications modeling for each application in order to avoid the overfitting caused by excessive packet length. MaMPF achieves 96.4% TPR and 0.2% FPR in private datasets.

3.2.2. Machine Learning

As part of the detection methods, ML includes not only detecting tunnel traffic, but also identifying the application involved. Specifically, Draper-Gil [40] and Anderson [29] examined spatio-temporal characteristics, including in/out bytes, packet sizes, packet numbers, packet length, time series, distribution of bytes, etc.

It has always been important to monitor VPN traffic under HTTPS tunnels due to its role as an indicator of HTTPS tunnel traffic. Draper-Gil [40] used only time-related features along with C4.5 and KNN to achieve an accuracy rate of over 80% in the ISCXVPN2016 public datasets [65]. Guo [34] proposed two models based on deep learning: CAE (convolutional automatic coding) and CNN, which separated traffic into VPN and non-VPN traffic, and further identified VPN traffic generated by six different applications. By using the unsupervised algorithm of CAE, they can extract the hidden layer features from the traffic samples to generate conversation images. CNN excels at extracting two-dimensional local features. The CAE-based model has the best recognition effect in the selected datasets, with an overall recognition accuracy rate of 98.77%; CNN is the best for all six types of application traffic. In ISCXVPN2016, Parchekani [51] classified VPN and non-VPN traffic based on E2E with MLP and LSTM, achieving an overall recognition accuracy rate of 92.92%.

3.3. DNS

Intro

In a DNS tunnel, the DNS server checks its database for the address to be resolved when it receives a DNS request. If no records are found in the database, the server sends the request to the specified domain. According to known analysis of the DNS protocol, the QNAME field in the query area and the RDATA field in the response area are the highest frequency areas where features are embedded. In terms of analyzing $\|A \circ B\|_S$ from DNS traffic, extracting resource records such as TXT, A, AAAA, and MX and observing the access count of resource records are common methods.

The parser depends on the number of resource records specified in the header to determine the parsed data, and the last record is considered to have reached the end. As a result, DNS packets can be tipped with any amount of data, and RawUDP is where the information is embedded at high frequency. The statistical distribution of DNS tunnel traffic is very different from that of normal domain. The former obeys random distribution, while the latter obeys Zipf distribution. A number of studies have been conducted on detecting DNS tunnels by analyzing the statistical information in DNS packets, such as domain name length and entropy. Tools such as NSTX, DNSCat, and Iodine have been used to detect DNS tunnels.

Table 4. Summary of HTTPS tunnel detection. Summary of HTTPS tunnel detection. T represents traditional method. ML represents machine learning method. I represents detecting with coarse content. II represents detecting in a specific location.

Ref.	Year	Datasets	T/ML	Fingerprint	Behavior	Flow	Packet	Spatial	Temporal	I	II	Model
Herrmann [52]	2009	Private	T		√	√		√		Payload	Size	MNB
Sun [17]	2010	Private	T	√	√	√		√	√	Handshake + Payload	Size + Duration	Match + Bayes
Durumeric [15]	2013	Private	T	√		√		√		Handshake	Entities	Match
Korczyński [25]	2014	Public	T	√		√		√		Handshake	ClientHello	Markov
Draper-Gil [40]	2016	Public	ML		√	√			√	Payload	Duration	C4.5 + KNN
Anderson [29]	2017	Private	ML		√		√	√		Payload	Size + Duration	Linear Regression + L1/L2-logistic regression + RF + SVM + Decision Tree + Multi-layer Perceptron
Liu [38]	2018	Private	T		√	√			√	Payload	Duration + Order	MaMPF
Guo [34]	2020	Public	ML		√	√	√	√		Payload	Size + Duration	CAE + CNN
Parchekani [51]	2020	Public	ML		√	√	√	√	√	Payload	Size + Duration	MLP + LSTM

3.3.1. Traditional Detection

Fingerprint

In traditional detection methods (in Table 5), most researchers focus on whether it is tunnel traffic or not. Because most DNS tunnel detection data are text-like, word frequency analysis is a good starting point. The statistical distributions of domain names in data packets were analyzed by Burghouwt [49] as features for detecting and identifying DNS tunnel traffic. N-gram (N generally takes 1 or 2) word frequency modeling is used to identify tunnel traffic. The abnormal degree distribution of the visited domain in the message was detected by Burghouwt [49] without requiring any statistical information about message content and traffic. Using graph theory, the author calculated the degree distribution of domains based on the number of computers connected to different domains over a specified time period. Abnormal domains can be distinguished from normal domains to identify DNS tunnels, and their FPR is 0.073% in private datasets.

Statistic Feature

There has been a significant interest in detecting DNS tunnel traffic based on the time and space statistical characteristics of packets/streams, such as the number of packets/streams, the number of bytes, and the duration of streams. These features have been used by Marchal [55]. The average TTL value of domain records, total number of domain name requests during the observation period, request ratio in each time period, whether the domain name is blacklisted, and other characteristics were also extracted by Marchal [55].

There is also a high degree of entropy detection in DNS tunnel traffic. Meanwhile, Karasaridis [18] determined whether the packet size distribution was met with cross entropy, and then defined a distance function to measure entropy, which controlled the threshold range to identify DNS tunnel traffic effectively.

3.3.2. Machine Learning

Most of machine learning detection has been conducted to detect whether traffic is tunnel traffic, while Wang [48] introduced eight features to analyze five kinds of DNS tunnel behaviors. Bilge [41] has integrated the above statistical features. To describe different attributes of DNS names and how they are queried, Bilge [41] used 15 DNS traffic features, nine of which were not previously proposed, including time-based features, DNS response-based features, TLL change features, and domain name features. When combined with a decision tree, this method can identify DNS tunnel traffic accurately, resulting in fewer false alarms than 1% and a 98% accuracy rate.

Wang [60] and Palau [58] developed a way to solve domain name text detection lightly to reduce storage overhead and computational complexity. A domain name specificity score, enhanced TF-IDF, and other algorithms were used by Wang [60] to detect DNS tunnel traffic in private datasets, with 99.92% accuracy in detecting DNS tunnel traffic. It is worth noting that Palau [58] proposed a detection method based on CNN with a simple structure called 1D-CNN that detects 99% of normal domains and 92% of tunnel domains, even though its structure is simple. D'Angelo [59] also used 2D-CNN with a straightforward network structure to detect tunnel traffic. They used a 24-dimensional matrix to represent 22 different features, including request/response type, etc.

Siby [37] focused on DoH [66] (DNS over HTTPS) in order to detect encrypted DNS tunnel traffic because traditional website fingerprint features are insufficient for describing DoH traffic. As a result, when it is combined with RF, it introduces N-grams with TLS record lengths as new features and is able to identify DNS tunnel traffic with 84% accuracy in private datasets. A variational autoencoder (VAE) [67] was proposed by Ding [35] as an E2E model for learning long sequential and structural information beyond the capability of traditional machine learning methods. VAE used bidirectional Gated Recurrent Units (Bi-GRU) as encoders and decoders to automatically extract latent feature representations from raw traffic. Semi-supervised training is used to train the model on normal traffic patterns. The accuracy rate is over 99% on the CIRA-CIC-DoHBrw-2020 dataset [68]. Moreover, the complexity of their model from the number of parameters and consuming time are all obviously less than others [69].

3.4. SSH

Intro

The SSH protocol provides users with secure remote login or other network services through encryption in unsecured networks. As a result, plaintext features are made obsolete in a dense stream of information. To identify SSH tunnel traffic, statistical features and machine learning methods have become mainstream, along with the identification of passenger protocols, applications, or services.

3.4.1. Traditional Detection

Statistic Feature

It is possible to identify SSH tunnel traffic among the detection methods (in Table 6), as well as to classify passenger protocols and applications in traffic using statistical features. With fine granularity, Alshammari [61] and Maiolini [56] classified SSH tunnel traffic applications and services. Ashammari [61] used AdaBoost and RIPPER to identify SSH tunnel traffic with 95% accuracy in public datasets. The classification covers 11 applications/services, including local tunnels, remote tunnels, FTP, and Shell, with the accuracy of 99% and false alarm rates of 0.7%. According to Maiolini [56], the *k*-means clustering analysis is used for real-time traffic classification, which results in an accuracy of 99.5% for SSH tunnel traffic and 99.88% for SCP-SSH, SFTP-SSH, and HTTP-SSH passenger protocols in tunnels. Although these two methods are lightweight, their disadvantages include their confusion when there are similar applications or services in the tunnel traffic, such as HTTP and FTP, and the model will become less accurate as more protocols are added.

3.4.2. Machine Learning

In addition to detecting passengers, ML is capable of classifying their protocols and applications. Agghey [43], Alshammari [42], Jian [44], Pradhan [57] and Hynek [62] have all selected features to train from including packet size, forward/backward average IAT of the first few packets of flows. Without using IP, port number, and payload, Alhammari [42] used C4.5, genetic programming (GP) and AdaBoost classifiers to classify SSH tunnel encrypted traffic. Experiments are conducted in public and private datasets using 39 packet header features (IP header length, checksum, etc.) and 22 flow-based features. In the best case, GP can achieve 98% accuracy, and it can also be classified by applications in tunnel traffic such as SCP, SFTP, Shell, X11 session, Telnet, etc. At its best, C4.5 can achieve 100% accuracy. An unsupervised clustering algorithm On-Line Density-Based Spatial Clustering (OLDBSC) was proposed by Jian [44] to resolve the problems of large computation and high memory consumption. Instead of clustering the entire stream's features, the best priority feature algorithm is used to find an optimal feature set for a sub-stream and then to map the applications that have the highest probability to the application types using the best priority feature algorithm. This method is capable of identifying SSH tunnel traffic as well as classifying applications and identifying unknown application types in traffic. The accuracy rate in private datasets is as high as 99%.

3.5. TCP

Intro

TCP tunnel is a common form in the transport layer. Based on $\|A \circ B\|_S$, it is possible to hide information in IHL, checksum, ISN, and IP ID field. In addition, $\|A \circ B\|_T$ is also used as a common form of TCP tunnel by changing the time interval sequence of packets, network jitter and network delay.

3.5.1. Traditional Detection

Fingerprint

Most fingerprint detection methods (as shown in Table 7) analyze tunnel construction in a theoretical manner, without an actual detection model. As such, they can only determine whether it is tunnel traffic. According to Zseby [19], TCP acknowledgment and sequence number fields and ISNs are common methods for building covert tunnels.

Statistic Feature

Statistical feature detection methods can identify both tunnel traffic and passenger protocol. Gianvecchio [20] detected hidden time channels based on entropy and modified conditional entropy. The author focused on four typical $\|A \circ B\|_T$: IPCTC, TRCTC, MBCTC and JitterBug. To detect these above, they combined fine-binned and coarse-binned estimation of corrected conditional entropy. The corrected conditional entropy can detect the $\|A \circ B\|_T$ with abnormal regularity, while the entropy test can detect the hidden time tunnel with small changes throughout the distribution. It is possible to achieve 100% detection using a combination of the two methods. To detect hidden communication in TCP flows under passenger protocols such as HTTP, FTP, TELNET, SSH and SMTP, Zhai [26] proposed a detection method based on maximum a posteriori probability MAP, a Markov chain description of TCP handshake behavior. For small traffic windows of private datasets, the algorithm is 100% accurate.

Table 5. Summary of DNS tunnel detection. T represents traditional method. ML represents machine learning method. I represents detecting with coarse content. II represents detecting in a specific location.

Ref.	Year	Datasets	T/ML	Fingerprint	Statistic	Flow	Packet	Spatial	Temporal	I	II	Model
Burghouwt [49]	2010	Private	T	√			√	√		Payload	Domain	Degree distributuion
Bilge [41]	2011	Public	ML		√	√	√	√	√	Payload	Domain + TTL + Duration + Answer	Decision Tree
Marchal [55]	2012	Private	T		√	√	√	√	√	Payload	Domain + TTL + Duration	k -means
Wang [60]	2019	Private	ML		√		√	√		Payload	Domain	TF-IDF
Siby [37]	2019	Public	ML		√	√		√		Handshake	TLS	N-gram + RF
Palau [58]	2020	Public	ML		√	√		√	√	Payload	Domain	CNN
Ding [35]	2021	Public	ML		√	√		√	√	Payload	Size + Duration	VAE + Bi-GRU
D'Angelo [59]	2022	Private	ML		√		√	√		Header + Payload	Size + Record	CNN
Wang [48]	2022	Private	ML		√	√	√	√	√	Header + Payload	Domain + Size + Duration	RF + KNN

Table 6. Summary of SSH tunnel detection. T represents traditional method. ML represents machine learning method. I represents detecting with coarse content. II represents detecting in a specific location.

Ref.	Year	Datasets	T/ML	Fingerprint	Statistic	Flow	Packet	Spatial	Temporal	I	II	Model
Alshammari [61]	2007	Public	T		√	√	√	√	√	Payload	Size + Duration	AdaBoost + RIPPER
Maiolini [56]	2009	Private	T		√	√	√	√	√	Payload	Size + Duration + Direction	k -means
Jian [44]	2010	Private	ML		√	√	√	√	√	Payload	Size + Duration + Direction	OLDBSC
Alshammari [42]	2011	Private	ML		√		√	√	√	Payload + Head	Size + Duration	C4.5 + GP + AdaBoost
Pradhan [57]	2018	Public	ML		√	√	√	√	√	Payload + Head	Size + Duration + Number	k -means + RBFN
Hynek [62]	2020	Private	ML		√	√	√	√	√	Payload	Size + Duration	AdaBoost
Agghey [43]	2021	Public	ML		√	√	√	√	√	Payload	Size + Duration + Number + Port	KNN + NB + RF+ Decision Tree

Table 7. Summary of TCP tunnel detection. T represents traditional method. ML represents machine learning method. I represents detecting with coarse content. II represents detecting in a specific location.

Ref.	Year	Datasets	T/ML	Fingerprint	Statistic	Flow	Packet	Spatial	Temporal	I	II	Model
Gianvecchio [20]	2010	Public	T		√		√		√	Payload	Corrected Conditional Entropy	Match
Zhai [26]	2013	Private	T		√		√	√	√	Field	Flag	Markov
Shrestha [30]	2015	Private	ML		√		√	√	√	Field	Kolmorov–Smirnov Test Score (K–S score) + Regularity Score + Entropy + Corrected Conditional Entropy (CCE)	SVM
Zseby [19]	2016	Private	T	√			√	√	√	Head + Field	Time-Related Properties	Match
Fu [31]	2018	Private	ML		√		√	√	√	Payload + Field	Kernel Density Estimation + Variation Coefficient + Fragility Entropy + Autocorrelation Coefficient	SVM

3.5.2. Machine Learning

Machine learning detection methods typically use SVMs with excellent pattern classification performances, but most of them are used to detect whether tunnel traffic is present, and little research is available on tunnel traffic protocol and application behavior. To study covert tunnels and distinguish between normal TCP traffic and tunnel traffic, Shrestha [30] used the IP header's identification and the TCP header's serial number field.

In addition to the header fields of the IP and TCP headers, machine learning is also an objective method for analyzing the regularity and inner correlation. The regularity or correlation between continuous packets will be changed if the information in the TCP header is hidden. Using kernel density estimation, variation coefficient, fractal entropy and autocorrelation coefficient, Fu [31] further transformed these features into an eigenvector matrix.

3.6. ICMP

Intro

The principle of ICMP $\|A \circ B\|_S$ is to encapsulate IP traffic in the data field of ICMP request packet and send it to the ICMP server, which unpacks and forwards IP traffic. In order to establish an ICMP tunnel, packets are encapsulated in an ICMP reply packet and sent back to the client. ICMP tunnels can be built using icmptunnel, ptunnel or icmpsh.

3.6.1. Traditional Detection

Fingerprint

Among the traditional detection methods (in Table 8), most papers focus on detecting whether ICMP tunnel traffic is present, without paying much attention to fine-grained classification. A simple string scan of "passwd", "root", "tmp", "ls", and "dir" was used to complete Singh's [70] preliminary detection in the unencrypted ICMP tunnel. While this method has low overhead and high speed, there are a few disadvantages to its use, including the need to maintain a database of suspicious strings regularly, as well as a high false negative rate. As characteristics of ICMP tunnel traffic matching, Govil [22] defined 89 field types (such as AS MPLS label, IPv6 address, and AS number related to data) and found that hiding data in ICMP tunnels is also common practice by using byte of payload.

Statistic Feature

ICMP tunnel traffic can be identified by marking abnormally large and often persistent or burst packets to proxy nodes. According to Barbhuiya [23], ICMP tunnel traffic can be detected by detecting traffic congestion in gateways according to bandwidth utilization, processing all unreachable messages from hosts and networks in the network, establishing a protocol for unreachable messages, etc. The paper presents only theory, not experimental data. To identify normal traffic or ICMP tunnel traffic, Sayadi [24] first compared the number of packet bytes, then checked the ICMP message rate, ICMP sequence number, and fast random pattern matching in the feature library or MD5 hash verification.

3.6.2. Machine Learning

Machine learning detection methods include detecting whether tunnel traffic is the majority, while few people pay attention to fine-grained classified traffic. From the original packets, Sohn [21] extracted 13-dimension features from the ICMP payload, 15-dimension features from the 2-dimension features from the ICMP payload, and 4 bytes from the ICMP header. The SVM was able to detect ICMP hidden tunnels with almost 99% accuracy. A model proposed by Cho [47] combining RF with ICMP checksum status, identifier, serial number, and data has a higher accuracy (over 99.9%) than the SVM and Naïve Bayes models.

Table 8. Summary of ICMP tunnel detection. T represents traditional method. ML represents machine learning method. I represents detecting with coarse content. II represents detecting in a specific location.

Ref.	Year	Datasets	T/ML	Fingerprint	Statistic	Flow	Packet	Spatial	Temporal	I	II	Model
Sohn [21]	2003	Private	ML		√		√	√		Payload + Head	Size + Flag	SVM
Singh [70]	2003	Public	T	√			√	√		Payload	String	Match
Govil [22]	2007	Private	T	√			√	√	√	Field	MPLS + AS	Match
Barbhuiya [23]	2012	Private	T		√	√	√		√	Payload	Duration	Match
Sayadi [24]	2017	Private	T	√	√		√	√	√	Payload	Size + Rate + SEQ	Match + MD5
Cho [47]	2019	Private	ML		√		√	√		Payload + Field	SEQ + Flag	RF

3.7. IPSec

Intro

IPsec security architecture includes three basic protocols: AH (Authentication Header) protocol provides information source verification and integrity assurance for IP packets, ESP (Encapsulated Security Payload) protocol provides encryption assurance, and ISAKMP protocol provides shared security information when both parties communicate. IPSec tunnel mode is mostly IPSec VPN, which is a kind of encrypted $|A \circ B|$. Due to the limited number of existing studies, this section only presents the methods of machine learning detection in this mode.

Machine Learning

An important part of reading ciphertext is extracting sensitive features that are different from plaintext. With EFM (Estimated Feature Method), Okada [32] selected 29 strong correlation features with thresholds greater than 0.7 and then compared the accuracy of SVM, Naïve Bayes, and C4.5. The best EFM using SVM was found to identify IPSec tunnel traffic with 97.2% accuracy.

IPSec tunnel traffic can also be used to identify protocols and applications at a fine-grained level. Okada [32] approximated each encrypted tunnel traffic feature through Gaussian distribution by using a Naïve Bayes classifier. When mixed with HTTP, FTP, SMTP and SSH passenger protocols, this method improves IPSec tunnel traffic protocol identification accuracy by 28.5%. In order to reduce computation, Kumano [33] first used C4.5 to classify the encryption types of tunnel traffic. The author then used a small number of data packets to represent the flow characteristics and combined SVM to identify the tunnel traffic applications. It is possible to achieve 92.5% accuracy when using private datasets. KNN was chosen by Zhao [46], a decision followed by binary classification of tunnel traffic noise and then multi-classification of tunnel traffic. In private datasets, TMT-RF achieves the best classification performance of 93% by dividing overlapping traffic into multiple segments and making predictions. It does not take time to find the dividing point but divides the traffic into many segments and makes predictions.

4. Evaluation

In this section, we first use AHP to quantitatively evaluate the methods and reviews of detecting tunnel traffic. This method aims to take a complex target of a decision-making problem as a system, decomposes the target into multiple targets or criteria, and calculates the correlation through qualitative indicators as a systematic method for multi-scheme optimization decision making. In the process of AHP, there are generally four steps which are shown in Figure 4.

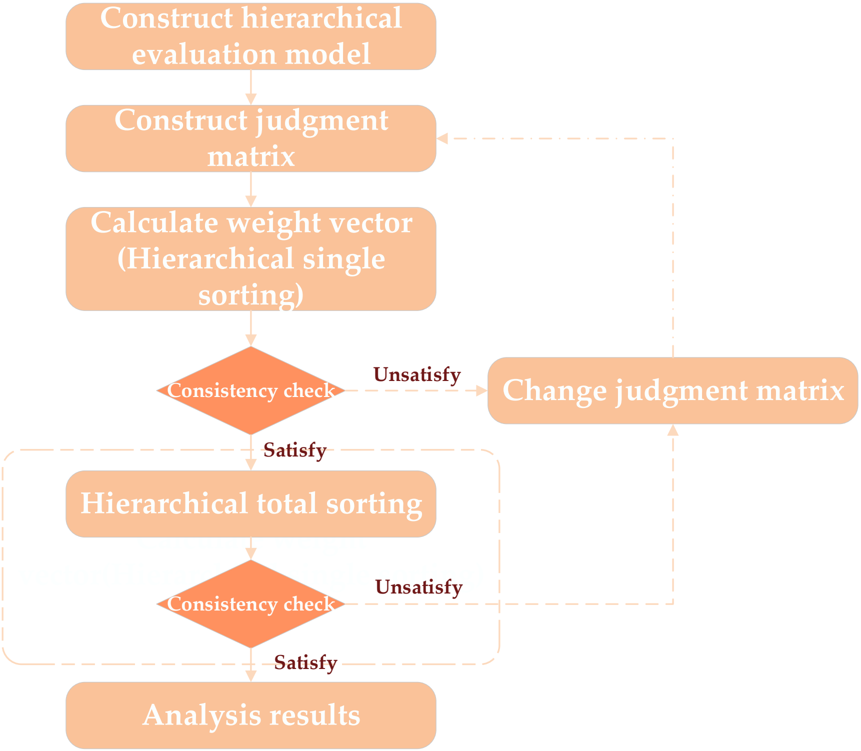

Construct hierarchical evaluation model
Construct judgment matrix
Calculate weight vector (Hierarchical single sorting)
Consistency check
Unsatisfy
Change judgment matrix
Satisfy
Hierarchical total sorting
Consistency check
Unsatisfy
Satisfy
Analysis results

Figure 4. AHP analysis and calculation process.

i. Construct hierarchical evaluation model:

✓ **Target layer**: optimal paper selection/rank.

✓ **Criteria layer**: innovation, granularity of distinguishing applications and behaviors, completion (including accuracy rate, precision rate, recall rate, F1, etc.), computational complexity (time complexity and space complexity) and reproducibility (in Section 4.1); protocol categories, granularity of distinguishing applications and behaviors, diversity of detection methods, computational complexity of methods (time complexity and space complexity) and compatibility (in Section 4.2).

✓ **Scheme layer**: Paper 1, Paper 2, Paper 3, . . . , Paper i, . . . , Paper n.

ii. Construct judgment matrix

The construction of a judgment matrix is performed to compare each element with each other and determine the weight of each criterion layer compared to the target layer. In short, we do this to judge the indicators of the criterion layer in pairs, and usually we use the 1–9 scalar method to designate them (as shown in Table 9).

Table 9. Constructing the scale table of judgment matrix.

Scalar	Explanation
1	Two elements are of equal importance compared with each other.
3	The former is slightly more important than the latter.
5	The former is more important than the latter.
7	The former is extremely more important than the latter.
9	The former is completely more important than the latter.
2, 4, 6, 8, 10	The intermediate value of adjacent judgments.
Inverse of 1~9	The importance of the exchange order of the corresponding two elements.

For n elements, we can obtain pairwise comparison judgment matrix: $A = \left(a_{ij}\right)_{n\times n}$, which satisfies: $a_{ij} \geq 0, a_{ij} = \frac{1}{a_{ji}}, a_{ii} = 1$. ($a_{ij}$ means that compared with j, i's importance degree).

iii. Hierarchical single sorting and consistency check

Then, the weights are calculated, the vectors of each row of A are geometrically averaged, and the results are normalized to obtain the weights of each evaluation index and feature vector W. We calculate a weight vector and maximum characteristic root λ_{max}, where n is the order of the judgment matrix:

$$W = (w_1, w_2, \ldots, w_i, \ldots, w_n)^T \tag{1}$$

$$w_i = \frac{1}{n}\sum_{j=1}^{n} \frac{a_{ij}}{\sum_{k=1}^{n} a_{kj}} \tag{2}$$

$$\lambda_{max} = \frac{1}{n}\sum_{i=1}^{n} \frac{(AW)_i}{w_i} = \frac{1}{n}\sum_{i=1}^{n} \frac{\sum_{j=1}^{n} a_{ij}w_j}{w_i} \tag{3}$$

Finally, we calculate the consistency indicator CI and consistency ratio CR. (The average random consistency indicator RI is obtained by arithmetic average after repeated calculation of the characteristic root of random judgment matrix for more than 500 times. This can be obtained by looking up public information, and so this paper will not repeat it here.):

$$CI = \frac{\lambda_{max} - n}{n - 1} \tag{4}$$

$$CR = \frac{CI}{RI} \tag{5}$$

When it is less than 0.1, it is generally considered that the consistency of the judgment matrix is acceptable. The meaning of consistency test is used to determine whether there are logical problems in the constructed judgment matrix, for example, if the judgment matrix is constructed with c_1, c_2 and c_3, if it is judged that c_1 is equivalent to c_2 as 3 (c_1 is slightly more important than c_2) and that c_1 is equivalent to c_3 as $1/3$ (c_3 is slightly more important than c_1). When judging that c_2 is equivalent to c_3, according to the above logic, c_3 is supposed to be more important than c_2. If we mistakenly fill in c_2 as equivalent to c_3 as 3 when constructing the judgment matrix (c_2 is slightly more important than c_3), then there will be a logical error.

iv. Hierarchical total sorting and consistency check

In the previous section, we obtained the weight vector of the second layer (criterion layer) in relation to the first layer (target layer). Next, we need to obtain the weight vector of the third layer (scheme layer) to each element of the second layer (criterion layer).

Because there are n schemes in our scheme layer (Paper 1, Paper 2, Paper 3, . . . , Paper i, . . . , Paper n), innovation, granularity of distinguishing applications and behaviors, completion (including accuracy rate, precision rate, recall rate, F1, etc.), computational complexity (time complexity and space complexity) and reproducibility (in Section 4.1) are compared in pairs. For example, the first n-order matrix is formed by comparing the completion of Paper 1 with the completion of Paper 2 and Paper 3, and so we should construct five n-order matrices (B_1, B_2, B_3, B_4, B_5) subjectively according to the content of the papers. Therefore, after we calculate the weight vector of each matrix and consistency check, we also should calculate the weight and consistency check of the total ranking of levels. That is to say, we must calculate the weight vector of the scheme layer to the target layer.

Here, the weight of Paper 1 to the total target, that is to say, the weight of completion, reproducibility, computational complexity, innovation and distinguishing applications (in Section 4.1) of Paper 1, Paper 2 and Paper i is multiplied by the weight vector of the fifth-order matrix A to obtain the weight at the target level. Then, the ranking of papers' contribution degree is obtained.

4.1. Contribution of Detections

According to the background information, expert experience and engineering practice knowledge of tunnel traffic detection, we have selected the most suitable criteria. Our evaluation of tunnel detection papers (in Section 3) is based on their innovation, granularity of distinguishing applications and behaviors, completion (including accuracy rate, precision rate, recall rate, F1, etc.), computational complexity (time complexity and space complexity) and reproducibility. Additionally, we have given them a subjective judgment matrix A_1. In Table 10, the weight ranking of these five indicators can be calculated and their results can be obtained by the above four steps according to Figure 4. Our results are shown in Figure 5a–f. In Figure 5, we rank the papers describing tunnel traffic detection papers from the point of view of protocol (HTTP, HTTPS, DNS, SSH, TCP, ICMP and IPSec) according to AHP method, and the contribution decreases from top to bottom. This provides a more concise and convenient way for more scholars exploring in the field of tunnel detection and saves a lot of time and energy.

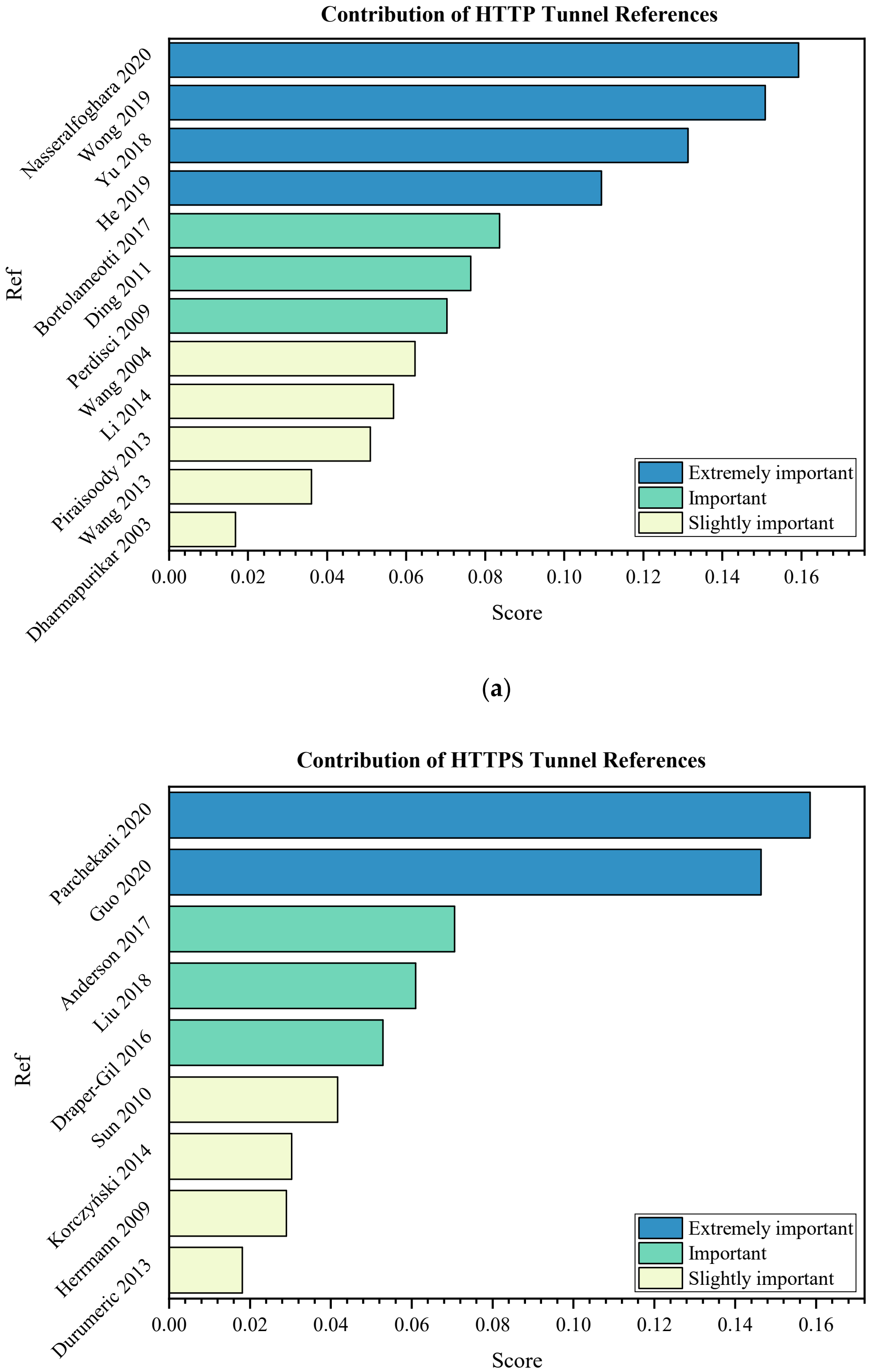
$$A_1 = \begin{pmatrix} 1 & 2 & 1/6 & 1/4 & 1/5 \\ 1/2 & 1 & 1/7 & 1/5 & 1/6 \\ 6 & 7 & 1 & 2 & 3 \\ 4 & 5 & 1/3 & 1 & 1/2 \\ 5 & 6 & 1/2 & 2 & 1 \end{pmatrix} \tag{6}$$

Table 10. Evaluation indicators of tunnel detection.

	Completion	Reproducibility	Complexity	Innovation	Distinguish Application
Rank	1	2	3	4	5
Weight	0.4292	0.2782	0.1841	0.0648	0.0437

4.2. Contribution of Reviews

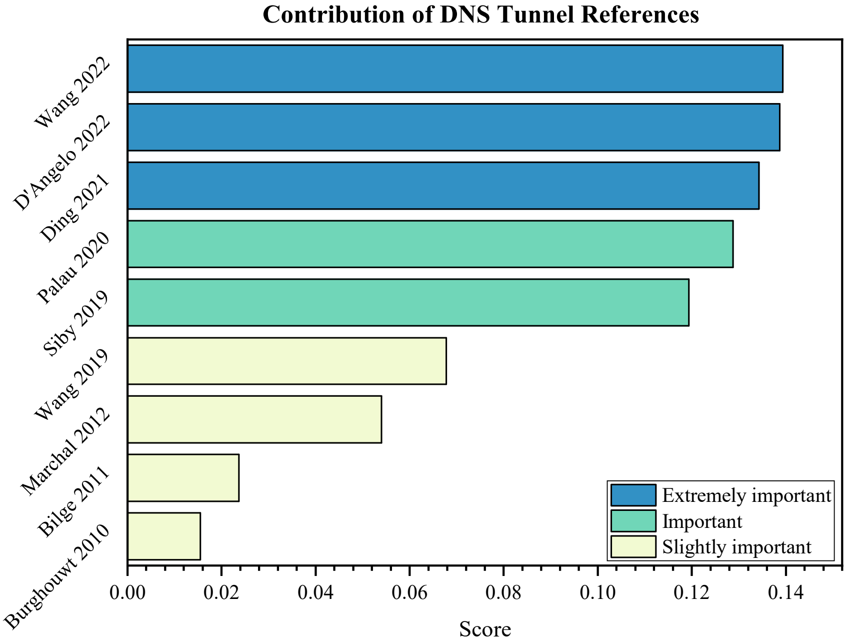
We summarize some related reviews on tunnel detection in Table 11. Zander [45] summarized the tunnel protocols between the application layer and the network layer, such as HTTP, DNS, SSH and ICMP, etc., and also summarized the idea of detecting $\|A \circ B\|_T$ traffic by using time stamp, packet arrival interval and $\|A \circ B\|_S$ with payload. Dakhane [71] introduced TCP tunnel in detail to $\|A \circ B\|_S$ and identified tunnel traffic from the message field. However, the protocol and detection methods described in this review are too monotonous and not suitable for most of the current situations. TCP tunnel traffic detection methods in $\|A \circ B\|_S$ and $\|A \circ B\|_T$ were summarized by Goher [72], who briefly classified the applications in tunnel traffic. By using statistics and machine learning, Wendzel [73] identified tunnel traffic using 109 technologies that hide protocol information through tunnels, simplified them into 11 patterns, and used statistics to identify tunnel traffic. Although they gave a variety of patterns, they failed to give a formal representation or give a specific detection method for each or fixed patterns. Carrara [74] regarded the tunnel from an attacker's perspective. By using bandwidth and entropy as metrics, the attacker can identify tunnel traffic by determining the probability of passing through the tunnel. Although this review has a novel angle, it failed to give an actual detection method, and as time goes by, the attack patterns of attackers may be updated iteratively. Yassine [75] summarized the $\|A \circ B\|_S$ detection methods of embedded data in DNS requests and responses based on machine learning. Besides that, Wang [76] introduced a plethora of literature, including rule-based and model-based methods. However, they only identified whether the tunnel traffic or not in DNS tunnel and failed to give fine-grained classification methods. At present, this kind of detection is not enough for all kinds of behaviors mixed in tunnel traffic. According to Elsadig [77], tunnel traffic identification is affected by three problems: the rapid advancement of network technology, switching techniques, and micro-protocols. A comprehensive analysis of traffic's metadata features and RF performance was performed by Mazel [78], who summarized the methods for detecting and classifying tunnel passenger protocols and tunnel applications. Based on the analysis of concealment, robustness, and transmission efficiency, Tian [79] proposed three new network tunnels among streaming media, blockchain, and IPv6, described possible tunnel forms and provided a new idea for tunnel detection. This is a new summary of tunnel transmission environments, and so we should focus on a more novel detection environment to obtain better results.



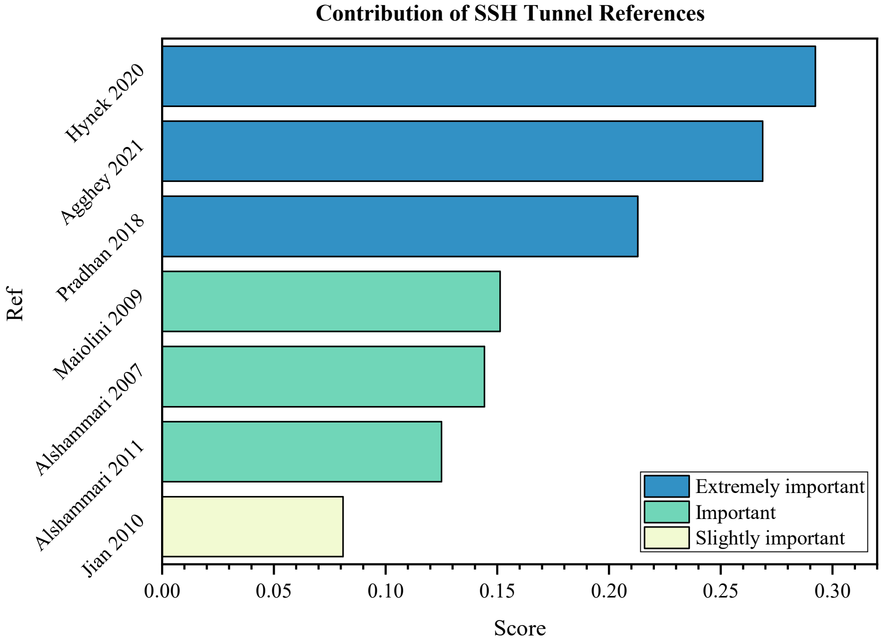
(**a**)

(**b**)

Figure 5. *Cont.*

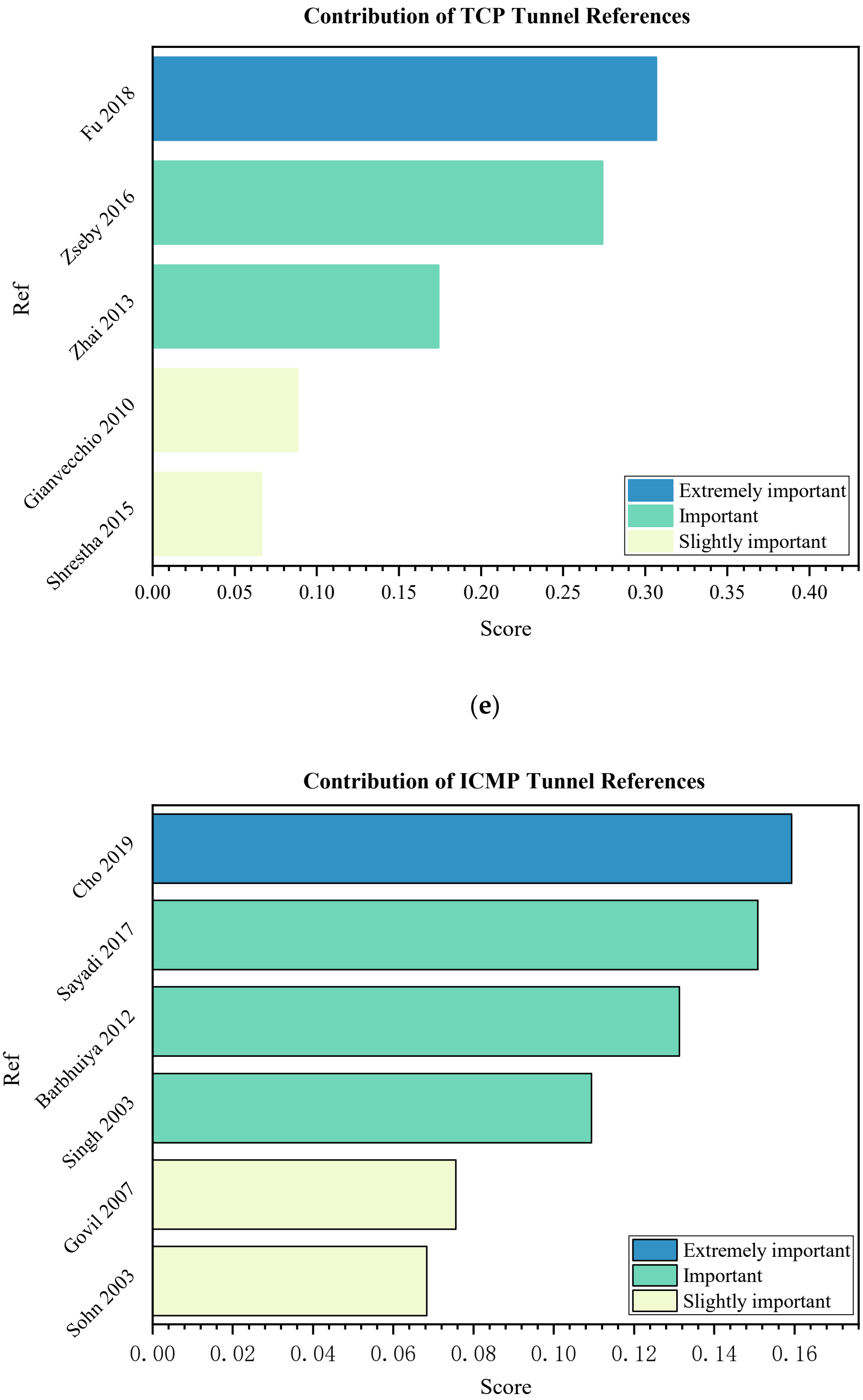


(**c**)



(**d**)

Figure 5. *Cont.*



(**f**)

Figure 5. Rank of contribution of 6 protocols in tunnel detection. (**a**) Contribution of HTTP tunnel conferences, (**b**) Contribution of HTTPS tunnel conferences, (**c**) Contribution of DNS Tunnel references, (**d**) Contribution of SSH tunnel conferences, (**e**) Contribution of CTP tunnel conferences and (**f**) Contribution of ICMP tunnel conferences.

Table 11. Summary of reviews on tunnel detection.

		Application Layer				Transport Layer		Network Layer		
Ref.	**Year**	**SSH**	**HTTP**	**HTTPS**	**DNS**	**TCP**	**SSL/TLS**	**ICMP**	**IPsec**	**Classify Application**
Zander [45]	2007	√	√		√			√		
Dakhane [71]	2012					√				
Goher [72]	2012	√	√			√	√			
Wendzel [73]	2015		√		√	√				
Carrara [74]	2016					√		√		
Yassine [75]	2018				√					
Elsadig [77]	2018	√				√				
Tian [79]	2020		√	√		√	√		√	
Wang [76]	2021				√					
Mazel [78]	2022	√					√		√	√

According to the background information, expert experience and engineering practice knowledge of tunnel traffic detection, we have also selected the most suitable criteria, including five indicators: protocol categories, granularity of distinguishing applications and behaviors, diversity of detection methods, computational complexity of methods (time complexity and space complexity) and compatibility. Additionally, we have given them a subjective judgment matrix, A_2. In Table 12, the weight ranking of these five indicators is presented and their results can be obtained by performing the above four steps according to Figure 4.

$$A_2 = \begin{pmatrix} 1 & 4 & 2 & 5 & 7 \\ 1/4 & 1 & 1/3 & 3 & 4 \\ 1/2 & 3 & 1 & 4 & 5 \\ 1/5 & 1/3 & 1/4 & 1 & 2 \\ 1/7 & 1/4 & 1/5 & 1/2 & 1 \end{pmatrix} \tag{7}$$

We evaluated the 10 reviews above and our paper (11 reviews in total) according to AHP according to the method in Figure 4 and the results are shown in Figure 6. In Figure 6, we rank the contribution degree decreases from top to bottom according to the contribution degree of tunnel traffic detection reviews. Combined with the comprehensive analysis above, we provide a more concise and convenient way for more scholars exploring in the field of tunnel detection. Additionally, it can be found that this paper, as a review, has certain advantages in the comprehensive score of the five indicators we have given and has good reference value.

Table 12. Evaluation indicators of tunnel detection reviews.

	Protocol Category	**Detection Diversity**	**Distinguish Application**	**Complexity**	**Compatibility**
Rank	1	2	3	4	5
Weight	0.4422	0.2824	0.1514	0.076	0.048

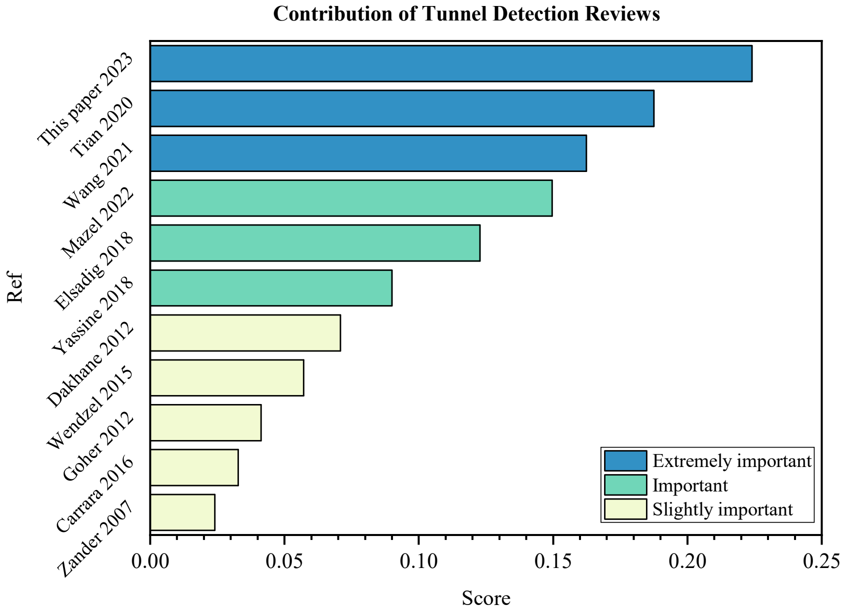


Figure 6. Rank of contribution of 7 protocols in tunnel detection reviews.

5. Conclusions

5.1. Datasets

Choosing abundant labeled and category balance datasets for machine learning training remains an urgent issue. There are three aspects of datasets that need to be considered.

The first is the impact of datasets from different network environments on traffic features. Even within the same network, changing the location of geographic capture will change the statistical features of the data flow, and traffic congestion will change the time features significantly. It is important to ensure the reproducibility and authenticity of datasets.

Furthermore, the unbalanced classification of training datasets is considered. Some tunnel traffic applications will generate an enormous amount of traffic, while others will generate only a very small amount. A huge imbalance in datasets will cause the training models to fail. So, datasets should be trained by selectively extracting sensitive features and suppressing unimportant ones, which reduced the class imbalance to some extent and improved classification accuracy.

Finally, there are few public datasets (raw traffic) in detecting tunnel traffic, and so the amount of labeled data is smaller. The situation of deficient data is detrimental to supervised learning. To put it another way, even though we can collect tunnel traffic from various software, all kinds of noises, mimic traffic or nothing to do with being detected behavior caused by them in order to avoid censorship, also bring trouble to classification.

5.2. Feature Engineering

Feature engineering is an important step in machine learning that deserves consideration in two ways. In general, stream-level data are not very accurately marked (packets of the same quintuple are combined into one stream). In tunnel traffic, the applications share the same quintuple, and so it is difficult to separate the streams by beginning and ending times. Trojan malware is one of the typical applications for tunnels. As such, the quintuple feature is invalid because ports are frequently changed, and so we should use quadruple or triple depending on the actual situation. Additionally, the sample includes not only tunnel traffic but also the flows generated by normal communication, these latter flows being called noise traffic.

Conversely, if features from each flow are extracted separately, correlation features between flows called host-level features may be overlooked (focusing primarily on communications between hosts, such as all traffic with hosts or all traffic with a particular IP and port of hosts). Thus, not only must the number of packages, the average packet length and IAT be aggregated, but also the number and expiration of flow-signed packages. Additionally, since malicious IP semi-connections and non-connections are distributed differently from normal IP, it is necessary to count the number of flows with Alert and the connection status of different flows.

5.3. Trend of Tunnel Detection

In this paper, we compare the number of papers published over the past decade between traditional and machine learning detection, based on the three challenges from coarse to fine in Figure 7. We have found that among non-encrypted protocols, most people study whether there is tunnel traffic or not whereas, in encrypted protocols, most people are inclined to study passenger protocols, applications, and other fine-grained identification. The flaws brought by plaintext transmission to the tunnel are enormous, and the tunnel detection methods based on rules and features have been able to achieve good results. Tunnel detections using encryption protocols (including custom encryption protocols) often focus on hidden passenger protocols and services.

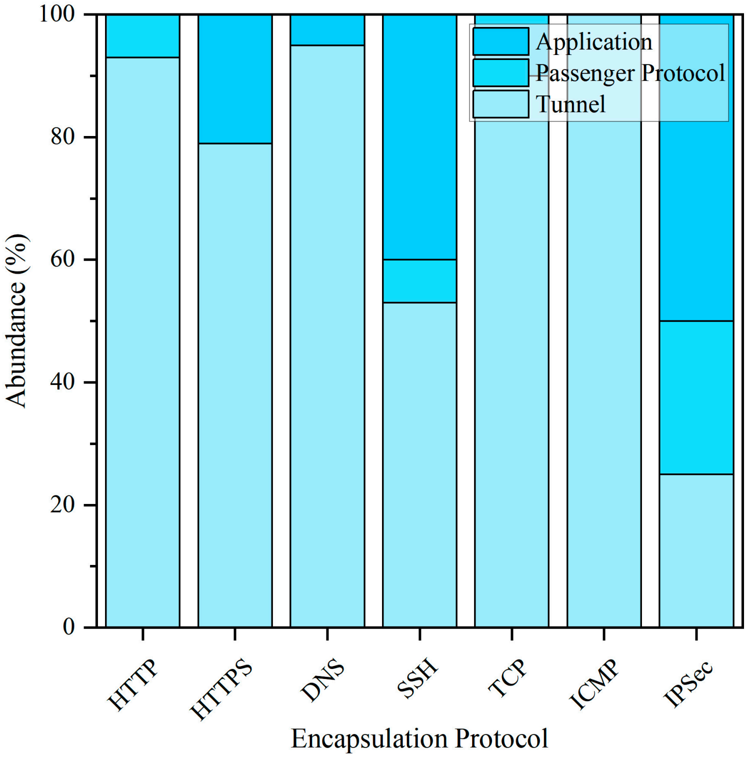


Figure 7. Abundance of three challenges of 7 protocols.

Meanwhile, Figure 8 compares the papers that have solved these challenges over the past decade. Even though there continue to be problems, we have found that the amount of research on how to deal with these challenges is not adequate and that fine-grained identification of encrypted traffic in tunnels should be a major focus of future research. Additionally, we have also found DNS tunnel research has prevailed more than others in recent years.

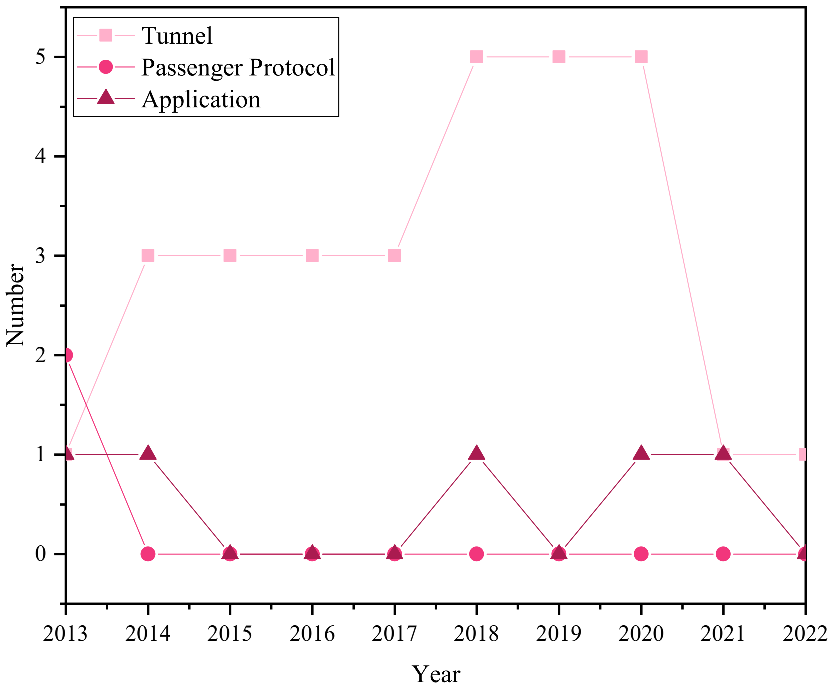


Figure 8. Number of three challenges solved by tunnel detection.

6. Discussion

6.1. Open Questions

i. Principle of classification. Our classification primarily depends on the discriminative features between normal and tunnel traffic. Using fingerprint detection, plaintext tunnel traffic may be found to contain specific strings. From the point of view of traffic statistics, even encrypted tunnel traffic has different temporal and spatial characteristics than normal traffic. As a result, tunnel traffic detection and classification are based on determining how to best fit the tunnel traffic characteristics.

ii. Classification of unknown tunnel protocol and passenger protocol. There are not enough self-adapting studies of detecting unknown protocols, and some classification results continue to be inaccurate. A significant number of passenger protocols in the tunnel are very similar, making classification difficult.

iii. Granularity of classification. There is a need for fine-grained tunnel traffic classification, and this is currently lacking. Identifying the type of protocol to which tunnel traffic belongs is the first step but identifying applications and user behavior is much more important.

iv. Efficiency of classification. The classification efficiency of most methods cannot be met by real-time tunnel traffic classification, and so they can only be trained and tested offline.

6.2. Future Direction

We present the future direction of research after taking into account the public questions that are mentioned above.

i. Identifying more fine-grained behaviors of applications in tunnel is extremely important. Tunnel traffic is frequently mixed with a variety of applications and noise, and so when more and more new applications/services emerge, the question of how to classify them accurately will be extremely vital.

ii. The tunnel application scenario calls for distinguishing malicious tunnel traffic from legitimate tunnel traffic, which can generally be divided into three types. In type one, normal traffic is sent by the agent, and the sending function is limited to one protocol for passengers; whereas type two contains suspicious traffic that encapsulates some (any) types of passenger protocols without causing a threat; in type three, malicious traffic is transmitted by external network attackers to achieve multiple-stage behaviors within the internal network.

iii. The universality of machine learning models needs to be explored further. Machine learning provides a new idea for detecting and identifying tunnel traffic, but in machine learning, selection of datasets determines how well the model will train, and feature extraction determines whether the model will be overfitted, all of which determines how well it will generalize. Semi-supervised and unsupervised learning methods may provide new ideas for the classification of unlabeled datasets.

iv. Balance and exploit the capability of traditional methods and machine learning methods. Although traditional methods are gradually being omitted by more researchers, machine learning methods still fail to be implemented at scale without incurring significant collateral damage from false positives in real time and traditional methods can deal with it quickly. Therefore, for some specific situations, traditional methods do achieve better results.

v. There is a need to identify which device and cluster generate tunnel traffic. As internet devices are continuously updated, different types of tunnel traffic will be generated by the same device. By generating device fingerprints based on traffic characteristics, it is difficult to identify IoT devices. In the cloud market, detecting tunnel traffic and locating the cluster from which it originates has become more challenging due to the popularity of cloud services.

Author Contributions: Conceptualization: Z.S. and H.S.; methodology, Z.S. and H.S.; validation, F.K. and Y.H.; investigation, F.K. and Y.H.; writing—original draft preparation, Z.S., H.S., Y.H. and G.H.; writing—review and editing, Y.H. and G.H.; project administration, Z.S. All authors have read and agreed to the published version of the manuscript.

Funding: This paper is supported by the National Key R&D Program of China, grant number 2019QY1305.

Acknowledgments: We would like to thank the anonymous referees for their helpful comments and suggestions.

Conflicts of Interest: The authors declare that they have no known competing financial interest or personal relationship that could have appeared to influence the work reported in this paper.

References

1. Available online: https://www.cert.org.cn/publish/main/upload/File/CNCERTreport202112.pdf (accessed on 1 October 2022).
2. Zang, X.; Gong, J.; Mo, S.-H.; Jakalan, A.; Ding, D.-L. Identifying Fast-Flux Botnet With AGD Names at the Upper DNS Hierarchy. *IEEE Access* **2018**, *6*, 69713–69727. [CrossRef]
3. Available online: https://www.secrss.com/articles/40646 (accessed on 1 October 2022).
4. Do, V.T.; Engelstad, P.E.; Feng, B.; Do, T.V. Detection of DNS Tunneling in Mobile Networks Using Machine Learning. In Proceedings of the International Conference on Information Science and Applications, Macau, China, 20–23 March 2017.
5. Safari Khatouni, A.; Seddigh, N.; Nandy, B.; Zincir-Heywood, N. Machine Learning Based Classification Accuracy of Encrypted Service Channels: Analysis of Various Factors. *J. Netw. Syst. Manag.* **2020**, *29*, 8. [CrossRef]
6. Wee, B.v.; Banister, D. How to Write a Literature Review Paper? *Transport. Rev.* **2016**, *36*, 278–288. [CrossRef]
7. Simmons, G.J. The Prisoners' Problem and the Subliminal Channel. In *Advances in Cryptology: Proceedings of Crypto 83*; Chaum, D., Ed.; Springer: Boston, MA, USA, 1984; pp. 51–67.
8. Karagiannis, T.; Broido, A.; Faloutsos, M.; Claffy, K.C. Transport layer identification of P2P traffic. In Proceedings of the ACM/SIGCOMM Internet Measurement Conference, Taormina Sicily, Italy, 25–27 October 2004.
9. Karagiannis, T.; Broido, A.; Brownlee, N. Is P2P dying or just hiding? In Proceedings of the IEEE Global Telecommunications Conference, GLOBECOM'04, Dallas, TX, USA, 29 November–3 December 2004; Volume 3, pp. 1532–1538.
10. Cotton, M.; Eggert, L.; Touch, J.D.; Westerlund, M.; Cheshire, S. Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry. Available online: https://www.rfc-editor.org/rfc/pdfrfc/rfc6335.txt.pdf (accessed on 1 October 2022).
11. Dainotti, A.; Pescapé, A.; Claffy, K.C. Issues and future directions in traffic classification. *IEEE Netw.* **2012**, *26*, 35–40. [CrossRef]
12. Dharmapurikar, S.; Krishnamurthy, P.; Sproull, T.S.; Lockwood, J.W. Deep packet inspection using parallel bloom filters. In Proceedings of the 11th Symposium on High Performance Interconnects, Stanford, CA, USA, 20–22 August 2004; Volume 24, pp. 52–61.

13. Bortolameotti, R.; Ede, T.v.; Caselli, M.; Everts, M.H.; Hartel, P.H.; Hofstede, R.; Jonker, W.; Peter, A. DECANTeR: DEteCtion of Anomalous outbouNd HTTP TRaffic by Passive Application Fingerprinting. In Proceedings of the 33rd Annual Computer Security Applications Conference, Orlando, FL, USA, 4–8 December 2017.
14. Nasseralfoghara, M.; Hamidi, H.-R. Entropy-based analyzing anomaly WEB traffic. *J. High. Speed Netw.* **2020**, *26*, 255–266. [CrossRef]
15. Durumeric, Z.; Kasten, J.; Bailey, M.; Halderman, J.A. Analysis of the HTTPS certificate ecosystem. In Proceedings of the 2013 conference on Internet measurement conference, New York, NY, USA, 23–25 October 2013.
16. Wang, F.; Huang, L.; Chen, Z.; Miao, H.; Yang, W. A Novel Web Tunnel Detection Method Based on Protocol Behaviors. In Proceedings of the SecureComm, Sydney, NSW, Australia, 25–28 September 2013.
17. Sun, G.; Xue, Y.; Dong, Y.; Wang, D.; Li, C. An Novel Hybrid Method for Effectively Classifying Encrypted Traffic. In Proceedings of the 2010 IEEE Global Telecommunications Conference GLOBECOM 2010, Miami, FL, USA, 6–10 December 2010; pp. 1–5.
18. Karasaridis, A.; Meier-Hellstern, K.S.; Hoeflin, D.A. NIS04-2: Detection of DNS Anomalies using Flow Data Analysis. In Proceedings of the 1st IEEE Workshop on Automotive Networking and Applications (AutoNet 2006), San Francisco, CA, USA, 1 December 2006; pp. 1–6.
19. Zseby, T.; Iglesias, F.; Bernhardt, V.; Frkat, D.; Annessi, R. A Network Steganography Lab on Detecting TCP/IP Covert Channels. *IEEE Trans. Educ.* **2016**, *59*, 224–232. [CrossRef]
20. Gianvecchio, S.; Wang, H. An Entropy-Based Approach to Detecting Covert Timing Channels. *IEEE Trans. Dependable Secur. Comput.* **2011**, *8*, 785–797. [CrossRef]
21. Shon, T.; Moon, J.; Lee, S.; Lee, D.H.; Lim, J. Covert Channel Detection in the ICMP Payload Using Support Vector Machine. In Proceedings of the International Symposium on Computer and Information Sciences, Antalya, Turkey, 3–5 November 2003.
22. Govil, J.; Jivika, G. Criminology of BotNets and their detection and defense methods. In Proceedings of the 2007 IEEE International Conference on Electro./Information Technology, Chicago, IL, USA, 17–20 May 2007; pp. 215–220.
23. Barbhuiya, F.A.; Roopa, S.N.; Ratti, R.; Biswas, S.; Nandi, S. An Active Detection Mechanism for Detecting ICMP Based Attacks. In Proceedings of the 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, Liverpool, UK, 25–27 June 2012; pp. 51–58.
24. Sayadi, S.; Abbes, T.; Bouhoula, A. Detection of Covert Channels Over ICMP Protocol. In Proceedings of the 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA), Hammamet, Tunisia, 30 October–3 November 2017; pp. 1247–1252.
25. Korczyński, M.; Duda, A. Markov chain fingerprinting to classify encrypted traffic. In Proceedings of the IEEE INFOCOM 2014—IEEE Conference on Computer Communications, Toronto, ON, Canada, 27 April–2 May 2014; pp. 781–789.
26. Zhai, J.; Liu, G.-j.; Dai, Y. Detection of TCP covert channel based on Markov model. *Telecommun. Syst.* **2013**, *54*, 333–343. [CrossRef]
27. Piraisoody, G.; Huang, C.; Nandy, B.; Seddigh, N. Classification of applications in HTTP tunnels. In Proceedings of the 2013 IEEE 2nd International Conference on Cloud Networking (CloudNet), San Francisco, CA, USA, 11–13 November 2013; pp. 67–74.
28. Perdisci, R.; Ariu, D.; Fogla, P.; Giacinto, G.; Lee, W. McPAD: A multiple classifier system for accurate payload-based anomaly detection. *Comput. Netw.* **2009**, *53*, 864–881. [CrossRef]
29. Anderson, B.; McGrew, D.A. Machine Learning for Encrypted Malware Traffic Classification: Accounting for Noisy Labels and Non-Stationarity. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 13–17 August 2017.
30. Shrestha, P.L.; Hempel, M.; Rezaei, F.; Sharif, H.R. A Support Vector Machine-Based Framework for Detection of Covert Timing Channels. *IEEE Trans. Dependable Secur. Comput.* **2016**, *13*, 274–283. [CrossRef]
31. Fu, G.; Li, Q.; Chen, Z.; Zeng, G.; Gu, J. Network Storage Covert Channel Detection Based on Data Joint Analysis. In Proceedings of the ICCCS, Haikou, China, 8–10 June 2018.
32. Okada, Y.; Ata, S.; Nakamura, N.; Nakahira, Y.; Oka, I. Application identification from encrypted traffic based on characteristic changes by encryption. In Proceedings of the 2011 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR), Seattle, WA, USA, 21–25 March 2011; pp. 1–6.
33. Kumano, Y.; Ata, S.; Nakamura, N.; Nakahira, Y.; Oka, I. Towards real-time processing for application identification of encrypted traffic. In Proceedings of the 2014 International Conference on Computing, Networking and Communications (ICNC), Honolulu, HI, USA, 3–6 February 2014; pp. 136–140.
34. Guo, L.; Wu, Q.; Liu, S.; Duan, M.; Li, H.; Sun, J. Deep learning-based real-time VPN encrypted traffic identification methods. *J. Real-Time Image Process.* **2019**, *17*, 103–114. [CrossRef]
35. Ding, S.; Zhang, D.; Ge, J.; Yuan, X.-F.; Du, X. Encrypt DNS Traffic: Automated Feature Learning Method for Detecting DNS Tunnels. In Proceedings of the 2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), New York, NY, USA, 30 September–3 October 2021; pp. 352–359.
36. Wang, K.; Stolfo, S. Anomalous Payload-Based Network Intrusion Detection. In Proceedings of the International Symposium on Recent Advances in Intrusion Detection, Sophia Antipolis, France, 15–17 September 2004.
37. Siby, S.D.; Juárez, M.; Díaz, C.; Vallina-Rodriguez, N.; Troncoso, C. Encrypted DNS -> Privacy? A Traffic Analysis Perspective. *arXiv* **2019**, arXiv:1906.09682.

38. Liu, C.; Cao, Z.; Xiong, G.; Gou, G.; Yiu, S.-M.; He, L. MaMPF: Encrypted Traffic Classification Based on Multi-Attribute Markov Probability Fingerprints. In Proceedings of the 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), Banff, AB, Canada, 4–6 June 2018; pp. 1–10.
39. Ding, Y.; Cai, W.-d. A method for HTTP-tunnel detection based on statistical features of traffic. In Proceedings of the 2011 IEEE 3rd International Conference on Communication Software and Networks, Xi'an, China, 27–29 May 2011; pp. 247–250.
40. Draper-Gil, G.; Habibi Lashkari, A.; Mamun, M.S.I.; Ghorbani, A.A. Characterization of Encrypted and VPN Traffic using Time-related Features. In Proceedings of the International Conference on Information Systems Security and Privacy, Rome, Italy, 19–21 February 2016.
41. Bilge, L.; Kirda, E.; Krügel, C.; Balduzzi, M. EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis. In Proceedings of the Network and Distributed System Security Symposium, San Diego, CA, USA, 6–9 February 2011.
42. Alshammari, R.; Zincir-Heywood, A.N. Can encrypted traffic be identified without port numbers, IP addresses and payload inspection? *Comput. Netw.* **2011**, *55*, 1326–1350. [CrossRef]
43. Agghey, A.Z.; Mwinuka, L.J.; Pandhare, S.M.; Dida, M.A.; Ndibwile, J.D. Detection of Username Enumeration Attack on SSH Protocol: Machine Learning Approach. *Symmetry* **2021**, *13*, 2192. [CrossRef]
44. Zhang, J.; Qian, Z.; Shou, G.; Hu, Y. Traffic identification method based on on-line density based spatial clustering algorithm. In Proceedings of the 2010 2nd IEEE InternationalConference on Network Infrastructure and Digital Content, Washington, DC, USA, 20–22 August 2010; pp. 270–274.
45. Zander, S.; Armitage, G.J.; Branch, P. A survey of covert channels and countermeasures in computer network protocols. *IEEE Commun. Surv. Tutor.* **2007**, *9*, 44–57. [CrossRef]
46. Zhao, P.; Gou, G.; Liu, C.; Guan, Y.; Cui, M.; Xiong, G. TMT-RF: Tunnel Mixed Traffic Classification Based on Random Forest. In Proceedings of the Security and Privacy in Communication Networks, Virtual, 6–9 September 2021.
47. Cho, D.; Thuong, D.T.H.; Dung, N.K. A Method of Detecting Storage Based Network Steganography Using Machine Learning. *Procedia Comput. Sci.* **2019**, *154*, 543–548. [CrossRef]
48. Wang, B.; Xiong, G.; Fu, P.; Gou, G.; Qin, Y.; Li, Z. A Two-Stage Method for Fine-Grained DNS Covert Tunnel Behavior Detection. In Proceedings of the International Conference on Science of Cyber Security, Matsue, Japan, 10–12 August 2022; pp. 201–216.
49. Burghouwt, P.; Spruit, M.E.M.; Sips, H.J. Detection of botnet collusion by degree distribution of domains. In Proceedings of the 2010 International Conference for Internet Technology and Secured Transactions, London, UK, 8–11 November 2010; pp. 1–8.
50. Wong, M.Z. *Deep Learning Models for Malicious Web Content Detection: An Enterprise Study*; University of Toronto: Toronto, ON, Canada, 2019.
51. Parchekani, A.; Naghadeh, S.N.; Shah-Mansouri, V. Classification of Traffic Using Neural Networks by Rejecting: A Novel Approach in Classifying VPN Traffic. *arXiv* **2020**, arXiv:2001.03665.
52. Herrmann, D.; Wendolsky, R.; Federrath, H. Website fingerprinting: Attacking popular privacy enhancing technologies with the multinomial naïve-bayes classifier. In Proceedings of the Cloud Computing Security Workshop, New York, NY, USA, 13 November 2009.
53. Yu, Y.; Liu, G.; Yan, H.; Li, H.; Guan, H. Attention-Based Bi-LSTM Model for Anomalous HTTP Traffic Detection. In Proceedings of the 2018 15th International Conference on Service Systems and Service Management (ICSSSM), Hangzhou, China, 21–22 July 2018; pp. 1–6.
54. He, Y.; Zhu, Y.; Lin, W. HTTP Tunnel Trojan Detection Model Based on Deep Learning. *J. Phys. Conf. Ser.* **2019**, *1187*, 042055. [CrossRef]
55. Marchal, S.; François, J.; Wagner, C.; State, R.; Dulaunoy, A.; Engel, T.; Festor, O. DNSSM: A large scale passive DNS security monitoring framework. In Proceedings of the 2012 IEEE Network Operations and Management Symposium, Budapest, Hungary, 25–29 April 2012; pp. 988–993.
56. Maiolini, G.; Baiocchi, A.; Iacovazzi, A.; Rizzi, A. Real Time Identification of SSH Encrypted Application Flows by Using Cluster Analysis Techniques. In Proceedings of the Networking 2009: 8th International IFIP-TC 6 Networking Conference, Aachen, Germany, 11–15 May 2009.
57. Pradhan, A.; Behera, S.; Dash, R. Hybrid RBFN Based Encrypted SSH Traffic Classification. In Proceedings of the 2018 5th International Conference on Signal Processing and Integrated Networks (SPIN), Amity University, Noida, India, 22–23 February 2018; pp. 264–269.
58. Palau, F.; Catania, C.A.; Guerra, J.; García, S.; Rigaki, M. DNS Tunneling: A Deep Learning based Lexicographical Detection Approach. *arXiv* **2020**, arXiv:2006.06122.
59. D'Angelo, G.; Castiglione, A.; Palmieri, F. DNS tunnels detection via DNS-images. *Inf. Process. Manag.* **2022**, *59*, 102930. [CrossRef]
60. Wang, Z.; Dong, H.; Chi, Y.; Zhang, J.; Yang, T.; Liu, Q. DGA and DNS Covert Channel Detection System based on Machine Learning. In Proceedings of the 3rd International Conference on Computer Science and Application Engineering, New York, NY, USA, 22 October 2019.
61. Alshammari, R.; Zincir-Heywood, A.N. A flow based approach for SSH traffic detection. In Proceedings of the 2007 IEEE International Conference on Systems, Man and Cybernetics, Montréal, QC, Canada, 7–10 October 2007; pp. 296–301.
62. Hynek, K.; Benes, T.; Čejka, T.; Kubátová, H. Refined Detection of SSH Brute-Force Attackers Using Machine Learning. *ICT Syst. Secur. Priv. Prot.* **2020**, *580*, 49–63.

63. Lippmann, R.; Haines, J.W.; Fried, D.J.; Korba, J.; Das, K. The 1999 DARPA off-line intrusion detection evaluation. *Comput. Netw.* **2000**, *34*, 579–595. [CrossRef]
64. Li, S.; Yun, X.; Zhang, Y. Anomaly-based model for detecting HTTP-tunnel traffic using network behavior analysis. *High Technol. Lett.* **2014**, *1*, 63–69.
65. Available online: https://www.unb.ca/cic/datasets/vpn.html (accessed on 1 October 2022).
66. Hoffman, P.E.; McManus, P. DNS Queries over HTTPS (DoH). Available online: https://www.rfc-editor.org/rfc/pdfrfc/rfc8484.txt.pdf (accessed on 1 October 2022).
67. Kingma, D.P.; Welling, M. Auto-Encoding Variational Bayes. *arXiv* **2013**, arXiv:1312.6114.
68. MontazeriShatoori, M.; Davidson, L.; Kaur, G.; Habibi Lashkari, A. Detection of DoH Tunnels using Time-series Classification of Encrypted Traffic. In Proceedings of the 2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Calgary, Alta, 7 August 2020; pp. 63–70.
69. Liu, C.; He, L.; Xiong, G.; Cao, Z.; Li, Z. FS-Net: A Flow Sequence Network For Encrypted Traffic Classification. In Proceedings of the IEEE INFOCOM 2019—IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; pp. 1171–1179.
70. Singh, A.; Nordström, O.; Lu, C.; Santos, A.D. Malicious ICMP Tunneling: Defense against the Vulnerability. In Proceedings of the Australasian Conference on Information Security and Privacy, Wollongong, Australia, 9–11 July 2003.
71. Dakhane, D.D.M.; Patil, S.M.; Patil, M.V. Detection and elimination of covert communication in Transport and Internet layer—A Survey. In Proceedings of the IJCA Proceedings on International Conference on Recent Trends in Information Technology and Computer Science (ICRTITCS-2011), Chennai, India, 3–5 June 2011; pp. 28–30.
72. Gober, S.Z.; Javed, B.; Saqib, N.A. Covert channel detection: A survey based analysis. In Proceedings of the High Capacity Optical Networks and Emerging/Enabling Technologies, Istanbul, Turkey, 12–14 December 2012; pp. 57–65.
73. Wendzel, S.; Zander, S.; Fechner, B.; Herdin, C. Pattern-Based Survey and Categorization of Network Covert Channel Techniques. *ACM Comput. Surv. (CSUR)* **2014**, *47*, 1–26. [CrossRef]
74. Carrara, B.; Adams, C.M. A Survey and Taxonomy Aimed at the Detection and Measurement of Covert Channels. In Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security, New York, NY, USA, 20–22 June 2016.
75. Yassine, S.; Khalife, J.; Chamoun, M.; Ghor, H.E. A Survey of DNS Tunnelling Detection Techniques Using Machine Learning. In Proceedings of the 1st International Conference on Big Data and Cyber-Security Intelligence, BDCSIntell, Hadath, Lebanon, 13–15 December 2018.
76. Wang, Y.; Zhou, A.; Liao, S.; Zheng, R.; Hu, R.; Zhang, L. A comprehensive survey on DNS tunnel detection. *Comput. Netw.* **2021**, *197*, 108322. [CrossRef]
77. Elsadig, M.A.; Fadlalla, Y.A. Network Protocol Covert Channels: Countermeasures Techniques. In Proceedings of the 2017 9th IEEE-GCC Conference and Exhibition (GCCCE), Manama, Bahrain, 8–11 May 2017; pp. 1–9.
78. Mazel, J.; Saudrais, M.; Hervieu, A. ML-based tunnel detection and tunneled application classification. *arXiv* **2022**, arXiv:2201.10371.
79. Tian, J.; Xiong, G.; Li, Z.; Gou, G. A Survey of Key Technologies for Constructing Network Covert Channel. *Secur. Commun. Netw.* **2020**, *2020*, 8892896. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.