Zhao, Jingjing; Jing, Xuyang; Yan, Zheng; Pedrycz, Witold

Network traffic classification for data fusion: A survey

Published in:
Information Fusion

DOI:
10.1016/j.inffus.2021.02.009

Published: 01/08/2021

Document Version
Publisher's PDF, also known as Version of record

Information Fusion 72 (2021) 22–47

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Network traffic classification for data fusion: A survey

Jingjing Zhao [a], Xuyang Jing [a], Zheng Yan [a,b,*], Witold Pedrycz [a,c]

[a] State Key Laboratory on Integrated Services Networks, School of Cyber Engineering, Xidian University, China
[b] Department of Communications and Networking, Aalto University, Finland
[c] Department of Electrical and Computer Engineering, University of Alberta, Canada

ARTICLE INFO

Keywords:
Traffic classification
Machine learning
Security management
Data fusion

ABSTRACT

Traffic classification groups similar or related traffic data, which is one main stream technique of data fusion in the field of network management and security. With the rapid growth of network users and the emergence of new networking services, network traffic classification has attracted increasing attention. Many new traffic classification techniques have been developed and widely applied. However, the existing literature lacks a thorough survey to summarize, compare and analyze the recent advances of network traffic classification in order to deliver a holistic perspective. This paper carefully reviews existing network traffic classification methods from a new and comprehensive perspective by classifying them into five categories based on representative classification features, i.e., statistics-based classification, correlation-based classification, behavior-based classification, payload-based classification, and port-based classification. A series of criteria are proposed for the purpose of evaluating the performance of existing traffic classification methods. For each specified category, we analyze and discuss the details, advantages and disadvantages of its existing methods, and also present the traffic features commonly used. Summaries of investigation are offered for providing a holistic and specialized view on the state-of-art. For convenience, we also cover a discussion on the mostly used datasets and the traffic features adopted for traffic classification in the review. At the end, we identify a list of open issues and future directions in this research field.

1. Introduction

Data fusion is a process that deals with association, correlation, and combination of data from single and multiple sources to achieve refined, significant or valuable information [1]. It has been widely used in various fields [2,3], such as intrusion detection, target identification, image processing, and resource allocation. Traffic classification groups similar and related traffic data into a same category, which is one main stream technique for data fusion in the field of network management and security. Accurate and real-time network traffic classification is essential for network management, security monitoring, and intrusion detection. For network operators and administrators, correct identification of traffic categories generated by different applications and protocols helps them in providing high Quality of Service (QoS) for network users. Traffic classification can be used to identify user behaviors and predict traffic categories, which greatly assists network management. In addition, distinguishing abnormal network traffic is also extremely important for intrusion detection and network security measurement. For example, as shown in Fig. 1, in core detector module of Intrusion Detection System (IDS), it is necessary to identify malicious attack traffic by classifying obtained traffic data and matching with malicious attack patterns in the knowledge base. Nowadays, continuous development of network technology and rapid expansion of network scale inspire network traffic classification methods to be more accurate and faster.

Traditional widely used traffic classification schemes can be divided into port-based classification [4] and payload-based classification [5]. However, both two types of the methods have many limitations. For example, the payload-based methods cannot classify encrypted traffic. However, most of the traffic data is transmitted in an encrypted form at present, which makes the payload-based methods infeasible. The port-based methods can only classify data based on publicly known ports. As more and more traffic chooses to hide ports or uses dynamic ports, the port-based methods become invalid. Many new and effective classification methods have emerged in order to solve the limitations existing in the above two types of methods. Such as behavior-based, statistics-based, and correlation-based methods.

So far, there exist some related surveys about traffic classification. Different reviews have different focuses and summarize different methods, as shown in Table 1. Nguyen and Armitage [6] conducted a comprehensive review on the classification methods using machine

* Correspondence to: Xidian University, 119 POX, No. 2 South Taibai Road, Xi'an 710071, China.
E-mail address: zyan@xidian.edu.cn (Z. Yan).

https://doi.org/10.1016/j.inffus.2021.02.009
Received 31 October 2020; Received in revised form 1 January 2021; Accepted 7 February 2021
Available online 12 February 2021

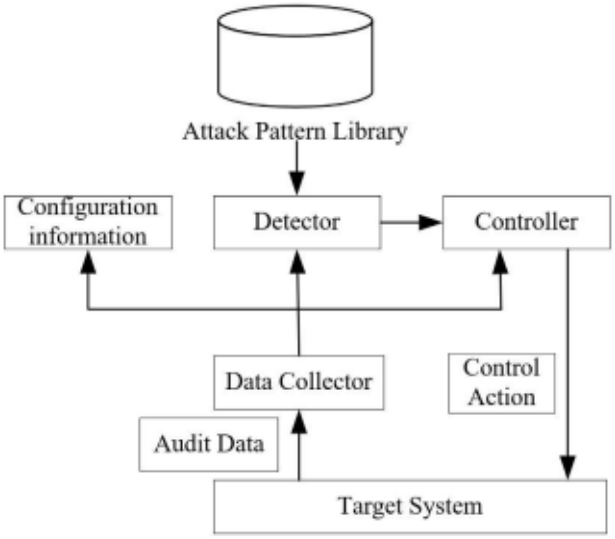

Fig. 1. General intrusion detection model.

learning in 2008. This survey mainly summarizes statistics-based classification methods. Callado et al. [7] provided a detailed introduction to the techniques used in traffic classification and reviewed the existing classification methods from port-based, packet-based, and flow-based perspectives. Cao et al. [8] summarized the classification methods of encrypted traffic from four aspects, namely port-based, payload-based, statistical-based, and behavior-based. They also indicated the challenges in current encrypted traffic classification. Finsterbusch et al. [9] focused on the payload-based classification methods by analyzing and comparing the performance, classification accuracy and technical requirements of deployable Deep Packet Inspection (DPI) classification modules. Gomes et al. [10] mainly reviewed the classification methods for P2P traffic. They focused on the port-based and payload-based classification methods mainly. In addition, they also paid attention to the following aspects: the features of packets or flows used in classification, the classification methods using active crawlers, and the classification using combined approaches. Valenti et al. [11] summarized and compared existing classification methods by concerning port-based methods, deep packet inspection, stochastic packet inspection, statistical and behavioral perspectives. In addition, they discussed two of the mostly used machine learning methods, i.e. Support Vector Machine (SVM) and decision tree, in traffic classification. Shafiq et al. [12] compared classification performance of four machine learning algorithms (C4.5, SVM, Bayes Net and Naïve Bayes) for five traffic classes: WWW, DNS, FTP, P2P and Telnet applications.

Recently, Pacheco et al. [13] provided discussions on traffic classification from the perspective of machine learning. They introduced traffic classification from five stages: data collection, feature extraction, feature reduction and selection, algorithm selection, and model deployment. the technology used in each stage is summarized and analyzed. The paper also described the different data categories used in the classification methods, such as statistical characteristics, payloads, and host behaviors. Comparing with this survey, the differences and advantages of our paper are provided as follows. First, Ref. [13] only discusses the classification methods based on machine learning, while the scope of our review is wider. Second, our review is based on a holistic series of evaluation criteria, which is missed in [13]. Thus, our review is performed in a uniform way and is deep insight. It is possible for us to intuitively obtain interesting findings from comparing results. Third, we analyze the datasets and features on various data levels, such as flow level and packet level, which impacts classification performance. However, this analysis is missing in [13]. Finally, we identify new open issues and indicate novel research directions, which are different from [13] and other existing surveys.

Although there are several surveys about network traffic classification [6–13], our paper begins with different concerns. We found that the existing reviews discuss too little on behavior-based and correlation-based classification methods. They do not consider user privacy, limitations of feature redundancy, etc. This gives us motivation to complete this survey.

In this paper, we review existing network traffic classification methods from a new and comprehensive perspective by classifying them into five categories, i.e., correlation-based classification, statistics-based classification, behavior-based classification, payload-based classification, and port-based classification. A series of criteria are also proposed for the purpose of evaluating the performance of existing traffic classification methods. For each specified category, we analyze and discuss the details, advantages and disadvantages of its existing methods, and also the traffic features commonly used. Summaries of investigation are offered for providing a holistic and specialized view on the state-of-art. For convenience, we present a discussion on the mostly used datasets and the traffic features adopted for traffic classification in the review. At the end, we identify a list of open issues and future directions in this research field. By comparing with previous surveys, we summarize the main contributions of this paper as follows:

- We thoroughly review current traffic classification methods by classifying them into five categories: statistics-based, correlation-based, behavior-based, payload-based, and port-based. The advantages and disadvantages of each classification method are also discussed and compared.
- We analyze and quantify classification granularity and divide it into four levels, namely application type layer, protocol layer, application layer, and service layer, which is missing in the previous reviews.
- Comprehensive evaluation criteria are presented to assess the performance and quality of the classification methods and are used to compare their pros and cons.
- We summarize traffic features and datasets mostly used in current traffic classification methods. We list the publicly available datasets for the convenience of other researchers.
- We further figure out a number of open issues and propose future research directions to motivate network research.

The rest of this paper is organized as follows. Section 2 introduces the basic of traffic classification by specifying classification process, comparing several classification algorithms and summarizing the features and datasets used in classification. We also classify existing methods into five categories with four classification levels. Section 3 specifies the evaluation criteria for network traffic classification. In Section 4, we review the existing classification methods based on their categories and evaluate them by using the proposed evaluation criteria. We summarize open issues and suggest future research directions in Section 5. Finally, conclusions are drawn in the last section. For convenience, the reader can refer to Table 2 for all abbreviations used throughout the paper.

2. Traffic classification overview

In this section, we provide an overview of traffic classification procedure, introduce five main categories of traffic classification methods and machine learning algorithms used in traffic classification.

2.1. Machine learning

Machine learning is an interdisciplinary subject that can be used in multiple fields, enabling computers to learn automatically by extracting data and improving algorithm performance during learning. The use of machine learning in traffic classification can be traced back to a paper

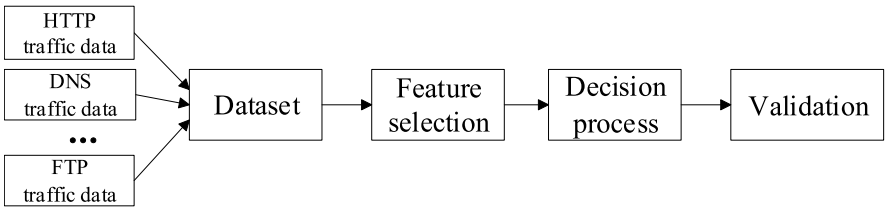

Fig. 2. Process of traffic classification.

Table 1
Comparison of our survey with other existing surveys.

Reviewed methods and contents	[6]	[7]	[8]	[9]	[10]	[11]	[12]	[13]	Our survey
Port-based method	N	Y	N	Y	Y	Y	Y	N	Y
Payload-based method	N	N	N	Y	Y	Y	Y	Y	Y
Behavior-based method	Y	Y	Y	N	Y	Y	N	Y	Y
Statistics-based method	Y	Y	Y	Y	N	Y	Y	Y	Y
Correlation-based method	N	N	N	N	N	N	N	N	Y
Dataset summary	N	N	N	N	N	N	N	Y	Y
Analysis on traffic data levels	N	Y	N	N	Y	Y	N	Y	Y
Discussion on classification granularities	N	N	N	N	N	Y	N	N	Y
Summary of data features	N	N	N	N	N	N	N	Y	Y

Y: discussed; N: not discussed.

Table 2
Abbreviations in this paper.

Abbreviation	Explanation
AFF	Advanced File Format
BNN	Bayesian Neural Network
CBR	Constant Bit Rate
CFS	Correlation-based Feature Selection
CNN	Convolutional Neural Network
DL	Deep Learning
DPI	Deep Packet Inspection
DDoS	Distributed denial of service
DNS	Domian Name System
FCBF	Fast Correlation-Based Filter
FS	Feature Selection
FTP	File Transfer Protocol
GR	Gain Radio
HTTP	HyperText Transfer Protocol
IANA	Internet Assigned Numbers Authority
IDS	Intrusion Detection System
IG	Information Gain
IMAP	Internet Mail Access Protocol
IP	Internet Protocol
K-NN	K-Nearest Neighbors
LSTM	Long Short Term Memory Network
NFI	Netflow Flow Identify
PC	Pearson Correlation
PCA	Principal Component Analysis
POP	Post Office Protocol
P2P	Peer-to-Peer
QoS	Quality of Service
SBE	Sequential Backward Elimination
SFS	Sequential Feature Selection
SMTP	Simple Mail Transfer Protocol
SPI	Stochastic Packet Inspection
SVM	Support Vector Machine
TLS	Transport Layer Security
VoIP	Voice over IP
WWW	World Wide Web

published in 1994 for intrusion detection [14]. Machine learning uses different algorithms to analyze data and learn from it to give decisions or predictions.

Machine learning algorithms are divided into supervised, semi-supervised and unsupervised learning. In supervised learning, an accurate prediction model is obtained by continuously comparing the prediction results with the actual labels of the training data during learning process. New data can be categorized (labeled) by using the prediction model obtained previously. Supervised learning algorithms include statistical classification and regression analysis. Unsupervised learning, also known as clustering algorithms, uses unlabeled data to distinguish different categories from a large dataset. Clustering and dimensionality reduction are the two main applications of unsupervised learning. Since a lot of data in the network is unlabeled, this method is practical in reality. In addition, the semi-supervised learning method that combines supervised and unsupervised learning is also applied for traffic classification. The accuracy of semi-supervised learning is improved compared with unsupervised learning. To a certain extent, manual participation is reduced.

Appropriate traffic classification algorithm plays an crucial role for achieving proper and efficient traffic classification. Various machine learning based methods are widely used in traffic classification. In Table 3, we summarize some popular machine learning algorithms used in existing traffic classification methods and discuss their advantages and disadvantages.

2.2. Classification procedure

A general process of classifying network traffic is shown in Fig. 2. The first step is to collect data from a network environment. In Fig. 2, some typical types of network traffic are provided as input to form a traffic dataset for feature selection. The second step is to extract and select traffic features. This process is important for the traffic classification because it impacts the robustness and efficiency of the classification. The third step is a decision process, which identifies the

Table 3
Review of some machine learning algorithms.

Algorithms	Descriptions	Advantages	Disadvantages
Naïve Bayes	Use probabilistic knowledge and need prior probability (supervised)	High accuracy, less estimated parameters, and insensitivity to irrelevant data	The assumption that the required attributes are mutual independent is difficult to satisfy
K-Nearest Neighbors (K-NN)	Use the distance between features for classification; also used for regression (supervised)	Simple, no feature assumption, suitable for multi-classification problems	Poor performance for unbalanced datasets, high computational cost
K-means	Determine K initial centroids, use distance to iteratively achieve clustering (unsupervised)	Simple implementation; good clustering effect	Sensitive to outlier, performance is affected by parameter k and initial centroids
Decision Tree	Completing match and classification based on feature attributes (supervised)	Small amount of calculation, fast classification	Suitable for high-dimensional data; easy to over-fit, ignoring correlation between data features
Support Vector Machine (SVM)	Find classification planes to achieve binary classification (supervised)	Improved generalization performance, can solve high dimensional and nonlinear problems	High memory cost
Random Forest	Consist of multiple decision trees (supervised)	Not easy to overfit, fast training	Not suitable for low-dimensional and small datasets
Logistic Regression	A generalized linear regression method, commonly used in binary classification (supervised)	Fast training, dynamic adjustment of classification threshold	Easy over-fitting, complex feature processing
Neural Networks	Mimic the behavioral characteristics of biological neural networks and perform distributed parallel information processing	High accuracy, strong distributed storage and learning ability, robustness	Requires a large number of parameters, long training time
AdaBoosting	Integrate multiple weak classifiers into one multi-classifier	High precision, no over-fitting	Sensitive to outlier

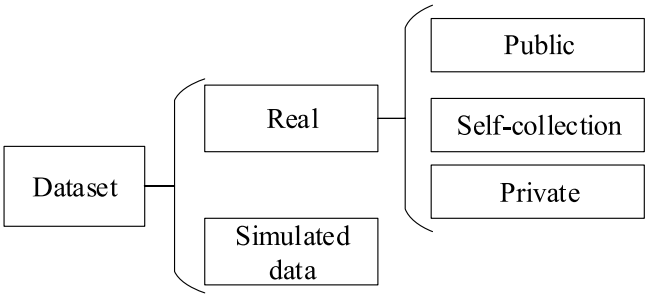

Fig. 3. General classification of datasets.

class of traffic flow through pattern matching or model training. The final step is to verify the results of the classification that aims to obtain the accuracy of the traffic classification. The ground truth of the data (i.e., the true classes of original data) is needed in this step. The detailed discussion of each classification step is presented as follows.

(1) *Datasets and Traffic Features:* In this part, we firstly introduce some datasets and features used in traffic classification. The dataset is very important for the feature selection during the classification process. The datasets obtained from different environments are distinct [31–36], regarding different applications (different service types), different characteristics and different usages. They relate to different user groups, such as education, business, medical, and so on. These datasets can be used to meet various classification needs. Analyzing different datasets can help us understand a traffic classification method in a holistic way. This is because researchers normally choose different datasets to meet their needs. For example, some classification methods focus on classifying video traffic so that they use the traffic generated by various video applications to classify; if a classification method aims to identify different network attack traffic, then they should at least choose the datasets that include various malicious traffic to analyze. Herein, we discuss and compare the public datasets that are widely used in current traffic classification methods, as shown in Table 4. In addition to these publicly available datasets presented here, there are many self-collecting real datasets used for evaluating some classification methods in experimental tests. There are also some studies that simulate a network and perform experiments with simulated network traffic data. In Fig. 3, we classify the datasets used for traffic classification research according to real and simulated data.

Traffic classification is normally based on a number of features. Herein, we summarize the traffic features that are mostly used in the classification. Traffic data can be divided into four levels, packet level, flow level, connection level and host level [37]. We observe that most of the traffic classification methods use packet level and flow level data, and a few classification methods use connection level data. The following is a brief introduction of these four levels of data.

Table 4
Publicly available datasets.

Dataset	Descriptions	Level	Ground truth
CAIDA [15]	Include different datasets for traffic analysis attacks	Packet and Flow	No
UNIBS [16]	Include data about several applications	Packet	No
MAWI/Wide/Keio [17]	Include different datasets for traffic analysis	Packet	Yes
Moore/Cambridge's Nprobe project [18]	Include 11 subdatasets for traffic analysis with 248 features	Flow	Yes
ISCX [19]	For anomaly detection, include normal traffic datasets and attack traffic datasets	Packet	Yes
IP Trace [20]	Include different datasets for traffic analysis	Flow	No
KDD Cup99 [21]	For anomaly detection	Packet	Yes
Digital Corpora [22]	Distributed in RAW Image Format (RAW, EnCase E01, and Advanced File Format (AFF) formats	Packet	No
NetOp [23]	For anomaly detection	Flow	Yes
GTA/UFRJ [24]	For anomaly detection	Packet	Yes
KAIST [25]	About normal traffic behaviors in Peer-to-Peer (P2P) networks	Packet	No
Snu [26]	About normal traffic datasets of Constant Bit Rate (CBR) and Voice over IP (VoIP)	Packet	No
Lbnl [27]	About internal enterprise traffic in an anonymized form	Packet	No
SIGCOMM2008 [28]	A detailed trace of network activity in an anonymized form	Packet	No
NLANR [29]	Provide in-depth information and technical support for high-performance networking	Packet	No
DARPA [30]	For anomaly detection	Packet	Yes

- Packet level data is generated when different hosts communicate with each other through network protocols. The packet level data consists of three parts, namely header information, payload information, and packet activity information that includes such information as Time to Live (TTL), the flags of the packet header such as FIN, SYN, and RST that can be used to describe the activity or communication behavior of a packet. The features of the packet level refer to the information that can be obtained from the packet(s), which can be obtained from a single data packet, such as the size of a data packet and the value of a specific field of a data packet, or obtained from multiple packets that may not be in the same flow. Therefore, we classify packet inter-arrival time and its statistical values, as well as the statistical values of packet size as the features of the packet level.
- Flow level data is a collection of packets that share same attributes. In general, packets with the same five-tuple (source IP, destination IP, source port, destination port, protocol) are integrated into a flow. The common flow level characteristics include the size of the flow (refers to the number of packets contained in the flow), the duration of the flow, the direction of the flow, and others.
- Connection level data describes the communication traffic between two IP addresses. It contains at least two data flows: inflow and outflow data. The number of connections, connection level, and connection duration are some common connection level data information.
- Host-level data are collected from a local host. They contain host activities, host changes, host resource consumption and other host-related information, thus can provide a complementary view on network events. Since most existing traffic classification methods do not use host-level data, we do not discuss host-level data features in this paper. But we think it is essential to introduce the host-level data since they could assist traffic classification.

The fusion of different level features helps improving classification performance and facilitating network management and attack detection [38,39]. [38] and [39] provide two DDoS attack detection schemes using traffic fusion based on a novel reversible sketch. In Table 5, we list the features that are commonly used in traffic classification and describe them from different levels. The number of features used in the classification methods highly impacts their performance such as robustness, classification speed, and other quality properties. So, we concern the number of features used in each method in its performance evaluation. For the sake of brevity, the number in front of the features in the table will be used directly in the latter section to represent the corresponding features.

(2) *Feature Selection:* Feature Selection (FS) is one of the most important steps in traffic classification because the practicability of the selected features could directly affect the performance of a classification method. The number of features and the degree of their redundancy also affect the speed of classification. Feature selection aims to reduce data dimensionality and solve over-fitting problems. Besides, feature extraction also helps understanding the relationships between features and feature values.

Feature selection methods can be divided into three categories, namely filtering, wrapping, and embedding [40].

- The filtering methods are roughly divided into two types. One is univariate filtering. This method does not interact with a classifier. The calculation is simple, but the

Table 5
Traffic features.

Type	Features	Descriptions
Packet level	1. Packet size	The length of a packet in bytes;
	2. Packet header length	The length of packet control information;
	3. The number of packets	The number of packets transmitted during a certain period of time;
	4. Packet inter-arrival time	The packet arrival interval (i.e., the time of packet arrival);
	5. Time stamp	The point-in-time of sending/receiving a packet;
	6. Size of first N bytes	The length of first N bytes in a packet;
	7. Specific string	The specific sequent bytes of packet payload;
	8. Statistical values of packet size	The maximum, minimum, average, standard deviation, etc. of packet length;
	9. Statistical values of packet inter-arrival time	The maximum, minimum, average, standard deviation, etc. of packet inter-arrival time;
	10. Source address	The IP address of a source interface;
	11. Destination address	The IP address of a destination interface;
	12. Source port	The end-point of a source interface;
	13. Destination port	The end-point of a destination interface;
	14. Protocol	The protocol used by an application;
	15. TTL	Time to Live (Max. hop count);
Flow level	16. Volume of bytes	The total number of bytes transmitted;
	17. Flow duration	The interval between the timestamps of the first packet and the last packet;
	18. Flow length	The total number of packets in a flow;
	19. Flow size	The sum of the packets size contained in a flow;
	20. Size of the first N packets	The size of the first few packets in a flow;
	21. Statistical values of the first N packets sizes	The maximum, minimum, average, standard deviation, etc. of the first N packets length;
	22. Ratio of Source and Destination Bytes	The ratio of packets transmitted between a source host and a destination host;
	23.min_iat	The minimum packet inter-arrival of a flow;
	24. mean_iat	The mean packet inter-arrival of a flow;
	25. var_iat	The variance packet inter-arrival of a flow;
	26. mean_data_wire	The mean Ethernet packet bytes of a flow;
	27. mean_data_ctrl	The mean control bytes of a flow;
	28. avg_win_agv_c	The average window size from a source to a destination;
	29. mean_data_wire_s	The mean Ethernet packet bytes from a destination to a source;
	30. mean_data_ip_s	The mean IP packet bytes from a destination to a source;
	31. RDBUB	The ratio of downstream bytes to upstream bytes;
	32. APITD	The average packet inter-arrival time downstream;
	33. IEPSD	The information entropy of packet downstream size;
	34. NDSF	The number of downstream sub-flows;
Connection level	35. Connection duration	The length of the time interval for which the tunnel is active;
	36. Connection inter-arrival time	Time elapsed time between two consecutive requests for establishing a tunnel from the same user and the same server.

correlation between features is ignored. Common univariate filtering methods include Information Gain (IG)/Gain Radio (GR), Chi-Square, etc. The other is multivariate filtering, which captures the correlation between features, but it performs slower than the single variable filtering method. Commonly, the multivariate filtering methods include correlation-based feature selection methods.

- The wrapping methods iterate the selection of feature subsets and a model training process to generate the most appropriate feature set. The advantage of the wrapper methods is the ability to interact with the classifier. However, they suffer from high computing overhead and possible over-fitting problems. Common wrapper methods include Sequential Backward Elimination (SBE), estimation of distribution algorithm [41].
- The embedding methods are a part of machine learning process, which is usually associated with a specific machine learning algorithm. The computational cost of

the embedding method is smaller compared to the wrapper ones, but it relies on the classifier for feature selection. Commonly used methods include random forests, SVM [42], and others.

We summarize the widely used feature selection algorithms in traffic classification, as shown in Table 6. In some other classification methods, multiple different feature extraction algorithms are integrated, and the optimal and most stable features are selected from the feature set obtained by each feature selection method. Dhote et al. [40] provided more specific FS algorithms.

(3) *Decision process:* The decision process is an important phase in classification, which is based on the previously obtained feature set for traffic classification by pattern matching or applying machine learning algorithms. Pattern matching is usually related to a specific field of a packet. The string matching algorithm is applied to compare with a predefined string library to determine the class of traffic. However, this type of matching has a lot of limitations when dealing with complex services due to its limited form of expression. Machine learning algorithms are currently widely used in traffic classification. Introduction to machine learning has been introduced in Section 1 of this section.

(4) *Validation:* The validation process tests the previous classification results with the purpose of obtaining the accuracy of the classification method. In this step, we need to compare the real category of the original data with the experimental results to get classification accuracy. The collection of true categories of original data, i.e., ground truth, is currently a challenge. A commonly used ground truth collection method is manually labeling, based on port collection and collection using DPI tools. This collection method has various shortcomings such as time-consuming, labor-consuming, inaccurate, and so on. To solve these problems, researchers proposed new collection methods using active measurement and heuristics analysis [43]. These methods improve the reliability of ground truth to a certain extent. However, there are still many problems, such as excessive load or the use of simulated traffic cannot accurately reflect real-world network traffic. Refer to [43] for specific reference.

2.3. Traffic classification methods

This paper reviews existing network traffic classification methods from a comprehensive perspective by classifying them into five categories, i.e., statistics-based classification, correlation-based classification, behavior-based classification, payload-based classification, and port-based classification. We classify existing traffic classification methods into five categories based on the representative features used by them. The representative features used in different classification methods are different. The representative features used by the statistics-based classification methods are the statistical values of the traffic at the packet level and/or the flow level, e.g., the maximum, minimum, mean, variance of packet size and flow duration. The correlation of flows is applied as a main feature for classification by the correlation-based classification methods, which can be seen as an expansion of the methods based on statistical features. This type of methods combines the statistical information of traffic with the correlation between flows to construct a constraint relationship during the process of classification in order to improve classification accuracy and efficacy. The behavior-based methods mainly use the communication or activity behavior of hosts to perform classification, e.g., host interaction and connection data. The representative features adopted by the payload-based classification methods mainly include packet contents or the contents of specific fields in a packet, e.g., the first few bytes of a packet. The port-based classification methods rely on traffic ports for classification, e.g., port number. Applying this taxonomy can well sort out most of existing classification methods and provide a comprehensive view on traffic classification from different key perspectives. We summarize the five types of methods and analyze their advantages and disadvantages of each method, as shown in Table 7.

(1) *Statistics-based Classification:* The statistics-based classification methods rely on the statistical features of traffic instead of the packet payload. Common statistical features include the minimum, the maximum, the mean of packet size and the number of packets, and so on. Different traffic classification methods may differ in adopted traffic features in traffic classification. Finding the best feature subsets through different feature extraction and feature selection methods and trying to train and classify the datasets using different machine learning algorithms can help improving the accuracy of the statistics-based classification methods.

(2) *Correlation-based Classification:* The correlation-based classification methods aggregate packets into flows and classify them according to the correlation between the flows in the network. Here, the definition of flow refers to a collection of data packets having the same five-tuple, that is, the same source IP, source port, destination IP, destination port, and protocol. In general, multiple flows are aggregated into Bag-of-Flow (BoF). Classifying BoF is a unified classification of all flows in BoF. This type of methods avoids the problem of feature redundancy faced by the statistics-based classification methods, but still has a high computational overhead in feature matching.

(3) *Behavior-based Classification:* Behavior-based approaches give a new perspective on traffic classification research. It performs traffic classification by checking and counting the behaviors of a host, for example, by checking which IP addresses the host communicates with, what protocol is used, and which ports are communicating in order to identify the applications in the host. Although this type of methods has high classification accuracy, its classification result is not fine-grained enough.

(4) *Payload-based Classification:* The payload-based classification methods are proposed to improve classification accuracy, which check the content of the packet, obtain the signature corresponding to the protocol, and match it with the signatures stored in the database to identify a particular application or protocol. We divide the payload-based classification method into two types according to the methods used for packet inspection, one is Deep Packet Inspection (DPI), and the other is Stochastic Packet Inspection (SPI). DPI is a network technology that detects network traffic and packet contents. For the detailed definition of DPI, refer to [49]. Because of its high classification accuracy, DPI technology is very popular in traffic management, security analysis and attack prevention. Commonly used DPI tools are nDPI [50], OpenDPI [51], L7-filter [52], Tstat [53], NarusInsight [54], etc. SPI was proposed to deal with the problem that DPI cannot classify encrypted data. This method does not directly use the content of the packet but uses the statistical information of the payload to automatically generate the protocol signature to identify different protocols. The classification method is also highly accurate. Although SPI can classify encrypted data, this method still faces the problem of excessive computational cost.

(5) *Port-based Classification:* Port-based traffic classification methods generally identify well-known applications or protocols based on port numbers specified by the Internet Assigned Numbers Authority (IANA) [55], such as port number 20, 21 for FTP traffic, port 25 for SMTP traffic, and port 80 for HTTP traffic. Therefore, using ports to identify applications or protocols is the easiest and most direct method. But today, with the proliferation of unknown applications, and the fact that many applications choose to use dynamic ports instead of using known fixed ports, this approach is no longer accurate and efficient.

Table 6
Widely used feature selection algorithms.

FS algorithms	Descriptions	Advantages	Disadvantages
Principal Component Analysis (PCA) [44]	A set of variables that may be related to each other are transformed into a set of linearly uncorrelated variables through orthogonal transformation. Mainly used for dimensionality reduction.	Simple calculation, no parameter restrictions	Not as interpretative as original samples
Pearson Correlation (PC)	This method measures the linear correlation between variables, and the value range of the result is [−1, 1].	Fast and easy to calculate	Sensitive to linear relationships only
Random Forest	Random forests include two methods of feature selection: mean decrease impurity and mean decrease accuracy.	High accuracy and good robustness	Dependent classifier selection feature
Distance correlation	Overcoming the PC's insensitivity to nonlinear relationships, the resulting value interval is [0,1].	Easy to calculate	The correlation interval is [0,1], and the correlation is not as rich as PC.
Correlation-based Feature Selection (CFS) [45]	Apply correlation metrics to assess the superiority of feature subsets.	Consider the correlation between features	Cannot handle large amounts of data in high dimensions
Fast Correlation-Based Filter (FCBF) [46]	An algorithm based on mutual relationship metrics, using correlation coefficients to analyze the relationship between features, categories and features.	Accurate and efficient	Cannot handle large amounts of data in high dimensions
Information Gain (IG) [47]	The number of bits of information provided in the category prediction is measured by the value of the feature. When the dataset category is extremely uneven, IG is better than Chi-square.	Simple and fast, able to process extremely large datasets	Poor performance when classes and features are unbalanced
Gain Ratio (GR) [47]	Normalize information gain by using split information metrics.	Overcoming the bias of information gain on features with a large number of different values	Ignore feature correlation
Chi-square [48]	Based on the chi-square distribution of statistics, the degree of metrics and category independence is lacking. The larger the chi-square, the smaller the independence and the greater the correlation. When the dataset is evenly distributed, the effect of Chi-Square is slightly better than IG.	Simple and fast, able to process extremely large datasets	Ignore feature correlation

Table 7
Five categories of traffic classification methods.

Classification methods	Descriptions	Advantages	Disadvantages
Statistics-based	Use statistical features such as packet size etc.	Protect user privacy to a certain extent	Too much redundant features
Correlation-based	Uses correlations between flows to estimate traffic clustering	High accuracy	High computational overhead
Behavior-based	Capture social interaction observable from the perspective of a host	Make most encryption methods robust	Classification result is not fine enough
Payload-based	Generally use deep packet inspection to look into packet contents	Accurate classification	Cannot deal with encrypted payload, legitimacy and privacy issues
Port-based	Use statistics on the port number of packets	Fast and simple	Poor performance; infeasible for hidden ports

2.4. Classification granularity

In this subsection, we discuss the granularity of traffic classification results. In general, different classification methods give different classification categories. For example, some classification methods can give specific application names, such as Google, YouTube, Facebook, and so on. There are also some classification methods, as discussed in [56], they classify applications or protocols with the same functions into the same category, that is, each classification category contains different applications and protocols. Taking the Mail category as an example, protocols (such as SMTP, POP, and IMAP) are classified into the Mail category. Inspired by the paper [57], we divide the results of traffic classification into four levels of classification granularity: specific service layer (Level 1), application layer (Level 2), protocol layer (Level 3), and application type layer (Level 4) (see Fig. 4).

- Level 1 is the most finest granular level of classification results. It further categorizes different traffic generated by the same application based on Level 2. According to the different services provided by the traffic, it is divided into different categories, such as downloading, chatting, and so on.
- Level 2 classifies network traffic according to applications and gives specific application names related to the traffic.
- Level 3 classifies traffic at a protocol level. Because different applications are likely to use the same protocol, the granularity of protocol-based classification is slightly coarser than the application-based classification.

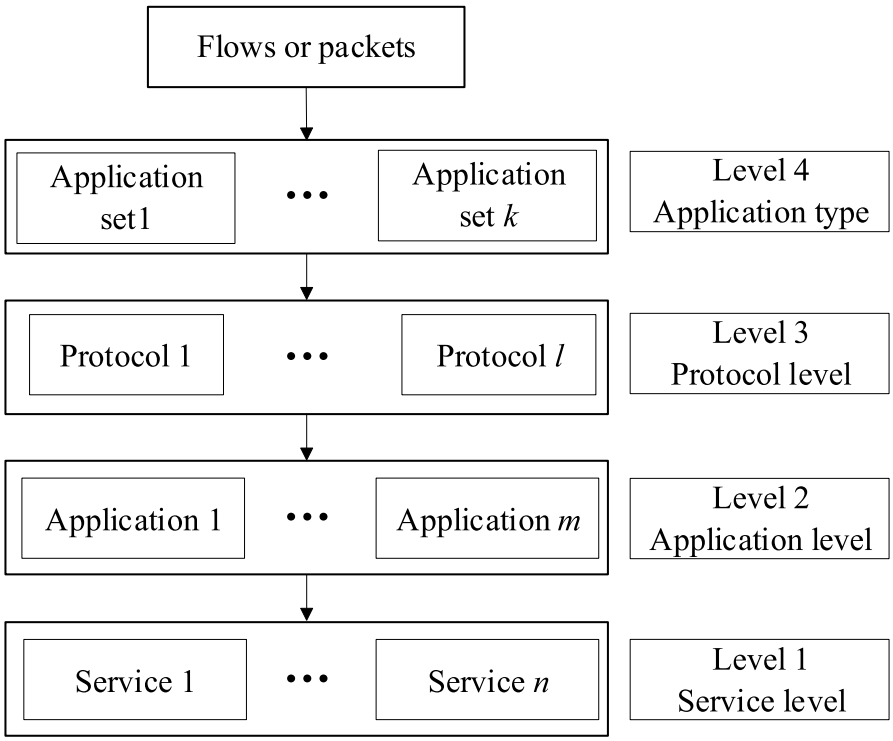

Fig. 4. Four-level classification granularity.

- Level 4 is to classify traffic according to the type of the application, that is, the applications with the same or similar functions are classified into the same category. Therefore, the classification result given by Level 4 is a collection of a series of applications, and it cannot give a specific application name.

3. Classification performance measure and criteria

In this section, we specify a number of criteria that can serve as a measure to evaluate the performance of traffic classification methods. Two types of criteria are concerned: criteria of classification effectiveness and criteria of classification performance.

3.1. Criteria of classification effectiveness

(1) *Information Granularity(G):* We use four-level information granularity as discussed in Section 2.4 to judge the granularity of different classification results. Different classification granularities can provide different traffic information. The finer the granularity of the classification results, the more traffic information is provided, and the readability of the data can also be enhanced. According to distinct requirements, traffic can be classified with distinct granularities. Similarly, different classification granularities can also help network managers providing different services.

(2) *Robustness (R):* The classification method should be able to perform stably in a constantly changing network environment, that is, it can still ensure high classification accuracy in the face of various problems in the network, such as congestion, packet loss, delay, etc. So, the robustness should be an important criterion for classification methods. We evaluate the robustness of a scheme based on whether the features used are universal or not, and whether the scheme can maintain the same classification performance for different network types or traffic types.

(3) *Identification of unknown applications (IUA):* The classification method is required to detect not only known labeled traffic in a training dataset, but also unknown or new applications. If only known applications can be detected, some new types of applications will be classified into known types, which definitely impacts detection accuracy. In the constantly changing and updated network environment, unknown traffic is appearing and increasing, so the ability to classify unknown traffic becomes very important. The fast and accurate classification of unknown traffic can greatly help us identify malicious traffic and provide network security.

(4) *Online classification:* An important criterion for assessing classification methods is online classification in real-time. The network traffic is updated regularly, so the classification methods should be able to classify traffic online in real-time. Efficient classification of network traffic is essential to improve service quality and detect malicious traffic. This requires that the classification methods can identify the correct category of traffic in a short period.

3.2. Criteria of classification performance

There are several metrics for evaluating the performance of the classification methods, as summarized in Table 8. In different traffic classification methods, researchers use different accuracy evaluation metrics to measure classification performance, such as overall accuracy, F-measure, precision, recall and so on. In this paper, we focus on the overall accuracy to evaluate and compare the performance of reviewed classification methods.

(1) *Overall Accuracy:* The Overall Accuracy (OA), as shown in Table 8, refers to the percentage of samples that are correctly classified in all samples.

(2) *Class Accuracy:* Class accuracy (CA) refers to the classification accuracy with regard to an individual class. For example, if a method divides network traffic into different categories such as P2P, HTTP, and SMTP, the classification accuracy of P2P, HTTP,

Table 8
Measures of classification performance.

Measures	Explanation
True Positive (TP)	The number of traffic that are correctly assigned to a specific traffic category.
False Positive (FP)	The number of traffic that are incorrectly assigned to a specific traffic category.
True Negative (TN)	The number of traffic that are not part of a specific traffic category and are classified into other categories.
False Negative (FN)	The number of traffic that belong to a specific traffic category and are classified into other categories.
TPR= TP/(TP+FN)	The proportion of traffic that belongs to an underlying traffic category and are really classified into that category.
FPR = FP/(FP+TN)	The proportion of traffic that is incorrectly assigned to a particular traffic category that it should not belong to.
Recall (R)	TP / (TP+FN) (=TPR) the percentage of objects from a given category that are properly attributed to that category.
Precision (P)	TP/(TP+FP) the ratio of flows correctly attributed to a category over the total flows attributed to that category.
F-measure	$(a^2+1)P*R/a^2(P+R)$
Sensitivity	TPR
Specificity	TNR= TN / (FP + TN) the percentage of negative objects identified correctly from all negative objects.
Accuracy	(TP+TN)/(TP+TN+FP+FN)

and SMTP is calculated separately. This makes it more intuitive to see which class is more sensitive to the classification method. Moreover, classification accuracy based on individual classes is also more helpful in understanding and analyzing the advantages and disadvantages of a classification method.

(3) *Flow Accuracy:* Flow accuracy is the classification accuracy about a single flow, often used in the methods that classify flows, such as correlation-based classification methods.

(4) *Byte Accuracy:* Byte accuracy refers to how many bytes are correctly classified in an entire dataset. In the classification of an imbalanced dataset, byte accuracy is very important. Because most traffic flows are mice flows in the Internet, and the bytes generated by the mice flows account for a relatively small portion, while the bytes generated by a small number of elephant flows account for a relatively high portion of total bytes in the entire dataset [58].

4. Review of existing classification methods

In this section, we review the existing traffic classification methods by classifying them into the five categories as specified above. In our review, we first introduce each work derived from the current literature. We then comment its performance and remark its pros and cons based on the criteria presented above. Finally, we summarize our review on each category in terms of the proposed criteria in a table.

4.1. Statistics-based classification methods

In this subsection, we review and comment main statistics-based methods published from 2012 to present by further grouping them according to machine learning types.

4.1.1. Supervised classification

Dong et al. [59] focused on the identification and classification of Skype traffic. They proposed a Naive Bayes-based Netflow Flow Identify (NFI) mechanism. Their goal is to solve the real-time problem of tracing and the problem when labeled data is not enough. This paper used a Fast Correlation-Based Filter (FCBF) [46] method for feature extraction. In the construction phase of classification modeling, they used a Bayesian update mechanism. It uses new training data to continuously update the classification model to improve the accuracy of the classification. They built their classifier upon the Netflow V5 format and extended Netflow records. In addition, they used Netflow sampling to improve the efficiency of traffic identification, which impact traffic behavior and classification accuracy. In experimental tests, they compared the proposed method with other four methods and discussed the effects of sampling rate on flow identification. This method can be used in online classification in real-time. But it cannot identify unknown traffic and its robustness is not good. The classification level of this method is unknown.

Moore and Zuev [60] used an improved kernel density estimation theory based on Bayesian to improve the overall accuracy of classification. In this method, 248 discriminators (i.e., features) per-flow were used. They also used the FCBF feature extraction algorithm proposed in [46] to process high-dimensional data and wrapper methods to perform feature extraction. In order to minimize the redundancy of feature sets, they used information entropy to calculate symmetry uncertainty to determine the correlation between features, reduce redundancy and select the best features. This method can only classify traffic of known categories, cannot identify unknown traffic. It cannot support the robustness and online classification . Its classification granularity is at Level 4.

Since most methods of using machine learning for traffic classification need to be completed under the assumption that training data and test data satisfy independent identical distribution. Most real data cannot fully satisfy this assumption, and when classifying traffic from different network environments, the previously trained classification model is likely to be unable to perform classification accurately. This is because different network environments have different data characteristics and traffic sizes. In order to solve this problem, Sun et al. [61] proposed a method based on transfer learning for traffic classification. This method can perform classification well without satisfying the above assumptions. About the transfer learning, refer to the survey [62]. Dai et al. used a TrAdaBoost algorithm proposed in [63], a new algorithm based on the traditional AdaBoost algorithm, to perform classification by applying a maximum entropy model (Maxent) as a classifier. Through experiments, they found that when the maximum entropy model is set to 5, the best classification effect can be

achieved. The classification accuracy of the method can reach 98.7% in total. For WWW and MAIL, the classification accuracy exceeds 99.5%. This method is also robust due to the use of transfer learning. It can offer robustness but cannot support online classification and IUA. The granularity is at Level 4.

Fahad et al. [64] presented an integrated FS technique. Three new evaluation metrics for feature picking techniques were proposed as well, namely goodness, stability and similarity. The authors applied six FS technologies (i.e., IG, GR, PCA, Correlation-based Feature Selection (CBF) [65], Chi-square, and Consistency-based Search (CBC) [66]). Using the Moore dataset, the above six FS techniques were evaluated and compared based on three metrics through experiments. Based on experimental results, they found that a single FS technology could not maintain good performance in all datasets. Therefore, a new FS technology called Local Optimization Approach (LOA), which combines the above five FS technologies (excluding PCA) and integrates their respective advantages, was proposed. LOA can select the most reliable feature subset from the feature subsets in different FS technologies. Therein, they proposed a Support concept to select the best feature subset. With three different datasets, LOA has shorter modeling and testing time than each individual FS method, and its performance is more prominent regarding the three metrics. Since the LOA method can achieve classification in a short time, this method can be used for online traffic classification. It can also satisfy robustness. But it cannot support IUA. The classification granularity of this method is unknown.

In [67], Sun et al. proposed an Incremental Support Vector Machines (ISVM) method based on SVM to save memory and CPU during training. This method can update the classifier according to the newly arrived traffic in time. The ISVM discards original training data and saves only the Support Vectors (SVs) generated during the last update. When new training data arrives, the classifier combines the new data with the SVs to re-train and update the SVs. In order to improve the classification accuracy of ISVM, they further improved this method and proposed the ISVM with attenuation factors (AISVM) scheme. The new scheme gave each SV a weight value. In the process of continuous updating, the SV with the weight value less than the threshold is discarded. Experiments showed that the AISVM is 1.2 percent more accurate than the ISVM solution. This method can support online classification but cannot satisfy robustness and IUA. The granularity of its classification is unknown.

Lopez et al. [68] proposed a fast method for data pre-processing, including a feature normalization algorithm and a correlation-based feature extraction algorithm. Because the range of features is different in different datasets, the feature normalization can improve classification accuracy when the feature span in a certain dimension is large. The feature normalization algorithm can complete data filling within O(log N) time complexity. In the feature extraction algorithm, they used the correlation between features in a dataset to weight the features. The weight value w ranges from 0 to N, where N represents the number of features contained in the dataset. The Pearson coefficient was used as a metric of correlation. They compared the method with three traditional algorithms (Relief [69], Sequential Feature Selection (SFS), and PCA), using four machine learning algorithms (Decision Tree (DT), C4.5, Artificial Neural Networks (ANNs), Support Vector Machines (SVMs)) to evaluate the performance of feature extraction algorithms. Experiments showed that different classification accuracy can be achieved under different feature numbers and different classifiers. Their feature extraction algorithm can achieve the best classification performance under the decision tree classifier, which is accurate to 97.2%, better than other feature extraction algorithms. But the performance under other classifiers is not as good as the PCA algorithm. The biggest advantage of this method is its low time consumption, which is beneficial for real-time monitoring and network traffic classification. However, this method cannot satisfy robustness and IUA. Its classification granularity is at Level 4.

Shafiq et al. [70] proposed a hybrid method, which includes two metrics of feature selection, one is Weighted Mutual Information (WMI) and the other is the area under the ROC curve (AUC). These two metrics constitute a new hybrid mechanism named WMI_AUC to pick out effective features in imbalanced traffic. In addition, they also proposed a Robust Feature Extraction (RFS) algorithm that can select robust and stable features from the features obtained in WMI_AUC. They extracted 22 features from a dataset using the NetMate [71] tool and verified the classification performance with 11 classifiers (Bayes Net, Naïve Bayes, SMO, AdaBoost, Bagging, Oner, PART, Hoeffding, C4.5, R/forest, and R/tree). Experiments showed that the overall accuracy of the method can reach more than 95%. The feature selection algorithm used in this method is very robust, so the method can support the robustness. In addition, this method can classify traffic online in real-time. However, it cannot satisfy IUA. Its classification granularity is at Level 2 and 3.

Aceto et al. [72] focused on a Multi-Classification System (MCS) approach to accurately classify mobile applications. The MCS architecture includes data pre-processing, Service Bursts (SB) (decomposition of traffic as in [73,74]), feature set extraction and several modules of different classifiers. In this article, they used seven existing base classifiers: Lib_NB in [75], Her_Pure, Her_TF, Her_Cos in [76], Tay_RF, Tay_SVC in [68], Classification And Regression Tree (CART) in [77]. These classifiers integrate Soft and Hard classifier fusion techniques. The six hard combiners used in this paper are Majority Voting, Weighted Majority Voting, Recall Combiner Naïve Bayes, Behavior-Knowledge Space method, Wernecke's method (MV, WMV, REC, NB, BKS, WER, respectively), and the five soft combiners are Non-trainable combiners (within CC), Fuzzy Integral (within CC), trainable linear combiners (within CC), Decision Templates (within CI), Dempster–Shafer approach (within CI). The fusion rules for different classifiers can be found in [78,79]. A dataset containing 49 different mobile applications was classified by the multi-classification system constructed using different combiners. This method can satisfy robustness but cannot support online classification and IUA. The granularity of classification is at Level 2.

In order to solve the problem that the common SVM algorithm is susceptible to the influence of dataset size and feature dimension, an accurate real-time classification method SPP-SVM [80] was proposed. This method uses unbiased samples to process original data. Also, to reduce the impact on classification performance and speed caused by the scale of data, they re-scaled the data. Experiments verified that the scaling method they used does improve classification accuracy. This method employs PCA to extract data features, which can reduce feature dimensions, reduce feature redundancy, and avoid over-fitting, thus increases the universality of the method. Then, an improved Particle Swarm Optimization (PSO) algorithm was used to optimize the parameters of a kernel function in order to reduce computational load. Finally, a cross-validation method was proposed to train the SPP-SVM classifier. The classifications of two-class and multi-class were verified. The accuracy of the two-class classification is 8.4% higher than the multi-class one, arriving at 98.6%. This method can support online classification because of low computational load, but cannot satisfy the requirement on robustness and IUA. Its classification granularity is at Level 4.

Tong et al. [81] proposed a method that can be used for real-time classification. They selected eight flow-level features that were considered to be the most efficient. These features were combined to form six feature sets. The continuous feature values are converted to discrete ones using the Minimum Description Length (Entropy-MDL) algorithm. Because the literature [82] proved that discretization could help to improve the accuracy of classification. A comparison experiment was performed on each feature set, and the feature set with the best classification accuracy was recorded as an Empirical Optimization Feature Set (EOFS). At the same time, they found in the experiment that extracting features from the first four packets of a flow can achieve the highest accuracy. Therefore, they used EOFS and the first four packets of the flow to train the classifier. In this method, they used

the classifier based on the C4.5 algorithm. In order to enable the proposed method to be used for online classification, they proposed two acceleration algorithms, Optimized Decision Tree (ODT) and Divide and Conquer (DQ). They compared these acceleration algorithms in the Field Programmable Gate Array (FPGA) and multi-core platforms. This method can support online classification, and its overall accuracy reaches 97.92%. But it cannot identify unknown traffic. Thus, this method can support online classification and robustness but cannot satisfy IUA. Its classification granularity is at Level 2.

Dong et al. [83] focused on classifying web browsing and video traffic and proposed a new feature selection algorithm based on the coefficient of variation [84]. The coefficient of variation was used to reflect the degree of dispersion of a dataset. They had verified that this algorithm is lower in time complexity than previous information gain based or Chi-Square methods. They selected a series of statistical features for classification and verified the validity of the selected features by analyzing the two-dimensional distribution of features. The proposed feature selection algorithm was verified by K-NN and SVM classifiers and compared with existing feature selection algorithms. This method achieves an accuracy of 98.17% and a very fine classification granularity. But it cannot support online classification, robustness and IUA. Its granularity level is at Level 1.

Later, in the literature [85], Dong et al. proposed a modified consistency-based feature selection algorithm. More than 40 features were extracted from an original dataset. Finally, four features were selected based on a consistency feature selection process, namely RDBUB (ratio of downstream bytes to upstream bytes), IEPSD (information entropy of packet size downstream), NDSF (the number of downstream sub-flows), APITD (average packet inter-arrival time downstream). A hierarchical K-NN classification scheme was employed to allow each sub-classifier to process a limited subset of video streams, making an applications easier to be distinguished. Their hierarchical K-NN classification scheme consists of two layers. The root K-NN classifier of the first layer divides all flows into symmetric flows and asymmetric flows. The features of RDBUB and IEPSD are used in this step. The second layer consists of a symmetric K-NN classifier and an asymmetric K-NN classifier, which respectively process the symmetric flows and the asymmetric flows generated by the upper layer. The symmetric flows are divided into three categories: QQ, Xunlei and Sopcase. The asymmetric flows are divided into three categories: ASD, AHD, and HTTP-download. Through experiments, they found that after the features used in other different classification schemes were changed to the features proposed in this method, the accuracy of the classification was improved to some extent in those schemes. This method is robust due to the usage of stable features. However, it cannot support online classification and IUA. Its granularity is at Level 1 and 2.

Alshammari et al. [86] proposed a method for classifying VoIP encrypted traffic. Instead of using IP addresses, ports or payload information to generate application signatures, they first proposed to generate signatures based on machine learning methods. There are three supervised machine learning methods applied in this paper, C5.0, AdaBoost and Genetic Programming (GP), which automatically generate signatures for classification. They collected three datasets in their own school and lab, and used PacketShaper (2008) to label traces, i.e., ground truth, for verifying classification accuracy. Because the previous literature [87] pointed out that selecting different subsets of samples has an impact on classification performance, this article evaluated three different sampling methods to select the best training data samples. The first method is a uniform random N sampling method, the second is a stratified sampling method, and the third is a continuous data flow. The obtained training dataset was trained by the three machine learning methods mentioned above to obtain a classification model. Through verification, they found that the C5.0 algorithm has the best classification accuracy and the lowest FPR. This method can satisfy robustness but cannot support online classification and IUA. Its granularity is at Level 3.

Wang et al. [88] proposed a random forest approach to implement the clustering process. This paper used 20 flow features to aggregate traffic into classes at a specific protocol level. Each tree is generated by iteratively splitting the nodes based on m variables randomly selected from input variables, where m is a predefined number of variables used to split the nodes. After the forest is constructed, the proximity of each pair of nodes is calculated in trees, and the proximity is divided by the number of trees to normalize it. The proximity between the node and itself is 1. In this way, a symmetric proximity matrix is generated, where each value has a range of [0, 1]. The proximity is repeatedly calculated by multiple iterations, and finally, the average value is taken as the final proximity matrix. Using the obtained proximity matrix as an input, the data points are clustered using a Partitioning Around Medoids (PAM) clustering algorithm. In addition, they used the out-of-bag data that were discarded in the sampling to estimate the error of the classification. This estimate has proven to be unbiased. The classification accuracy of this method is related to the number of clusters. When the number of clusters is set to 200, the accuracy can reach 93%. This method can identify unknown network traffic, but it cannot perform online classification in real-time. And this method cannot satisfy robustness. Its granularity is at Level 3.

Huang et al. [89] proposed a high-accuracy APPlication Round method (APPR) for traffic classification, which determines the features of traffic data flows from the application layer rather than the transport layer. They defined the concept of interaction rounds. An interaction round consists of a series of transport layer data segments transmitted in one direction and a series of data segments transmitted in a reverse direction. The first few inter-action rounds of a flow are used to identify the flow. The flow is identified mainly by different behaviors during a negotiation stage. In order to not involve the payload content of the packets, they used the network layer and transport layer headers to calculate statistical features. Experiments were performed using the machine learning tool WEKA [90] to evaluate the method using six different machine learning algorithms. Experiments showed that among the six methods, J48 has the best classification accuracy, the highest accuracy can reach 99.21%, and the average overall accuracy is 92.88%. This method can identify application traffic at an early stage, shortening model-built-duration and test-duration to varying degrees, so it can be used for online classification. However, unknown traffic cannot be detected. It cannot satisfy robustness. The granularity of this method is at Level 3.

In [91], based on the LOA method proposed in [64], Fahad et al. further proposed another method that can select optimal features and automatically find stable features from network traffic based on a Global Optimization Approach (GOA). GOA consists of three phases. The first phase is the same as the LOA method, that is, multiple FS technologies are combined, and a optimal feature set is selected in different datasets. In the second phase, the concept of maximum entropy is applied to estimate the probability distribution of the flow characteristics. The stable and robust features were selected based on the probability distribution. Because it is generally believed that features with distinct distributions are considered stable. They used an adaptive threshold to select features. The third phase uses a random forest filter to select the most representative features from the features obtained in the previous two steps for classification. In addition, they discretized features because this can help improving classification accuracy and speeding up classification. The classification accuracy of this method reaches 97.7%. However, unknown traffic cannot be recognized. And this method cannot support online classification. Its granularity is at Level 4.

Since most classification methods need to process all packets, this causes high resource consumption. Random Packet Sampling (RPS) could reduce the accuracy of the classification although it increases classification speed. So, Zander et al. [92] proposed a Sub-flow Packet Sampling (SPS) approach to reduce resource usage while keeping classification time within 1 s and maintaining classification accuracy around

98%. The SPS method takes W seconds as a time slot, and samples N consecutive packets in a flow in each time slot. They skipped the first O (O fairly small) packets of the flow to further reduce processing time because an active flow would last for a long time. They improved the method that samples data packets in a certain time interval by using the Bloom filter proposed in [93], because sampling starts after skipping O data packets. The authors focused on the identification and classification of VoIP and First Person Shooter (FPS) game traffic. This method can achieve online classification in real-time but cannot satisfy robustness and IUA. Its classification level is unknown.

Fukumoto et al. [94] categorize smartphones by whether they are active or not, because some of the applications or services (such as VoIP, Web browsing, etc.) running on the smartphone are quality-sensitive, which means best-effort is required, it is likely that the running of these applications on active state will affect the user quality of experience. The main feature of this scheme is to automate the periodic estimation of whether the smartphone is active, and then combine the time stamp as the training dataset of machine learning. First, the Logging Application collects the data generated on the smartphone and uploads it to the training data generator, which categorizes the data into active and inactive data based on user behaviors, and links a time series of active/inactive states with collected traffic data for each smartphone. Then, the machine learning module performs supervised learning based on the data provided by the training data generator. The classification granularity is rough because there are only two categories, active and inactive. It can be used for online classification, and the scalability and robustness of the scheme is verified in the LTE environment. However, it cannot support IUA.

Many existing classification methods ignore the problem of a mislabeled training dataset, which leads to poor classification performance when the training dataset is mislabeled. [95] is mainly aimed at this problem. The Noise-resistant Statistical Traffic Classification (NSTC) method can filter the noisy training samples so that the reliable training samples will be kept for traffic classifiers. This method is divided into two processes: offline training and online classification. In the process of offline training, firstly, data are aggregated into flows, then multiple classifiers are used to identify whether the flows are noise data or not, and non-noise data are selected as training datasets. They used a combination of several classification algorithms [96] to identify noise traffic data using a consensus filtering strategy. In the online classification, the scheme uses a classifier based on random forest to classify the online data. The classification accuracy can be kept above 80% under different noise ratio. This scheme supports online classification, robustness, but not satisfy IUA. Its granularity is at Level 3.

The main purpose of [97] is to propose a classification method which can deal with large-scale network traffic and real-time online classify. They used a parallelized Convolutional Neural Networks (CNN). The CNN model is parallelized by parallelizing the input data. This scheme divides the data into n data blocks RDDs (RDD is a data structure provided by Spark), and then n worker nodes train the n data blocks RDDs respectively. At the end of the training, the Master integrates the n weight file generated by n nodes into an average weight file. The CNN model is updated based on this weight file. The comparison experiment shows that the method has good stability and the accuracy of 99.17% is achieved while the classification time is greatly shortened. This scheme supports online classification, robustness, but not satisfy IUA. Its granularity is at Level 4.

Obaidy et al. [98] propose a method to categorize encrypted traffic generated by social media applications, such as Facebook, YouTube. This method mainly includes the process of data collection, feature extraction, machine learning training and testing. The dataset was collected using Wireshark from end-user machines, which included traffic generated by five applications, namely Facebook, YouTube, Skype, Netflix, and WhatsApp. Then, they use five common feature selection methods (ReliefF, Sequential Floating Forward Selection, Sequential Floating Backward Selection, Sequential Backward Selection, Sequential Forward Selection and Binary Genetic Selection [99,100]) to select 14 features that can maintain high classification accuracy in most cases. Four supervised machine learning methods are used for machine learning training and testing, Multilayer Perceptron (MLP), decision tree, SVM and C4.5. The C4.5 algorithm achieves the best classification accuracy of 88.29% in experiment. Because of its limited application scenarios, it is not robust. And it cannot be used to classify online and identify unknown traffic. The granularity of this method is at Level 2.

AlSabah et al. [101] proposed a real-time encrypted traffic classification method to enhance the performance of the Onion Router (Tor). Tor [102] is the implementation of the second generation of onion routing, which provides a low-latency anonymous communication service. The authors proposed an accurate and real-time traffic classification method to improve Tor's performance and usability, so as to provide users with high anonymous service quality. First, they collected different applications traffic appeared in Tor circuits in order to obtain useful application attributes for classification. By observing upstream and downstream traffic data, they extracted some attributes, such as link duration, total transmission volume, and cell arrival interval. They then used four different supervised machine learning algorithms, namely Naïve Bayes, Bayes Network, Functional Tree (FT) [103], Logistic Model Tree (LMT) classification [104] for classification based on the dataset collected in a Tor network, which includes three types of traffic: Bulk transfer, Interactive, and Streaming. The best classification accuracy can reach over 95%. This classification method can support online classification, but its robustness is weak. It also lacks ability to identify unknown traffic. Its classification granularity is at Level 4.

Muliukha et al. [105] proposed a method that mainly classifies the traffic generated by virtual connection technologies. There are two levels of virtual connections, one is technical virtual connections (TVC), and the other is information virtual connections (IVC). This paper focuses on the traffic generated by TVC. In addition, this method also classifies Virtual Private Network (VPN) connection traffic, which is a commonly used encrypted transmission technology. For TVC traffic, the features used in classification mainly contain a total of 67 flow-level features, such as the total number of packets in a flow, the number of packets with payload, and the number of packets with flags. For VPN connection traffic, the method uses both packet-level and flow-level features, such as IP address, port, flow duration, total number of packets transmitted in a flow, and packet inter-arrival time. In terms of datasets used in experiments, they used Network Namespaces to collect specific application traffic. For TVC traffic, they used a Naïve Bayes classifier to classify the traffic at the application level. And the VPN connection traffic mainly includes seven different types of traffic, such as Email, Chat, Browsing, VoIP, etc. They used the random forest algorithm for classification, and achieved classification accuracy as 87.9%. This method is not sufficiently robust. Online classification and IUA cannot be supported. Its classification granularity is at Level 4 and Level 2.

4.1.2. Unsupervised classification

Bernaille et al. [106] proposed a new method to identify the applications associated with TCP flows by only using the size of the first few packets of each TCP flow. They used an unsupervised clustering method (a K-Means algorithm) to aggregate the flows with common behaviors into clusters by monitoring and analyzing the first few packets of the TCP flows, as while as the supervised clustering with a pre-labeled set of samples to construct a model for each cluster. By analyzing a dataset containing 10 applications, they found that only the first five packets of the flow need to be analyzed to identify the application corresponding to the flow. The method firstly trains and learns the dataset offline, and uses the first 5 packets to form a 5-dimensional space. Each dimension uses the length of a packet as a coordinate, and uses the Euclidean distance to measure the similarity between the flows. For a newly arrived flow, the method compares its distance from the center of each cluster and classifies it into the nearest cluster. The

authors used a payload analysis tool, Qosmos, to associate each flow with a corresponding application. Then the two sets of offline learning phase outputs were used for online classification. This method can achieve online classification, but the classification accuracy is not high enough, which is just over 90% in a few application protocols. It cannot satisfy robustness and IUA. Its classification granularity is at Level 2 and 3.

Ahmed et al. [107] proposed an unsupervised and nonparametric clustering method using the Dirichlet Processing Mixture Model (DPMM). They classified normal traffic and attack traffic with application fingerprints. This classification method is roughly divided into two parts: using packet-level features for normal network traffic classification and using flow-level features for attack traffic detection. Firstly, they extracted the packet-level features from datasets for generating the application fingerprints, classified the obtained fingerprints using unsupervised clustering methods. The resulting cluster classes were mapped to applications by using the labeled training data obtained with supervised machine learning, and the clusters that were not mapped are marked as unknown. A multi-modal probability distribution was formed by using the mapped clusters, which were classified by modeling the normal traffic distribution of the application. Secondly, the flow-level features were merged with the package-level features. For extracting extended flow-level and packet-level features of unknown clusters, they used DPMM to analyze extended features to identify normal or attack traffic, mainly for detecting and classifying DDoS attacks (e.g., Slowloris and flooding attacks). This method can support IUA but cannot satisfy robustness and online classification. Its granularity is at Level 4.

Regarding classification of unknown traffic, Zhang et al. [108] proposed an unsupervised classification method. Traffic was divided into different categories based on applications by using the statistical features of flows and packet payloads. In the training phase, they aggregated traffic into clusters based on flow features. The bag-of-words model was introduced to represent the payload content of the cluster constructed by the flow statistical features. According to the payload content of the clusters, a Latent Semantic Analysis (LSA) method was used to analyze cluster similarity, and similar clusters are integrated by using a packet payload clustering method. The corresponding application was identified based on the specific string contained in the payload of the cluster. In the testing phase, in order to protect user privacy, the payload content of the packet was no longer used, and only the statistical information of the flows was used for classification. This classification method can identify unknown traffic and is very robust, but cannot be used for online classification. Its granularity is at Level 2.

In order to provide deep network visibility to network operators, Grimaudo et al. [109] proposed an adaptive method named Self-Learning Classifier for Internet Traffic (SeLeCT). This approach can provide network operators with specific traffic categories that are not even known by the operator. They used an unsupervised method (based on K-means algorithm) and an iterative seeding approach to automatically identify and classify traffic. Unsupervised data mining techniques automatically group flows into pure (or homogeneous) clusters with simple statistical features. In order to improve the homogeneity of clusters, they used an iterative clustering method to filter outliers in the clustering process. This method makes the overall homogeneity of the clusters obtained close to 100%, and the classification accuracy is close to 98%. An iterative seeding approach was used to improve the ability to classify new protocols and applications. Traffic categories are based on a specific application and can even be distinguished by different service types. This method can be used for online classification, and has good robustness, and can also classify unknown traffic. The granularity of this method is at Level 2.

4.1.3. Semi-supervised classification

Mahdavi et al. [110] proposed a method to classify encrypted traffic at the application layer. The method was divided into two steps: training and classification. In the training phase, unlabeled data is clustered firstly. The graph theory and Minimum Spanning Tree algorithms were used here to complete the clustering of flows. In order to optimize the number of clusters, they merged the clusters with the same label regarding intra-cluster distances. A label propagation technique was then used to label the generated clusters by using smaller tagged datasets. In the classification phase, the obtained labeled clusters were taken as inputs, and the statistical features of the data were used to construct a classification model by using the C4.5 algorithm. Then the classification model can be used for subsequent classification. They used four public datasets to validate and evaluate the proposed method. They measured the detection rates of SSH [111] and NOTSSH in different datasets and evaluated the relationship between standard deviation, number of clusters, and detection rate. However, this method cannot support robustness, online classification and IUA. Its classification level is at Level 4.

Vlăduţu et al. [112] had the same classification idea as [110]. That is, the unsupervised clustering algorithm was first used to aggregate traffic flows into clusters according to the similarity between the data flows. The clusters were then used as training input to construct a training model for classification using supervised learning methods. In this article, the unsupervised K-means algorithm was used to implement the clustering process, and the supervised C4.5 algorithm was used to train the classifier. They converted the dataset in the form of .pcap generated by the application traffic simulator Ixia BreakingPoint [113] into unidirectional flows and bidirectional flows, and selected the 25 most relevant statistical features from the data flows, such as the number of packets contained in the flow, the flow duration, etc. Through experiments, they compared the classification accuracy under different numbers of clusters. It was found that when the number of clusters was 15, the classification performance was the best. At this time, the classification accuracy of the unidirectional flow was 85%, and the bidirectional flow was 86%. This method cannot support robustness, online classification and IUA. Its classification level is at Level 2 and 3.

Ran et al. [114] proposed a Self-adaptive Semi-supervised Traffic Classification System (SSTCS) that can automatically select parameters and increase cluster centers. This approach also aggregates the flows into clusters using an unsupervised clustering algorithm and then maps each cluster to an application-oriented traffic class with the help of a labeled dataset. In order to solve the problem of identifying different protocols that need to set different optimal features, they used fixed features to simplify the process of feature selection. They calculated the information gain of the selected features in the labeled dataset as the weight of the features. These weighted fixed features were used to train to fit into different input protocols. They used the improved K-means algorithm to cluster traffic. This algorithm can dynamically increase the number of clustering centers during the iterative process. The generated clusters were labeled with a probability allocation mechanism (i.e., the clusters are mapped to different application categories), and for clusters that do not contain any labeled flows, they are defined as unknown categories. In addition, in order to analyze unknown categories, the proposed classification system also includes update operations. They picked a certain amount of traffic flows in an unknown category and performed manual inspections. If the selected flows corresponded to a new application, a new category would be added to the system as training data; if the selected flows corresponded to a known category, the unknown cluster would be merged with the corresponding category. So, this method is very robust, can satisfy online classification and IUA. The granularity of this classification method is at Level 3.

4.1.4. Others

Since many feature extraction and selection algorithms cannot extract enough stable and robust traffic features for machine learning algorithms, Shi et al. [115] proposed a novel Feature Selection algorithm Principal Component Analysis-based FS (PCABFS). Firstly, to overcome the problem that complex non-linear characteristics of network traffic cannot be described by Transport Layer Statistics (TLS), they proposed to use Wavelet Leaders Multifractal Formalism to extract multifractal features. They also proposed a feature selection algorithm based on PCA. The algorithm can select optimal and stable features from multi-fractal features to reduce data dimensions and redundancy. Because the proposed method selects stable and robust features, this classification method has improved robustness. They analyzed the differences between multifractal features in different traffic flows and gave reasons. They verified through experiments that the multifractal features used in the paper are indeed superior to the TLS in classification performance. Besides, since the method can classify traffic in a short time, it is suitable for online classification. This method has good robustness but cannot satisfy IUA. The granularity of this classification is at Level 3 and 4.

In order to classify encrypted traffic and shorten classification training time, Zhang et al. [116] proposed a classification algorithm named Stereo Transform Neural Network (STNN). STNN is a multi-classification system that classifies multiple traffic categories by constructing multiple single-class classifiers. Each class classifier includes Long Short Term Memory Network (LSTM) and Convolutional Neural Network (CNN). In the phase of offline training, 23 statistical features at flow level are firstly extracted and converted into an image. The image is sent as input to two LSTM models for processing, and the output is a 3D image; then the 3D image is the input of the CNN module to extract representative features. Finally, a softmax function gives the traffic category corresponding to the representative features. In the phase of online classification, for an underlying testing flow, if only one classifier recognizes the flow, it should be labeled as the category represented by the classifier; if no classifier recognizes the flow, it should be labeled as an unknown flow. For the case that multiple classifiers identify the flow, a voting mechanism is applied to decide final classification. Experiments show that the average accuracy of this classification method is over 99.5%. Moreover, the method is scalable and robust, and does not need to be retrained for identifying new application traffic. It can also support IUA and online classification. The granularity of its classification is at Level 2.

- *Discussions:* We summarize the above survey results about statistics-based classification methods in Table 9. From Table 9, we can see the most frequently used features (e.g., feature number 3, 8, 9, 18, etc.). They are the number of packets, the statistical value of the packet size, the packet inter-arrival time, and the total bytes transmitted. This is mainly because these features are simple, and easy to be obtained and handled. In addition, these features also have good stability and robustness. By comparing the accuracy of the methods based on the classification of statistical information, we find that different methods have large differences in accuracy. Some are as high as 98% and some are only about 80%. Their classification granularity also varies widely. In addition, we find that most of the methods cannot meet all the criteria as we proposed. Only three papers [109,114,116] can satisfy robustness, online classification, and are capable of classifying unknown traffic. Moreover, the classification methods in these two papers can also realize the reclassification of unknown traffic to determine their categories. Overall, in terms of classifying unknown traffic and classification granularity, statistics-based classification methods need to be further improved.

4.2. Correlation-based classification methods

In this subsection, we review and comment main correlation-based classification methods by further grouping them according to machine learning types.

4.2.1. Supervised classification

Zhang et al. proposed several methods that use correlation to classify traffic, of which [117,118] mainly aim at solving the problem of low classification accuracy. The method in [117] obtains the Destination IP, Destination port, Protocol 3-tuple based on the header information of packets, integrates the packets into a Bag-of-Flow (BoF) by using the 3-tuple, and then extracts the features of the flow, such as the number of packets, the Mean, Min., Max., Std dev. of packet size, etc. And then, the features are discretized. Finally, multiple single Naïve Bayes (NB) classifiers and aggregated predictor were used to obtain the posterior probability set and traffic classes of the flows. The aggregated predictor uses sum rule, max rule, median rule, and the majority vote rule to aggregate the flows. This method requires manual labeling of training samples in advance, so it is not suitable for online real-time classification. They choose the features which are not sensitive to the classifier and noisy data to aggregate the flows, so this scheme is robust. Its granularity is at Level 3.

In [118], in the same way, they used BoF and the same features as in [117]. This method is designed to solve the problem of low accuracy when the training samples are too small. They find that the Nearest Neighbor (NN) algorithm has the best performance So they use three flows aggregate strategies based on NN algorithm, AVG-NN, MAX-NN, MIN-NN classifies traffic. Experiments showed that AVG-NN has the best classification accuracy. Because the NN algorithm used in this method is suitable for various complex network environments, it makes the method meet the requirements of robustness. However, this method also needs manual labeling of training samples, so it does not support online classification. It can identify unknown traffic and reduce the number of unknown flows as much as possible to provide classification accuracy. The granularity of this method is at Level 3.

Divakaran et al. [119] proposed a classification method based on the K-NN algorithm that could self-learn and classify non-static traffic. Similarly, the newly arrived packets were preprocessed to generate BoF. The system consists of two main components, in which a classifier component labels the BoF and determines classes, a developer component is used to dynamically select a sufficiently good training dataset and feed it back into the classifier component to improve classification accuracy. This method is very robust and can support online classification because it has a good self-learning ability and can adapt to dynamic real network environments. But it cannot support IUA. It granularity is at Level 3.

Ding et al. [120] wanted to reduce the computational overhead by reducing the number of packets used in the classification process. They established a constraint relationship among flows that have some of the same attributes (such as source IP, source port, and destination IP). For example, if two flows have the same source IP, source port, and destination IP, the relationship between them is considered to be L3SRC, “3” means that three attributes are the same, and “SRC” indicates that the two flows are involved in source side. According to this analysis, they divided the relationships between flows into seven different relationship types, namely L3SRC, L3DST, L2+, L2SRC, L2DST, L1SRC, and L1DST. Based on the different types and expanding windows, the Expanding Vector (EV) of the flow was calculated by using the number of related flows contained in the expanding set. Two methods were proposed to construct the expanding vector. In addition, they constructed an EV tree based on the temporal activity of the flow. In each extending set, there is a dominant application that classifies the entire extending set. They considered five machine learning classifiers, in which the random forest classifier has the best classification accuracy and achieves an accuracy of 99.8%. This method is very robust when facing problems such as packet loss. It can also identify unknown traffic in a dataset. However, it cannot support online classification. Its granularity is at Level 3.

Table 9
Summary and comparison of statistics-based classification methods.

Ref.	Data type	Features	NoF	Analysis methods	Datasets	OA	R	Online	IUA	G
[59]	Flow level	17–19	16	Naïve Bayes	Real	96.7%	N	Y	N	–
[60]	Flow level	2	248	Naïve Bayes	Cambridge's Nprobe Project	96.29%	N	N	N	Level3
[61]	Flow level	12,13,23–30	9	Transfer learning	Cambridge's Nprobe Project	98.7%	Y	N	N	Level4
[64]	Flow level	–	–	Bayes	Moore, Wide, KDD99	–	Y	Y	N	–
[67]	Packet level	–	–	SVM	Cambridge's Nprobe Project	95.9%	N	Y	N	–
[68]	–	–	4	Unsupervised ML	NSL-KDD, GTA/UFRJ, NetOp	97.2%	N	Y	N	Level4
[70]	Flow level	4,15, 17, 22	22	Supervised ML	Self-collection	More than 95%	Y	Y	N	Level4
[72]	Packet level	1, 8	–	Multi-Classification	Self-collection	–	Y	N	N	Level2
[80]	–	–	–	SVM	Moore	98.6%	N	Y	N	Level4
[81]	Flow level	8, 12–14, 20	8	C4.5	Tstat	97.92%	Y	Y	N	Level2
[83]	Flow level	8, 9	–	K-NN, SVM	Self-collection	98.17%	N	N	N	Level1
[85]	Flow level	31–34	4	hierarchical K-NN	Self-collection	98.97%	Y	N	N	Level1-2
[86]	Flow level	3, 4, 8, 14, 16, 17	22	Supervised ML	Self-collection	97%	Y	N	N	Level3
[88]	Flow level	3, 8, 9, 16	20	Random Forest proximity, PAM	Keio, Wide	93%	N	N	Y	Level3
[89]	Flow level	4, 12–14, 16	14	J48, Naïve Bayes, PART,zeroR, oneR, BN(K2)	Self-collection	92.88%	N	Y	N	Level3
[91]	Flow level	3, 12, 16, etc.	–	Random Forest	Moore	97.7%	Y	N	N	Level4
[92]	Flow level	8, 9	–	C4.5, Naïve Bayes	Real	98%	N	Y	N	–
[94]	Packet level	3,8, etc.	10	SVM	Simulation	96.2%	Y	Y	N	–
[95]	Flow level	3,8,9,16	20	Random Forest	ToN,ISP	–	Y	Y	N	Level3
[97]	Packet level	–	249	Convolutional Neural Networks	Moore	99.17%	Y	Y	N	Level4
[98]	Packet level	3, 10–13, 16, 31, etc.	14	C4.5, SVM, MLP, Decision Tree	Self-collection	88.29%	N	N	N	Level2
[101]	–	–	–	Supervised ML	Self-collection	More than 95%	N	Y	N	Level4
[105]	Packet and Flow level	1–4, 10–13, 17, 18	–	NB, Random Forest	Self-collection	–	N	N	N	Level4, Level2
[106]	Flow level	20	–	K-means	–	More than 80%	N	Y	N	Level2-3
[107]	Flow and packet level	17–19	18	Unsupervised ML	CAIDA, ISCX, KAIST	97.5%	N	N	Y	Level4
[108]	Flow level	3, 7–9, 16	20	K-means	Real	More than 85%	Y	N	Y	Level2

(*continued on next page*)

Table 9 (*continued*).

Ref.	Data type	Features	NoF	Analysis methods	Datasets	OA	R	Online	IUA	G
[109]	Flow level	–	–	K-means	Self-collection from ISP	Around 98%	Y	Y	Y	Level2
[110]	Flow level	–	–	Semi-supervised; C4.5	NLANR (AMP), MAWI, DARPA99, Moore	–	N	N	N	Level4
[112]	Flow level	3, 8, 9, 16, 18	25	K-means, C4.5	Simulation	86%	N	N	N	Level2-3
[114]	Flow level	3, 8, 9	9	K-means	Moore	95.8%	Y	Y	Y	Level3
[115]	Flow level	–	–	PCABFS	Self-collection	95.67%	Y	Y	N	Level3-4
[116]	Flow level	–	23	LSTM, CNN	Self-collection	More than 99.5%	Y	Y	Y	Level2

NoF: Number of Features; OA: Overall Accuracy; R: Robustness; IUA: Identify Unknown Applications; G: Granularity. Y: supported or considered; N: not supported, discussed or considered.

4.2.2. Semi-supervised classification

Wang et al. [121] proposed the concept of equivalence sets, each of which contains the flows with the same {Destination IP, Destination Port, Protocol} 3-tuple during a certain period of time. So the flows in the same equivalence set belong to the same type of traffic. They used the Gaussian Mixture Model (GMM) to model data and constraints and applied an unsupervised learning algorithm Set-Based Constrained K-Means (SBCK) to approximate GMM. This method cannot satisfy robustness, IUA and online classification. Its granularity is at Level 3.

In another article [122], Wang et al. proposed a traffic classification method based on semi-supervised learning, which also uses the 3-tuple given in [123]. They used the form of paired mandatory link constraints to express the background information of the network traffic, i.e., the correlation between the packets. A must-link means that two entities must belong to the same cluster. In their scenario, three improved K-means algorithms were used to implement traffic classification, namely COP-KMEANS, MPCK-MEANS, LCVQE. This method can support IUA but cannot satisfy robustness and online classification. Its granularity is at Level 3.

The other two papers [124,125] mainly addressed the problem of unknown applications or flow classification (i.e., zero-day application). In [124], Zhang et al. incorporated correlation information between flows into a semi-supervised learning framework for traffic classification. The framework consists of three modules: flow label propagation, Nearest Cluster Based Classifier (NCC), and compound classification. In the flow label propagation module, a dataset is a mixture of a small pre-labeled flow-set and a large unlabeled dataset. The data was trained and labeled using an unsupervised learning algorithm. In the second module, they built an NCC by using the k-means algorithm and finally used the majority vote rule to classify the BoF in the third module. This method can satisfy IUA and has a better robustness than other semi-supervised classification methods but cannot support online classification. Its granularity is in Level 3. In [125], Zhang et al. also used BoF to express the correlation between flows. They used the k-means and random forest algorithm to identify zero-day applications and improve the accuracy of classification. BoF was used to divide the flows into N+1 categories and the newly identified zero-day application category was added to the classifier to improve the classification granularity. So this method can satisfy IUA. This method can satisfy robustness and RT because it can auto-update parameters to optimize the classification process during online classification. Its granularity is at Level 3.

Ede et al. [126] proposed a method to classify encrypted network traffic generated by mobile-apps. This method can quickly identify previously unseen apps, even there is no prior knowledge about the unseen apps before classification. They used four encrypted traffic datasets and applied Adjusted Mutual Information (AMI) to analyze and rank the features available in the datasets. They finally got some highly-ranked features, such as packet size, arrival interval, source/destination IP address, TLS certificate, etc. Their proposed approach was built upon the observation that mobile apps are composed of different modules, each of which communicates with a static set of network destinations. Based on this character, different communication modules corresponding to different patterns can be discovered. They first aggregated and clustered encrypted flows based on destinations, and then grouped individual flows into corresponding clusters based on temporal correlation. When each cluster has a strong correlation, app fingerprints can be generated based on multiple features and temporal correlation. Finally, the obtained fingerprint was matched with a database of known fingerprints to identify apps or detect unseen apps. This method can recognize unknown apps. It can perform online classification with robustness. Its classification accuracy can reach 89.2%, with granularity at Level 2.

4.2.3. Others

Canini et al. [123] proposed a system by combining hardware and software to automatically identify traffic. The hardware is used to provide minimal packet processing latency and to ensure no unexpected packet loss occurs. The software can support complex functions such as rule creation and flow identification, as well as managing and controlling hardware components. They found that each application had its unique {IP, Port, Protocol} 3-tuple in a given period, so they had a reason to believe that flows with the same {IP, Port, Protocol} should belong to the same application. Therefore, utilizing the correlation between such flows enables to improve the speed of traffic identification, which costs less memory than other methods. Their system consists of two caches, Host Cache (HC) and Flow Cache (FC). Firstly, the matching process is performed on the newly arrived packets in the HC. If the 3-tuple of the packet already exists in the HC, the instantaneous classification can be implemented. If the packets cannot be matched in the HC, the FC is used to classify the packets. The FC classification method is performed by collecting the features of the first few packet headers. Due to its fast classification speed, it can be used for online real-time classification. However, it cannot identify unknown applications and cannot support robustness in a good way. The classification level of this method is unknown.

- *Discussions:* By summarizing the correlation-based classification methods in Table 10, we find that all correlation-based classification methods classify traffic by using flow data. This is not difficult to understand. Because the methods classify all the flows in the flow cluster consisting of the same attributes(such as three-tuples,temporal correlation). The features used by the correlation-based methods are mostly header information (such as

Table 10
Summary and comparison of correlation-based classification methods.

Ref.	Data type	Features	NoF	Analysis methods	Datasets	OA	R	Online	IUA	G
[117]	Flow level	3, 8, 9, 16	20	Naïve Bayes	Wide, isp	90%	Y	N	N	Level 3
[118]	Flow level	3, 8, 9, 16	20	Nearest Neighbor	Wide, isp	About 90%	Y	N	Y	Level 3
[119]	Flow level	8, 9	–	K-NN	Wide, isp	98%	Y	Y	N	Level 3
[120]	Flow level	–	–	–	Self-collection	99.8%	Y	N	Y	Level 3
[121]	Flow level	3, 8, 9, 16	20	GMM, SBCK	Wide, Keio, isp	–	N	N	N	Level 3
[122]	Flow level	3, 8, 9, 16, 17	21	K-means	Wide, Keio, Lbnl, Sigcomm, isp	90%	N	N	Y	Level 3
[123]	Flow level	–	–	–	–	More than 99%	N	Y	N	–
[124]	Flow level	3, 8, 9, 16	20	K-means	Wide, isp	–	Y	N	Y	Level 3
[125]	Flow level	3, 8, 9, 16	20	Random Forest	Wide, Keio, isp	95%	Y	Y	Y	Level 3
[126]	Flow and packet level	1,9–11, 17	–	Semi-supervised	ReCon [127,128], Cross Platform [129], Andrubis [130] and Self-collection	89.2%	Y	Y	Y	Level 2

NoF: Number of Features; OA: Overall Accuracy; R: Robustness; IUA: Identify Unknown Applications;G: Granularity. Y: supported or considered; N: not supported, discussed or considered.

IP address, port number, protocol, etc.), statistical information of the packets (the statistics of the packet size and inter-arrival time, etc.). Most of these methods use supervised machine learning algorithms due to its high classification accuracy. Regarding the evaluation criteria we proposed in Section 3, the correlation-based classification methods need further improvement. Because we found that except for the papers [125,126], other methods cannot simultaneously satisfy the requirements of robustness, online classification and identify unknown traffic. We see that the accuracy of the most correlation-based classification methods is above 90%, and some methods even exceed 99%. However, it should be pointed out that the classification granularity of majority of reviewed methods is to classify traffic at the protocol layer (Level 3). The granularity also needs further refinement.

4.3. Behavior-based classification methods

In this subsection, we review and comment main behavior-based classification methods by further grouping them according to machine learning types.

4.3.1. Supervised classification

Grimaudo et al. [131] improved the behavior-based classification method to refine classification results and used a hierarchical classifier to divide network traffic into more than 20 categories. They used a tree-based structure to divide the classification process into several stages and gradually refine classification results. The benefit of using hierarchical classification is that each sub-classifier only needs to process very small datasets and traffic classes. They selected a appropriate feature set for each sub-classifier from more than 200 features. They also compared a single-stage classifier (called Flat classifier) with a hierarchical classifier by using seven classification algorithms (Naive Bayes, Bayesian Kernel Estimation, Rule-Based, Decision Trees, Neural Networks, Support Vector Machine (SVM), K-Nearest Neighbor (K-NN)). Through comparison, they showed the advantages of the hierarchical classifier and selected an appropriate classification algorithm. Experiments showed that the classification accuracy of hierarchical classifiers is always higher than that of the Flat classifier, and it has good robustness. The hierarchical classifier divides the classification results into "known" and "unknown" at its root node. This paper only discusses the refinement classification for known applications, and there is no discussion about the unknown. This method can satisfy robustness because the hierarchical classifiers used offer good robustness, but cannot support online classification. Its granularity is at Level 2.

Kohout et al. [132] used snapshots of individual user activities to perform context simulation on HTTP-based encryption requests to compensate the inconvenience caused by encrypted information. It can classify network traffic as abnormal or normal. A communication snapshot is defined as a collection of all requests issued by the same user for establishing a TLS tunnel (which is used for HTTP communication) during a 5-minute interval. This communication snapshot is represented by the statistical descriptor of the HTTP connection in a server, that is, a fixed-dimensional real vector is used. Based on the communication snapshot, three machine learning algorithms (Neural Networks, Random Forests, and gradient boosted trees implemented via the Extreme Gradient Boosting algorithm) were used for abnormal traffic identification. They mapped the real vector representing the communication snapshot to the interval [0, 1], with 0 representing normal behavior and 1 representing abnormality. This method can be combined with other methods as a pre-filtering tool, by adjusting a recall rate and precision rate mode to filter normal flows or abnormal flows, so as to further analyze retained flows.

4.3.2. Unsupervised classification

Iliofotou et al. [133] combined network-wide behaviors with flow-level features and used Traffic Dispersion Graphs (TDGs) to classify each set of flows. They used directed graphs G(V, E) to represent TDGs, where V is the set of IP addresses of the nodes contained in the flow set, E represents a collection of connections between nodes. That is to say, TDG indicates the communication relationships between the nodes. This graph-based classification framework (called Graption) consists of three steps. The first step is flow isolation, i.e., using port, IP or

payload information to isolate the already classified flow. The second step is flow grouping, the flows generated by the same application are aggregated into groups, but at this time, the specific application category represented by each group is not known. The third step is to label the groups. By constructing a TDG for each group generated in the previous step, they classified groups based on the selected metrics and rule sets. When this framework is used for the classification of P2P traffic, it first isolates legacy applications using a port-based classification based on the usually used default ports of the legacy applications. Then they used a payload-based approach to group the flows. They analyzed the first 16 bytes of each flow and used the K-means algorithm to generate clusters. Then, they applied the Hamming distance to measure the similarity between the clusters. The next step was to generate TDG according to the flow group, and marked each group by metric to identify P2P traffic. Although the method presented in this paper mainly identifies P2P traffic recorded a dataset, it can also be used to classify other categories of traffic by changing parameters and metrics. So it offers robustness. However, it cannot identify the unknown traffic category and cannot support online classification. Its granularity is at Level 4.

4.3.3. Others

In 2005, a behavior-based host traffic classification method named BLINC was firstly proposed by Karagiannis et al. [56]. This method analyzed the host behavior patterns from social, functional and application layers. At the social layer, it acquires other hosts that communicate with the host. At the functional layer, it obtains the host information from the host's role in the network, generally to identify whether the host is a server or a consumer. At the application layer, it captures the interaction behaviors between the host and a specific host on a specific port. They proposed a new concept called graphlets. Each graphlet in a graphlet library identifies the most common behavioral features of an application. Each graphlet consists of 4-tuple {source IP, destination IP, source port, destination port}. They identified an application by matching its graphlet that is closest to the host's behavior. Then, according to the characteristics of the flow, such as the protocol, the size of the packet, etc., the classification is further refined. This method can find unknown applications in a dataset. But it cannot satisfy robustness and online classification. Its granularity is at Level 4.

Qu et al. [134] evaluated a traffic morphing strategy for behavior-based classification schemes through a series of experiments. Traffic morphing is a technique that masks or modifies the statistical features of packets in order to protect user privacy or hide attacks. This article divides the existing traffic morphing strategy into two types: packet size (PS) based and packet inter-arrival time (IAT). The PS morphing strategies modified the packet size and IAT modified the packet inter-arrival time. They used three typical traffic classification algorithms SVM, Bayesian, and C4.5 to evaluate the impact of traffic morphing on classification, and evaluated the performance of the above two morphing strategies. They pointed out that the combination of two strategies has the best performance for weakening traffic classification, while the PS-based strategy has the worst performance.

Choi et al. [135] proposed a four-phase classification system. The first phase is to convert packets into two-way flows that are required for classification. The second phase uses four different classification schemes to classify the generated flows. These schemes are head signature classifier, statistical signature classifier, payload signature classifier, and behavior algorithm classifier. The flows generated in the first phase are respectively passed into the four classifiers to obtain respective classification results. In the third phase, i.e., a flow integration (FI) phase, the FI selects the classification result according to the priority of the four classifiers. Wherein the behavior algorithm classifier has the highest priority, the payload signature classifier has the second priority, and the head signature classifier has the lowest priority. The fourth phase classifies the flows that are not classified in the second phase by using a correlation algorithm classifier, and takes the integrated flow obtained in the third phase as an input, and uses three correlation algorithms to classify the additional flows. This integrated classification method uses the first few packets of a flow to classify the flow and complete the classification before the end of the flow, so it can be used online for real-time classification. Besides, this method can satisfy robustness and support IUA. Its granularity is at Level 2.

Glatz and Dimitropoulos [136] aimed to analyze and classify one-way flows. One-way flow refers to the traffic connection that does not accept network reply. Generally speaking, the one-way flow has the following sources: failed access due to failure or policy; attack traffic (such as vulnerability scanning, DoS attack, etc.); peer-to-peer applications etc. Through observation, they found that malicious scanning accounted for the largest proportion of the one-way flows, followed by P2P data. They monitored and analyzed the communications between the hosts and created a set of signs, including four types of sign categories: Host-pair signs, Remote host signs, Local host signs and Flow signs, for a total of 17 signs. They classified the flows into seven categories, including the unknown, and proposed 13 rules. They classified flows by matching the flows with the rules. For conflicting classification results, this method can reclassify them. And it also updates and refines the rules in the classification process. They reduced the conflicting flow through continuous iteration and repeated classification until all conflicts disappear and unclassified flows can no longer be reduced. Therefore, this method can classify unknown traffic and is also scalable. But it does not satisfy robustness and online classification. Its granularity is at Level 4.

- *Discussions:* We summarize our review on the behavior-based classification methods in Table 11. By analyzing Table 11, we find that the behavior-based classification methods prefer to use the header information of packets as features, and commonly used IP, port, and protocol. Other features such as interaction information between different hosts within the network are also widely used. The data used by the methods include flow level, packet level, and connection level data. Because communications are between hosts, the flows used by the behavior-based classification methods are two-way flows, except for analyzing abnormal attack traffic since this kind of analysis often uses a one-way flow. Using IP addresses, ports, and connection relationships between hosts, the behavior-based classification methods use the concept of graph to represent behavior patterns between hosts. From the papers we reviewed, we find that only one behavior-based method can classify traffic online in real-time with robustness and detect unknown traffic simultaneously. In addition, the classification granularity is not fine enough if it only relies on the header information of packets. This causes that the traffic was classified on the applications categories level, i.e., Level 4. Using more statistical features at the package or flow level can help us achieve fine-grained classification results.

4.4. Payload-based classification methods

In this subsection, we review and comment payload-based classification methods by further grouping them according to machine learning types.

4.4.1. Supervised classification

Kampeas et al. [137] proposed a precise online classification method based on Zero-Length Packets for TCP data. The Zero-Length Packets only contain control bit information without payload. Their method is based on their findings: the exchange of data between applications follows a specific features pattern, which can be used to identify the application. They built Application Protocol Data Units (APDUs) fingerprint sequences based on Zero-Length Packets and used the J48 decision tree algorithm in machine learning to classify traffic. They pointed out that the method is not affected by network delay, packet

Table 11
Summary and comparison of behavior-based classification methods.

Ref.	Data type	Features	NoF	Analysis methods	Datasets	OA	R	Online	IUA	G
[56]	Flow level	10–13	–	Feature matching	CAIDA	More than 95%	N	N	Y	Level4
[131]	Packet level	–	More than 200	Hierarchical classification	isp	More than 90%	Y	N	Y	Level2
[132]	Connection level	35,36	–	Neural Networks, Random Forests, XGBoost	Self-collection	–	–	–	–	–
[133]	Flow level	6	–	K-means	Self-collection	95%	Y	N	N	Level4
[134]	–	–	–	–	UNIBS 2009	–	–	–	–	–
[135]	Flow level	3, 5, 10–14, etc.	–	–	Self-collection	97.94%	Y	Y	Y	Level2
[136]	Flow level	–	17	–	Self-collection	–	N	N	Y	Level4

NoF: Number of Features; OA: Overall Accuracy; R: Robustness; IUA: Identify Unknown Applications; G: Granularity. Y: supported or considered; N: not supported, discussed or considered.

loss, re-transmission, and congestion, etc., so it has good robustness. They extended this method to UDP data. Experiments showed that this method is applicable for UDP data with high accuracy. This method can identify unknown traffic categories and it can support online classification in real time network environment. Its granularity is at Level 2 and Level 3.

Liu et al. [138] proposed a three-level classification scheme that incorporates multiple classification methods. First, the packet header is checked by the ServerTag method proposed in [139], and unknown traffic could be identified quickly. ServerTag cannot identify the traffic generated by an unknown server and a server hosting multiple applications, so this type of traffic will be classified in payload level and flow level. At the payload layer, unencrypted data is classified using Payload Distribution Inspection (PDI). The further step is classifying encrypted data in the flow layer, which cannot be recognized using PDI. The classification in the flow layer uses a random forest algorithm according to the characteristics of the data flow. This is because the random forest algorithm has the best performance compared with other algorithms. They divided the classification results into six categories, social, service, streaming, web, mail, and unknown. Thus, the granularity of the classification is not fine enough. This method can support IUA. This method has good stability and can adapt to most classification scenarios, so it is robust, but it cannot support online classification. Its granularity is at Level 4.

Wang et al. [140] proposed a method of traffic classification by using packet payload. This method first extracted the first N bytes of the flow payload. For a large number of byte strings of a given application class, a common substring extraction component was used to extract their common substring. They consolidated all application tokens into a single token set and deleted duplicate tokens. To further reduce the computation overhead, feature extraction algorithms were used to reduce the number of tokens contained in the token set. Finally, six machine learning algorithms (Naïve Bayes, Bayesian network, Multilayer perception, C4.5, random forest, AdaBoost) were used to evaluate the classification performance of this method. Experiments showed that the classification accuracy of this method can reach more than 99.5%. This method can be used for online classification because it has a low computational load and uses only the first N bytes of the flow. And this method can satisfy robustness, but cannot support IUA. Its classification granularity is at Level 2 and 3.

Finamore et al. [141] proposed an architecture that uses the statistical features of the payload to classify UDP traffic, named Chi-Square Signatures (KISS). A statistical packet inspection method was used to automatically identify the format of application protocol. It ignores the semantics and synchronization rules of the application. The statistical signatures of payload were obtained by chi-square-like test. Use the DPI tool (Tstat) to execute the KISS method and classify the P2P traffic from the application level. Geometric decision based on Euclidean distance and SVM-based decision were used to compare. They used a confusion matrix to represent the performance of the classification. Through experiments, they found that 98.1% of True Positive (TP) can be achieved using SVM. This method has inherent robustness to packet sampling, packet loss, reordering, and flow asymmetry, so it can be applied to most networks. And it does not require high computational and storage consumption, suitable for online classification in real-time. Besides, it supports IUA also. But like other payload-based classification methods, this method has the disadvantage of not being able to classify encrypted traffic. Its granularity is at Level 2.

Yang et al. [142] proposed a method that uses the content of handshake packets to classify encrypted traffic. Because end-hosts need to establish connection through handshake before performing encrypted secure communications. The handshake information used to authenticate each other is unencrypted. Therefore, the handshake information can be used to classify encrypted traffic. This method focuses on the classification of data traffic that uses the Transport Layer Security (TLS) protocol for encrypted communication. In the handshake phase of TLS, the server and client need to negotiate an encryption algorithm and secret session keys that will be used in data communication. This information can be used as features to identify which application generates the current traffic. So, they used cipher suites and compression methods [143], and TLS extensions [144] as input features, and applied Bayesian Neural Network for classification. The experimental dataset contains four different categories of traffic, namely web, mail, file, and VoIP. For each category, this method's classification precision and recall rate can reach 99%. However, this method cannot classify unknown traffic and perform online real-time classification. The granularity of its classification is at Level 4. In addition, this method is mainly for classifying TLS protocol data, which has somehow limitations, so its robustness needs to be further improved.

4.4.2. Unsupervised classification

Park et al. [57] proposed a new approach aiming to divide classification results into refined categories rather than improving classification accuracy. They further subdivided the traffic that generated by a single application into different traffic groups. They used the algorithm that was proposed in [145] to automatically generate application signatures and used document retrieval techniques to construct fine-grained traffic classifiers. Document retrieval technology judges the similarity between documents by the frequency of keywords in the document. They used similarity scores to divide the data flow into different flow groups. When the similarity score is less than a threshold, a classifier can create a new flow group in the classification process, where unsupervised

machine learning techniques were used to implement the classification process. So this method can identify unknown traffic. But its robustness is not good and it cannot online classification. Its granularity is at Level 1.

4.4.3. Semi-classification

Wang et al. [5] proposed a traffic classification method that does not require any prior knowledge to identify and classify unknown traffic. First, the flows were classified online. For unknown flows that cannot be identified online, an unsupervised clustering algorithm X-means proposed in [146] was used, which aggregates clusters according to the statistical features of the flows. The clusters were dominated by a single application. Then, the authors used a supervised machine learning method to build application signatures based on the payload and classified unknown flows into new application signatures. The offline generated application signatures were inserted into the online classifier to update the application signature to classify unknown traffic. Through experiments, the overall accuracy of the classification method can reach more than 99.5%, and this method implements the classification of unknown applications. This method can satisfy robustness and IUA and online classification. Its granularity is at Level 3.

4.4.4. Others

Mayank and Neminath [147] proposed a method to classify text-based protocol traffic, named RDClass. The method extracts keywords from the packet payload and uses the relative distance between keywords to identify an application. The keywords and relative distances were encoded in the form of a state transition machine. They proposed a new state transition machine, named Relative Distance Constrained Counting Automata (RDCCA). This method is able to generate one or more state machines for each application in the order of the keywords. The RDClass classifier consists of three components, the first is a flow constructor to aggregate packets into a flow, in which the packets have the same five-tuple. The second component is a term extractor, which uses newline characters, spaces, and special characters as delimiters to parse the content extracted from a payload to generate terms. The third component uses the state transition machine to process terms. The generated terms act as input to a series of state machines. Each state transition machine reads the term and performs an allowed state transition. If any of the state transition machines moves to an accept state, the flow is classified. Otherwise, the flow will not be classified. For unclassified flows, the RDClass method can generate RDCCA for further classification. Experiments showed that the method has a classification recall rate of over 99%. This method has good robustness and scalability in most text-based protocols cases, but it is not suitable for online classification, and it is not able to classify encrypted data. It cannot support IUA as well. Its granularity is at Level 3.

Khandait et al. [148] proposed a classification method based on Deep Packet Inspection. Because of the common payload-based classification methods involve two scanning process, first scanning is to extract the keywords, and second scan to match the keywords with characteristics of the applications. This method completed classification with only one scan. They use application-based keywords and keywords ordering, combined with a heuristic-based approach to speed up classification. The scheme consists of two parts. The first part is to generate signatures for each application, which is extracted from the first K byte of each application's flow. The signature includes both common and application-specific keywords. The second part is to use these signatures to classify the data flows. The classification accuracy of this method is up to 98%. This classification method is only suitable for unencrypted data, and does not satisfy online classification, IUA, and robustness. The classification level is in level 3.

Marin et al. [149] proposed a method based on deep learning (DL) to classify malicious network traffic. Because the classification method based on manually extracted features has certain limitations when facing a real and complex network environment. Deep learning can automatically learn representative features from raw data, so using deep learning can solve this problem well. This method builds a DL model based on the input byte flow without any sort of expert handcrafted inputs or data pre-processing process. The DL architecture of the raw packets and the DL architecture of the raw flows were constructed in the paper to evaluate and compare the performance of the DL architecture at the packet level and the flow level. The classification architecture was tested using publicly available attack detection dataset [150], and three different malicious attack traffic were detected and classified, namely Neris, Rbot, and Virut. The experimental results were represented with classification accuracy: the accuracy of Rbot can reach 99.9%, while the accuracy of Neris and Virut are 63.5% and 54.7%, respectively. This method has good robustness, but does not support the classification of unknown traffic and online classification. The classification level is relatively rough, at Level 4.

- *Discussions:* We summarize our review on the payload-based methods in Table 12. From Table 12, we find that major payload-based classification methods tend to use flow-level data. We observe that the mostly used features are the first few bytes in a packet or the first few packets in a flow. Besides, some methods use the specific sequence of the packet payload as a keyword or signature to classify traffic. By comparing the payload-based classification methods with other types of methods discussed above, the overall accuracy of the payload-based classification methods is higher, all exceed 95%. There are some methods that can reach 99%, and even 99.5% in [5,140]. In addition, by analyzing classification granularity and accuracy, we can see that the method with higher classification accuracy has a rougher classification granularity, the method with relatively lower accuracy has a finer classification granularity. Of course, classification accuracy is not only affected by the granularity of classification. The features, classification algorithms, etc. all affect accuracy. Taking [138] and [141] as examples, they both used the first few bytes of the packet. The method proposed in [138] classifies traffic at the level of the application type, that is, its classification granularity is rough, with classification accuracy as 99%. The method proposed in [141] classifies traffic at the application level, its classification granularity is finer than [138], with classification accuracy as 98.1%, lower than [138]. This is very easy to understand because the finer the classification granularity is, the harder to classify.

4.5. Port-based classification methods

In this subsection, we review the port-based classification methods. Since many applications no longer use known fixed ports but tend to use dynamic ports, the simple port-based classification methods can no longer meet the requirements of traffic classification, so research in this area is very rare. Here we briefly introduce a couple of methods fallen into this category.

Lin et al. [151] combined each connected Packet Size Distribution (PSD) with a port to identify an application. This method contains two phases, offline training and online classification. Since each application has its independent PSD, it can be identified by analyzing the PSD of each connection. After obtaining the PSD, they compared it with the representatives of all pre-defined applications, and calculated the Euclidean Distance. Classification was made based on one of their findings: an application tends to communicate using a consecutive port in a single session. Specifically, if two connections have the same Source IP address and destination IP address, as well as similar ports, they could be considered as generated by the same application. This method has a high classification accuracy for some applications such as Apache Server, which reaches an average of 98%, while other applications, such as Skype-Voice, have a low classification accuracy of only 74%. For the new traffic that did not occur in training phase, this method classifies them into unknown. This method can support IUA and online classification but cannot satisfy robustness. Its granularity is at Level 2.

Table 12
Summary and comparison of payload-based classification methods.

Ref.	Data type	Features	NoF	Analysis methods	Datasets	OA	R	Online	IUA	G
[5]	Flow level	3, 8, 9, 16, 17	11	X-means	Self-collection	More than 99.5%	Y	Y	Y	Level3
[57]	Packet level	–	–	Unsupervised ML	Self-collection	–	N	N	Y	Level1
[137]	Packet level	21	–	J48 decision tree	Simulated data	More than 97%	Y	Y	Y	Level 2-3
[138]	Packet and flow level	–	–	Hierarchical classification	Self-collection	99%	Y	N	Y	Level 4
[140]	Flow level	6	–	Supervised ML	Self-collection	More than 99.5%	Y	Y	N	Level 2-3
[141]	Flow level	6	–	SVM, Euclidean distance	Self-collection	98.1%	Y	Y	Y	Level 2
[142]	Packet level	7	392	BNN	Self-collection	–	N	N	N	Level 4
[147]	Flow level	–	–	RDCCA	Real and private data	–	Y	N	N	Level 3
[148]	Flow level	–	–	–	Self-collection, Digital Corpora	98%	N	N	N	Level 3
[149]	Packet level	–	–	CNN	Stratosphere IPS Project	More than 95%	Y	N	N	Level 4

NoF: Number of Features; OA: Overall Accuracy; R: Robustness; IUA: Identify Unknown Applications; G: Granularity. Y: supported or considered; N: not supported, discussed or considered.

Karagiannis et al. [4] proposed a systematic method to classify P2P traffic at the transport layer based on flow connection patterns. They used two heuristics, one is TCP/UDP IP pair, and the other is {IP, Port} pair, to classify P2P traffic. They found that more than half of traffic that uses both TCP and UDP protocols belongs to P2P. For non-P2P traffic that uses both TCP and UDP protocols also, they observed and listed their ports commonly used. If a TCP/UDP IP pair that uses both TCP and UDP protocol and the port it uses is not in the list, it can be considered as P2P traffic. For the second pair of IP, port, the number of different connection IPs equals to the number of different connection ports is considered to be P2P. If the difference between the number of different connection IPs and the number of ports is too big, it is considered as non-P2P traffic. This method can classify unknown traffic. However, since it only classifies P2P traffic and uses only packet header information, the method is not robust. It is also not suitable for online real-time classification. Its granularity is at Level 3.

5. Open issues and future research directions

In this section, we discuss and analyze some current challenges and future research directions of network traffic classification.

5.1. Open issues

According to the above review, discussions, and comparisons, we identify some open problems in the field of traffic classification.

First, it is not easy to get reliable ground truth with regard to dataset collection, which greatly impact the accuracy of traffic classification. An inaccurate ground truth also affects the efficiency of machine learning [152]. However, traditional dataset labeling methods (including manually labeling, port-based labeling and DPI-based labeling) suffer from some problems. These methods are either time consuming or not accurate enough. Some new dataset labeling methods are active measurement [153] and heuristics-based analysis [154]. These methods can achieve high accuracy, but face some other problems, such as increased computational overhead, limited application scope, and so on. Oliveira et al. [155] pointed out that perfect ground truth is difficult or impossible to obtain. Therefore, it is necessary and significant to work out a classification method that does not rely on ground truth too much.

Second, efficiency of classification cannot satisfy the real-time traffic classification requirement in practice. Although there are many classification methods proposed, we find that many of them cannot classify traffic in real-time. As observed from Tables 9–13, more than half of the traffic classification methods do not explicitly consider the real-time of traffic classification. However, it is very important to identify the traffic within a short time after it is generated. The real-time classification is necessary for network operators to improve networking QoS, and observe abnormal network traffic timely for security purposes. Since traffic classification involves multiple processes such as extracting features, training data, and matching features, the real-time problem of traffic classification is still a major challenge in existing research. Researchers need to propose lightweight classification methods to improve classification speed in each of classification stages from multiple aspects.

Third, we find that many studies ignore the classification of unknown traffic. Many existing classification methods classify all traffic into known established categories. This causes classification results not accurate enough in some situations. For example, if the method encounters a traffic category that did not appear during the training phase, it would incorrectly classify this category traffic into a known one. We also notify that there are a few methods that can classify traffic into an unknown category rather than a known category. However, it is rare for them to reclassify unknown traffic in order to determine its real category. The identification and reclassification of unknown traffic require a classification method that continuously and adaptively updates its classification model adaptive to specific traffic. At present, accurate and efficient reclassification of unknown traffic is still an open problem in the field of traffic classification.

Fourth, fine-grained traffic classification lacks accuracy, efficiency and support on IUA. Most existing methods perform traffic classification at the application category or protocol level, i.e. Level 4 or Level 3.

Table 13
Summary and comparison of port-based classification methods.

Ref.	Data type	Features	NoF	Analysis methods	Datasets	OA	R	Online	IUA	G
[151]	Packet level	12, 13	–	PSD	Self-collection	96%	N	Y	Y	Level 2
[4]	Flow level	4–7	–	System methodology	–	94%	N	N	Y	Level 3

NoF: Number of Features; OA: Overall Accuracy; R: Robustness; IUA: Identify Unknown Applications; G: Granularity. Y: supported or considered; N: not supported, discussed or considered.

Some of these classification methods can achieve high classification accuracy. However, there are very few methods that can classify traffic with the granularity of Level 1. Of all the papers we reviewed, there are only two proposed methods [57,83] that can reach this goal. And there are few methods that can classify traffic with the granularity of Level 2. Furthermore, for the methods with classification in Level 2 and Level 1, their classification accuracy still needs to be further improved. Different classification granularity can provide network managers with different information at different levels. More fine-grained classification granularity can help network operators and network administrators to improving network services and optimize network resource allocation. Therefore, obtaining fine-grained classification with high accuracy, efficiency and IUA support still needs further research.

5.2. Future research directions

The above open issues motivate future research. We further suggest some promising future research directions in this field.

First, the classification methods that with low computational load and memory consumption needs to be explored. Many existing classification methods consider a large number of traffic features so that the computational cost is high when performing feature matching in the process of identifying network traffic. These methods spend much computation time and consume a large amount of system resources, which makes classification efficiency very slow. Therefore, lightweight classification methods that only need to analyze few features or packet payloads are highly expected. Meanwhile, classification optimization becomes essential for the purpose of improving classification accuracy and realizing real-time classification.

Second, effective methods for ground truth collection needs further exploration. According to the previous discussion, the methods of obtaining ground truth still face big challenges. Due to the limitations of passive measurement collection methods (that is, port-based and DPI tool based methods), such as the inability to cope with dynamic ports and encrypted traffic, the methods for active measurement and heuristic analysis are expected. At the same time, in addition to further improving the reliability of ground truth, there are other issues that need to be addressed. For example, most active measurement methods use simulated data to test the reliability of labels, which cannot fully reflect the complexity of real traffic. Therefore, it becomes essential to study a reliable and versatile ground truth collection method.

Third, the identification of unknown network applications becomes crucial, especially for security intrusion, attack and threat detection. With the development of network environments, more and more new applications are emerging. This requires classification methods to improve the ability to classify new applications in the constantly changing network environment. In other words, the classification method should be robust and scalable in order to adapt to various network environments. The classification of new application traffic helps us understanding the status of network traffic and identifying malicious applications. Therefore, identifying unknown traffic and determining its category still deserve more efforts.

Fourth, efficient and effective feature extraction request deep research for the purpose of reducing computational load and improving classification speed. The accuracy and efficiency of traffic classification highly depend on the performance of feature selection to a large extent. A low dimensional and irredundant feature set greatly helps improving classification speed and enabling real-time classification. We think feature selection should be further studied in order to reduce the number and dimension of features while maintaining the accuracy of the classification. Most of the existing machine learning classification methods use manually-designed features, which requires additional manpower. Automatically obtaining features directly from original data stream instead of manually designing features can further help improve classification performance. In particular, adaptive feature selection that fits into networking contexts and demands should be deeply pursued [156].

Fifth, traffic classification that can overcome traffic obfuscation with privacy preservation is a very interesting and significant research topic. Traffic obfuscation technology has emerged as a solution to counter traffic classification. Therefore, traffic classification methods need to resist the obstacles caused by traffic obfuscation. Due to personal privacy or other reasons, many network users (especially malicious users) do not want others to analyze the traffic they generate. Therefore, some technologies are used to hide or modify their traffic to obfuscate traffic. Traffic obfuscation techniques have caused confusion or obstacles to traffic classification to a large extent. By analyzing the existing traffic classification methods, we observed that there is almost no method to consider and deal with the impact of traffic obfuscation. However, the issue of traffic obfuscation should be paid special attention in the future research of traffic classification.

Sixth, we should pay special attention to the classification of encrypted network traffic. Because network users pay more and more attention to personal privacy, this makes efficient privacy-preserving classification become essential in a privacy-focused networking environment [157]. Due to privacy concern, a lot of traffic is encrypted for transmission. Encrypted communication connections are also widely used in various network fields, such as the Internet of Things, cloud computing, and blockchain and so on. The existing methods for classifying encrypted traffic need to be further improved in terms of scalability and accuracy. For example, deep learning, collaborative learning, meta-learning and other technologies could be applied to improve the performance of encrypted traffic classification. Using the self-learning and transfer learning capabilities of these technologies could further reduce the dependence of classification methods on classification features to a certain extent.

Seventh, it is highly suggested to investigate novel classification methods that can offer high efficacy regarding accuracy, fine granularity and efficiency, especially when a small number of training samples with labels are available. This is very beneficial for practical applications due to high cost of real-world data collection and labeling. In particular, embedding a knowledge graph [158] including correlation relationships [159] into learning constraints to instruct classification could greatly improve classification efficacy [160], especially when the number of labeled samples is small. From this point of view, constructing proper knowledge representation and computation like Zheng's method [158] becomes significant for improving classification quality towards wide adoption in practice. Thus, exploring how to effectively integrate a knowledge graph into traffic classification becomes a very interesting research topic. We hope this study can open a new research direction in traffic classification.

Last but not the least, we should further extend the basic requirement on classification robustness to allow traffic classification to effectively resist potential attacks, such as adversarial learning, malicious data provision, untrusted data collection, false training, etc. Future traffic classification should be intelligent to judge potential attacks during classification execution.

6. Conclusion

This paper gave a thorough review on the state of art of network traffic classification. We first introduced traffic features, summarized research datasets, and specified traffic classification granularity. We then reviewed feature selection and classification algorithms that are widely used in traffic classification. We further proposed a set of criteria for evaluating the performance of traffic classification methods. By employing the proposed criteria, we comprehensively surveyed and compared the traffic classification methods by classifying them into five categories, namely statistics-based, correlation-based, behavior-based, payload-based and port-based. At the end, we indicate some open issues and suggest future research directions for improving the performance of traffic classification. Shortly, high efficiency, low cost, unknown application identification, fine granularity with ensured accuracy, encrypted traffic classification, classification with small labeled data and advanced robustness with privacy concern are promising future research directions in the field of traffic classification.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62072351; in part by the Academy of Finland under Grant 308087 and Grant 335262; in part by the Shaanxi Innovation Team Project under Grant 2018TD-007; and in part by the 111 Project under Grant B16037, as well as Huawei Technologies Group Co., Ltd.

References

[1] F.E. White, Data Fusion Lexicon, 1991.
[2] W. Ding, X. Jing, Z. Yan, L.T. Yang, A survey on data fusion in internet of things: Towards secure and privacy-preserving fusion, Inf. Fusion 51 (2019) 129–144.
[3] T. Meng, X. Jing, Z. Yan, W. Pedrycz, A survey on machine learning for data fusion, Inf. Fusion 57 (2020) 115–129.
[4] T. Karagiannis, A. Broido, M. Faloutsos, K. Claffy, Transport layer identification of P2P traffic, in: Proceedings of the Fourth ACM SIGCOMM Conference on Internet Measurement, 2004, pp. 121–134.
[5] Y. Wang, Y. Xiang, S.Z. Yu, Automatic application signature construction from unknown traffic, in: Proceedings of IEEE International Conference on Advanced Information Networking and Applications, 2010, pp. 1115–1120.
[6] T.T.T. Nguyen, G. Armitage, A survey of techniques for internet traffic classification using machine learning, IEEE Commun. Surv. Tutor. 10 (4) (2009) 56–76.
[7] A. Callado, C. Kamienski, G. Szabo, B. Gero, J. Kelner, S. Fernandes, D. Sadok, A survey on internet traffic identification, IEEE Commun. Surv. Tutor. 11 (3) (2009) 52.
[8] Z. Cao, G. Xiong, Y. Zhao, Z. Li, L. Guo, A Survey on Encrypted Traffic Classification, Springer Berlin Heidelberg, 2014.
[9] Richter, Chris, Finsterbusch, Michael, Muller, Jean-Alexander, Rocha, Eduardo, Hanssgen, Klaus, A survey of payload-based traffic classification approaches, IEEE Commun. Surv. Tutor. 16 (2) (2014) 1135–1156.
[10] V. João, Gomes, R.M. Pedro, Inácio, Manuela, Pereira, Mário, Detection and classification of peer-to-peer traffic: A survey, ACM Comput. Surv. 45 (2013) 1–40.
[11] S. Valenti, D. Rossi, A. Dainotti, A. Pescapè, M. Mellia, Reviewing Traffic Classification, Springer Berlin Heidelberg, 2013.
[12] M. Shafiq, X. Yu, A.A. Laghari, L. Yao, F. Abdessamia, Network traffic classification techniques and comparative analysis using machine learning algorithms, in: Proceedings of the 2nd IEEE International Conference on Computer and Communications, ICCC, 2016, pp. 2451–2455.
[13] F. Pacheco, E. Exposito, M. Gineste, C. Baudoin, J. Aguilar, Towards the deployment of machine learning solutions in network traffic classification: A systematic survey, IEEE Commun. Surv. Tutor. 21 (2018) 1988–2014.
[14] J. Frank, Artificial intelligence and intrusion detection: Current and future directions, Comput. Secur. 14 (1995) 31–31.
[15] CAIDA, http://www.caida.org/data/ (Accessed 08 October 2020).
[16] UNIBS, http://netweb.ing.unibs.it/ntw/ (Accessed 08 October 2020).
[17] MAWI. MAWIWorking Group traffic archive, http://mawi.wide.ad.jp/mawi/ (Accessed 08 October 2020).
[18] Cambridge's Nprobe project, http://www.cl.cam.ac.uk/research/srg/netos/projects/archive/nprobe/data/papers/sigmetrics/index.html (Accessed 08 October 2020).
[19] ISCX, https://www.unb.ca/cic/datasets/index.html (Accessed 08 October 2020).
[20] IP Trace, http://iptas.edu.cn/ (Accessed 08 October 2020).
[21] KDD Cup99, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (Accessed 08 October 2020).
[22] Digital Corpora: 'Producing the digital body', http://digitalcorpora.org/ (Accessed 08 October 2020).
[23] M.A. Lopez, R.S. Silva, I.D. Alvarenga, G.A.F. Rebello, G. Pujolle, Collecting and characterizing a real broadband access network traffic dataset, in: Proceedings of the 1st Cyber Security in Networking Conference, CSNet, 2017, pp. 1–8.
[24] A.G.P. Lobato, M.A. Lopez, I.J. Sanz, A.A. Cardenas, O.C.M.B. Duarte, G. Pujolle, An adaptive real-time architecture for zero-day threat detection, in: Proceedings of IEEE International Conference on Communications, ICC, 2018, pp. 1–6.
[25] The Kaist/Wibro Dataset, https://crawdad.org/kaist/wibro/20080604/ (Accessed 10 October 2020).
[26] The Snu/Wow-Via-Wimax dataset, https://crawdad.org/snu/wow-via-wimax/20091019/ (Accessed 10 October 2020).
[27] R. Pang, M. Mark Allman, M. Bennett, J. Lee, V. Paxson, B. Tierney, A first look at modern enterprise traffic, in: The 5th ACM SIGCOMM Conference on Internet Measurement, 2005, pp. 2–2.
[28] Network traffic tracing at SIGCOMM, http://www.cs.umd.edu/projects/wifidelity/tracing/ (Accessed 10 October 2020).
[29] NLANR, http://pma.nlanr.net (Accessed 10 October 2020).
[30] DARPA, https://www.ll.mit.edu/r-d/datasets/ (Accessed 10 October 2020).
[31] D. Zhou, Z. Yan, Y. Fu, Z. Yao, A survey on network data collection, J. Netw. Comput. Appl. 116 (2018) 9–23.
[32] H. Lin, Z. Yan, Y. Chen, L. Zhang, A survey on network security-related data collection technologies, IEEE Access 6 (1) (2018) 18345–18365.
[33] L. He, Z. Yan, M. Atiquzzaman, LTE/LTE-A network security data collection and analysis for security measurement: A survey, IEEE Access 6 (1) (2018) 4220–4242.
[34] H. Xie, Z. Yan, Z. Yao, M. Atiquzzaman, Data collection for security measurement in wireless sensor networks: A survey, IEEE Internet Things J. 6 (2) (2019) 2205–2224.
[35] H. Lin, Z. Yan, Y. Fu, Adaptive security-related data collection with context awareness, J. Netw. Comput. Appl. 126 (2019) 88–103.
[36] G. Liu, Z. Yan, W. Pedrycz, Data collection for attack detection and security measurement in mobile ad hoc networks: A survey, J. Netw. Comput. Appl. 105 (2018) 105–122.
[37] X. Jing, Z. Yan, W. Pedrycz, Security data collection and data analytics in the internet: A survey, IEEE Commun. Surv. Tutor. 21 (1) (2018) 586–618.
[38] X. Jing, Z. Yan, X. Jiang, W. Pedrycz, Network traffic fusion and analysis against ddos flooding attacks with a novel reversible sketch, Inf. Fusion 51 (2019) 100–113.
[39] X. Jing, J. Zhao, Q. Zheng, Z. Yan, W. Pedrycz, A reversible sketch-based method for detecting and mitigating amplification attacks, J. Netw. Comput. Appl. 142 (2019) 15–124.
[40] Y. Dhote, S. Agrawal, A.J. Deen, A survey on feature selection techniques for internet traffic classification, in: Proceedings of International Conference on Computational Intelligence and Communication Networks, 2016, pp. 1375–1380.
[41] I. Inza, P. Larra?Aga, R. Etxeberria, B. Sierra, Feature subset selection by Bayesian network-based optimization, Artificial Intelligence 123 (1–2) (2000) 157–184.
[42] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, Gene selection for cancer classification using support vector machines, Mach. Learn. 46 (2002) 389–422.
[43] J. Yan, A survey of traffic classification validation and ground truth collection, in: Proceedings of the 8th International Conference on Electronics Information and Emergency Communication, ICEIEC, 2018, pp. 255–259.
[44] I.T. Jolliffe, Principal component analysis, J. Mark. Res. 87 (4) (2002) 513.
[45] M.A. Hall, L.A. Smith, Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper, in: Proceedings of the Twelfth International Florida Artificial Intelligence Research Society Conference, 1999, pp. 235–239.

[46] L. Yu, H. Liu, Feature selection for high-dimensional data: A fast correlation-based filter solution, in: Proceedings of the Twentieth International Conference, ICML, 2003, pp. 856–863.
[47] J. Han, M. Kamber, Data mining: Concepts and techniques, Data Mining Concepts Models Methods and Algorithms, vol. 5(4), second ed, 2006, pp. 1–18.
[48] C.T. Su, J.H. Hsu, An extended Chi2 algorithm for discretization of real value attributes, IEEE Trans. Knowl. Data Eng. 17 (3) (2005) 437–441.
[49] C. Xu, S. Chen, J. Su, S.M. Yiu, L.C.K. Hui, A survey on regular expression matching for deep packet inspection: Applications, algorithms, and hardware platforms, IEEE Commun. Surv. Tutor. 18 (4) (2016) 2991–3029.
[50] nDPI. Open and Extensible LGPLv3 deep packet inspection library, https://www.ntop.org/products/deep-packet-inspection/ndpi/ (Accessed 10 October 2020).
[51] OpenDPI, https://github.com/thomasbhatia/OpenDPI/ (Accessed 10 October 2020).
[52] L7-filter. Application layer packet classifier for Linux, http://l7-filter.clearos.com/ (Accessed 10 October 2020).
[53] A. Finamore, M. Mellia, M. Meo, M. Munafo, P. Torino, D. Rossi, Experiences of internet traffic monitoring with Tstat, IEEE Network 25 (3) (2011) 8–14.
[54] NarusInsight, http://www.narus.com/ (Accessed 10 October 2020).
[55] Internet assigned numbers authority (IANA). Port Numbers, http://www.iana.org/assignments/port-numbers (Accessed 10 October 2020).
[56] T. Karagiannis, K. Papagiannaki, M. Faloutsos, BLINC: multilevel traffic classification in the dark, in: Proceedings of Conference on ACM SIGCOMM Computer Communication Review, 2005, pp. 229–240.
[57] B. Park, W.K. Hong, Y.J. Won, Toward fine-grained traffic classification, IEEE Commun. Magazine 49 (7) (2011) 104–111.
[58] J. Erman, A. Mahanti, M.F. Arlitt, Byte me: A case for byte accuracy in traffic classification, in: Proceedings of the 3rd Annual ACM Workshop on Mining Network Data, 2007, pp. 35–38.
[59] S. Dong, R. Jain, Flow online identification method for the encrypted skype, J. Netw. Comput. Appl. 132 (2019) 75–85.
[60] Andrew, W. Moore, Denis, Zuev, Internet traffic classification using bayesian analysis techniques, ACM SIGMETRICS Perform. Eval. Rev. (2005) 50–60.
[61] G. Sun, L. Liang, T. Chen, F. Xiao, F. Lang, Network traffic classification based on transfer learning, Comput. Electr. Eng. 69 (2018) 920–927.
[62] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, G. Zhang, Transfer learning using computational intelligence, Knowl.-Based Syst. 80 (2015) 14–23.
[63] W. Dai, Q. Yang, G. Xue, Y. Yu, Boosting for transfer learning, in: Proceedings of the 24th International Conference on Machine Learning, 2007, pp. 193–200.
[64] A. Fahad, Z. Tari, I. Khalil, I. Habib, H. Ainuweiri, Toward an efficient and scalable feature selection approach for internet traffic classification, Comput. Netw. 57 (9) (2013) 2040–2057.
[65] L. Yu, H. Liu, Feature selection for high-dimensional data: A fast correlation-based filter solution, in: Proceedings of the Machine Learning-International Workshop Conference, 2003.
[66] M. Dash, H. Liu, Consistency-based search in feature selection, Artificial Intelligence 151 (1–2) (2003) 155–176.
[67] G. Sun, T. Chen, Y. Su, C. Li, Internet traffic classification based on incremental support vector machines, Mob. Netw. Appl. 23 (4) (2018) 1–8.
[68] M.A. Lopez, D.M.F. Mattos, O.C.M.B. Duarte, G. Pujolle, A fast unsupervised preprocessing method for network monitoring, Ann. Telecommun. 74 (2018) 139–155.
[69] K. Kira, L. Rendell, The feature selection problem: traditional methods and a new algorithm, in: Proceedings of the National Conference on Artificial Intelligence, John Wiley and Sons Ltd, 1992, pp. 129–134.
[70] M. Shafiq, X. Yu, A.K. Bashir, H.N. Chaudhry, D. Wang, A machine learning approach for feature selection traffic classification using security analysis, J. Supercomput. 74 (2018) 4867–4892.
[71] Introduction to NetMate Tool, https://dan.arndt.ca/nims/calculating-flow-statistics-using-netmate/comment-page-1/ (Accessed 10 October 2020).
[72] G. Aceto, D. Ciuonzo, A. Montieri, A. Pescapé, Multi-classification approaches for classifying mobile app traffic, J. Netw. Comput. Appl. 103 (2018) 131–145.
[73] M. Conti, L.V. Mancini, R. Spolaor, N.V. Verde, Analyzing android encrypted network traffic to identify user actions, IEEE Trans. Inf. Forensics Secur. 11 (1) (2016) 114–125.
[74] V.F. Taylor, R. Spolaor, M. Conti, I. Martinovic, AppScanner: Automatic fingerprinting of smartphone apps from encrypted network traffic, in: Proceedings of IEEE European Symposium on Security and Privacy, 2016, pp. 439–454.
[75] M. Liberatore, B.N. Levine, Inferring the source of encrypted HTTP connections, in: Proceedings of the 13th Acm Conference on Computer and Communications Security, 2006, pp. 255–263.
[76] D. Herrmann, R. Wendolsky, H. Federrath, Website fingerprinting: Attacking popular privacy enhancing technologies with the multinomial Naïve-Bayes classifier, in: Proceedings of the ACM Workshop on Cloud Computing Security, 2009, pp. 31–42.
[77] T. Bakhshi, B. Ghita, On internet traffic classification: A two-phased machine learning approach, J. Comput. Netw. Commun. 2016 (2016).
[78] A. Dainotti, A. Pescapè, C. Sansone, Early classification of network traffic through multi-classification, in: Proceedings of the Third International Workshop on Traffic Monitoring and Analysis, 2011, pp. 122–135.
[79] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience, 2004.
[80] J. Cao, Z. Fang, G. Qu, H. Sun, D. Zhang, An accurate traffic classification model based on support vector machines, Int. J. Netw. Manag. 27 (1) (2017).
[81] D. Tong, Y.R. Qu, V.K. Prasanna, Accelerating decision tree based traffic classification on FPGA and multicore platforms, IEEE Trans. Parallel Distrib. Syst. 28 (2017) 3046–3059.
[82] Y.-s. Lim, H.-c. Kim, J. Jeong, C.-k. Kim, T.T. Kwon, Y. Choi, Internet traffic classification demystified: On the sources of the discriminative power, in: Proceedings of ACM Conference on Emerging Networking Experiments and Technology, 2010, pp. 1–12.
[83] Y.N. Dong, L.T. Yao, H.X. Shi, Fine grained classification of Internet video traffics, in: Proceedings of the 21st Asia-Pacific Conference on Communications, APCC, 2015.
[84] J.A. Jacko, A. Sears, M.S. Borella, The effect of network delay and media on user perceptions of web resources, Behav. Inform. Technol. 19 (6) (2000) 427–439.
[85] Y.N. Dong, J.J. Zhao, J. Jin, Novel feature selection and classification of internet video traffic based on a hierarchical scheme, Comput. Netw. 119 (2017) 102–111.
[86] R. Alshammari, A.N. Zincir-Heywood, Identification of VoIP encrypted traffic using a machine learning approach, Comput. Inform. Sci. 27 (2015) 77–92.
[87] G.M. Weiss, F. Provost, Learning when training data are costly: The effect of class distribution on tree induction, Artif. Intell. Res. 19 (2003) 315–354.
[88] Y. Wang, Y. Xiang, J. Zhang, Network traffic clustering using Random Forest proximities, in: Proceedings of IEEE International Conference on Communications, 2013, pp. 2058–2062.
[89] N.F. Huang, G.Y. Jai, H.C. Chao, Y.J. Tzang, H.Y. Chang, Application traffic classification at the early stage by characterizing application rounds, Inform. Sci. 232 (2013) 130–142.
[90] WEKA, http://www.cs.waikato.ac.nz/ml/weka/ (Accessed 10 October 2020).
[91] A. Fahad, Z. Tari, I. Khalil, A. Almalawi, A.Y. Zomaya, An optimal and stable feature selection approach for traffic classification based on multi-criterion fusion, Future Gener. Comput. Syst. 36 (2014) 156–169.
[92] S. Zander, T. Nguyen, G. Armitage, Sub-flow packet sampling for scalable ML classification of interactive traffic, in: Proceedings of the 37th Annual IEEE Conference on Local Computer Networks, 2013, pp. 68–75.
[93] M. Canini, D. Fay, D.J. Miller, A.W. Moore, R. Bolla, Per flow packet sampling for high-speed network monitoring, in: Proceedings of the First International Communication Systems and Networks and Workshops, 2009, pp. 1–10.
[94] N. Fukumoto, K. Nakamura, M. Suzuki, Y. Hiehata, M. Miyazawa, Framework and implementation of online smartphone traffic classification according to quality sensitivity, in: Proceedings of IEEE ComSoc International Communications Quality and Reliability Workshop, CQR, 2019, pp. 1–6.
[95] B. Wang, J. Zhang, Z. Zhang, L. Pan, Y. Xiang, D. Xia, Noise-resistant statistical traffic classification, IEEE Trans. Big Data 5 (4) (2019) 454–466.
[96] V. Soto, S. Garcia-Moratilla, G. Martinez-Munoz, D. Hernandez-Lobato, A. Suarez, A double pruning scheme for boosting ensembles, IEEE Trans. Cybern. 44 (12) (2014) 2682–2695.
[97] X. Wang, Y. Liu, W. Su, Real-time classification method of network traffic based on parallelized CNN, in: Proceedings of IEEE International Conference on Power, Intelligent Computing and Systems, ICPICS, 2019, pp. 92–97.
[98] F. Al-Obaidy, S. Momtahen, M.F. Hossain, F. Mohammadi, Encrypted traffic classification based ML for identifying different social media applications, in: Proceedings of IEEE Canadian Conference of Electrical and Computer Engineering, CCECE, 2019, pp. 1–5.
[99] S. Agrawal, J. Kaur, B.S. Sohi, Machine learning classifier for internet traffic from academic perspective, in: Int. Conf. Recent Adv. and Future Trends in Inf. Technol. RAFIT, 2012, pp. 4–9.
[100] L. Cehovin, Z. Bosnic, Empirical evaluation of feature selection methods in classification, J. Intel. Data anal. 14 (3) (2010) 265–281.
[101] M. AlSabah, K. Bauer, I. Goldberg, Enhancing Tor's performance using real-time traffic classification, in: Proceedings of the 2012 ACM Conference on Computer and Communications Security, CCS, 2012, pp. 73–84.
[102] R. Dingledine, N. Mathewson, P. Syverson, Tor: The Second-Generation Onion Router, Naval Research Lab Washington DC, 2004.
[103] J. Gama, Functional trees for classification, in: Proceedings of the 2001 IEEE International Conference on Data Mining, 2001, pp. 147–154.
[104] N. Landwehr, M. Hall, E. Frank, Logistic model trees, Mach. Learn. 59 (1–2) (2005) 161–205.
[105] V.A. Muliukha, L.U. Laboshin, A.A. Lukashin, N.V. Nashivochnikov, Analysis and classification of encrypted network traffic using machine learning, in: Proceedings of the 2020 International Conference on Soft Computing and Measurements, SCM 2020, pp. 194–197.
[106] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, K. Salamatian, Traffic classification on the fly, in: Proceedings of ACM SIGCOMM Computer Communication Review, 2006, pp. 23–26.
[107] M.E. Ahmed, S. Ullah, H. Kim, Statistical application fingerprinting for DDoS attack mitigation, IEEE Trans. Inf. Forensics Secur. 14 (6) (2019) 1471–1484.

[108] J. Zhang, Y. Xiang, W. Zhou, Y. Wang, Unsupervised traffic classification using flow statistical properties and IP packet payload, J. Comput. System Sci. 79 (5) (2013) 573–585.
[109] L. Grimaudo, M. Mellia, E. Baralis, R. Keralapura, Self-learning classifier for Internet traffic, in: Proceedings of IEEE INFOCOM, 2013, pp. 3381–3386.
[110] E. Mahdavi, A. Fanian, H. Hassannejad, Encrypted traffic classification using statistical features, ISeCure 10 (1) (2018) 29–43.
[111] SSH, http://www.rfcarchive.org/getrfc.php?rfc=4251, (Accessed 10 October 2020).
[112] A. Vlăduţsu, D. Comăneci, C. Dobre, Internet traffic classification based on flows' statistical properties with machine learning, Int. J. Netw. Manag. 27 (3) (2016).
[113] Ixia BreakingPoint, http://www.ixiacom.com/products/ixia-breakingpoint (Accessed 10 October 2020).
[114] J. Ran, X. Kong, G. Lin, D. Yuan, H. Hu, A self-adaptive network traffic classification system with unknown flow detection, in: Proceedings of the 3rd IEEE International Conference on Computer and Communications, ICCC, 2017, pp. 1215–1220.
[115] H. Shi, H. Li, D. Zhang, C. Cheng, W. Wu, Efficient and robust feature extraction and selection for traffic classification, Comput. Netw. 119 (2017) 1–16.
[116] Y. Zhang, S. Zhao, J. Zhang, X. Ma, F. Huang, STNN: A novel TLS/SSL encrypted traffic classification system based on stereo transform neural network, in: Proceedings of IEEE 25th International Conference on Parallel and Distributed Systems, ICPADS, 2019, pp. 907–910.
[117] J. Zhang, C. Chen, Y. Xiang, W. Zhou, Y. Xiang, Internet traffic classification by aggregating correlated naive Bayes predictions, IEEE Trans. Inf. Forensics Secur. 8 (1) (2013) 5–15.
[118] J. Zhang, Y. Xiang, Y. Wang, W. Zhou, Y. Xiang, Y. Guan, Network traffic classification using correlation information, IEEE Trans. Parallel Distrib. Syst. 24 (1) (2013) 104–117.
[119] D.M. Divakaran, L. Su, Y.S. Liau, V.L. L. Thing, SLIC: Self-learning intelligent classifier for network traffic, Comput. Netw. 91 (2015) 283–297.
[120] L. Ding, J. Liu, T. Qin, H. Li, Internet traffic classification based on expanding vector of flow, Comput. Netw. 129 (2017) 178–192.
[121] Y. Wang, Y. Xiang, J. Zhang, W. Zhou, G. Wei, L.T. Yang, Internet traffic classification using constrained clustering, IEEE Trans. Parallel Distrib. Syst. 25 (11) (2014) 2932–2943.
[122] Y. Wang, Y. Xiang, J. Zhang, S.Z. Yu, A novel semi-supervised approach for network traffic clustering, in: Proceedings of the 5th International Conference on Network and System Security, 2011, pp. 169–175.
[123] M. Canini, W. Li, M. Zadnik, A.W. Moore, Experience with high-speed automated application-identification for network-management, in: Proceedings of the 5th ACM/IEEE Symposium on Architectures for Networking and Communications Systems, 2009, pp. 209–218.
[124] J. Zhang, C. Chen, Y. Xiang, W. Zhou, An effective network traffic classification method with unknown flow detection, IEEE Trans. Netw. Serv. Manag. 10 (2) (2013) 133–147.
[125] J. Zhang, X. Chen, Y. Xiang, W. Zhou, J. Wu, Robust network traffic classification, IEEE/ACM Trans. Netw. 23 (4) (2015) 1257–1270.
[126] T.v. Ede, R. Bortolameotti, A. Continella, J. Ren, D.J. Dubois, M. Lindorfer, D. Choffnes, M. van Steen, A. Peter, FLOWPRINT: Semi-supervised mobile-app fingerprinting on encrypted network traffic, in: Proceedings of Network and Distributed System Security Symposium, NDSS, 2020.
[127] J. Ren, M. Lindorfer, D.J. Dubois, A. Rao, N. Vallina-Rodriguez, Bug fixes, improvements, ... and privacy leaks - A longitudinal study of PII leaks across Android App Versions, in: Proceedings of the Network and Distributed System Security Symposium, NDSS, 2018.
[128] J. Ren, A. Rao, M. Lindorfer, A. Legout, D. Choffnes, ReCon: Revealing and controlling PII leaks in mobile network traffic, in: Proceedings of the International Conference on Mobile Systems, Applications and Services, MobiSys, 2016.
[129] J. Ren, D.J. Dubois, D. Choffnes, An International View of Privacy Risks for Mobile Apps, 2019.
[130] M. Lindorfer, M. Neugschwandtner, L. Weichselbaum, Y. Fratantonio, V.V.D. Veen, C.P. Andrubis, ANDRUBIS–1, 000, 000 Apps Later: A view on current android malware behaviors, in: Proceedings of 2014 Third International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security, BADGERS, 2014, pp. 3–17.
[131] L. Grimaudo, M. Mellia, E. Baralis, Hierarchical learning for fine grained internet traffic classification, in: Proceedings of 8th International Wireless Communications and Mobile Computing Conference, IWCMC, 2012, pp. 463–468.
[132] Kohout, Jan, Komarek, Tornag, Cchc, Premysl, Bodnar, Lokoc, Jakub, Learning communication patterns for malware discovery in HTTPs data, Expert Syst. Appl. 101 (2018) 129–142.
[133] M. Iliofotou, H.C. Kim, M. Faloutsos, M. Mitzenmacher, G. Varghese, Graph-based P2P traffic classification at the internet backbone, in: Proceedings of IEEE International Conference on Computer Communications Workshops, 2009, pp. 1–6.
[134] B. Qu, Z. Zhang, X. Zhu, D. Meng, An empirical study of morphing on behavior-based network traffic classification, Secur. Commun. Netw. 8 (1) (2015) 68–79.
[135] M.J. Choi, J.S. Park, M.S. Kim, An integrated method for application-level internet traffic classification, KSII Trans. Internet Inform. Syst. 8 (3) (2014) 838–856.
[136] E. Glatz, X. Dimitropoulos, Classifying internet oneway traffic, in: Proceedings of Acm Conference on Internet Measurement Conference, 2012, pp. 37–50.
[137] J. Kampeas, A. Cohen, O. Gurewitz, Traffic classification based on zero-length packets, IEEE Trans. Netw. Serv. Manag. 15 (2018) 1049–1062.
[138] Z. Liu, R. Wang, D. Tang, Extending labeled mobile network traffic data by three levels traffic identification fusion, Future Gener. Comput. Syst. 88 (2018) 453–466.
[139] P. Casas, P. Fiadino, A. Bar, IP mining: Extracting knowledge from the dynamics of the Internet addressing space, in: Proceedings of the 25th International Teletraffic Congress, ITC, 2013, pp. 1–9.
[140] Y. Wang, Y. Xiang, S. Yu, Internet traffic classification using machine learning: A token-based approach, in: Proceedings of the 14th IEEE International Conference on Computational Science and Engineering, 2011, pp. 285–289.
[141] A. Finamore, M. Mellia, M. Meo, D. Rossi, KISS: Stochastic packet inspection classifier for UDP traffic, IEEE/ACM Trans. Netw. 18 (5) (2010) 1505–1515.
[142] J. Yang, J. Narantuya, H. Lim, Bayesian neural network based encrypted traffic classification using initial handshake packets, in: Proceedings of the 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks–Supplemental Volume (DSN-S), 2019, pp. 19–20.
[143] L. Robert, SSL/TLS cipher suite analysis and strong cipher enablement, Symantec (2014) 3–22.
[144] H. Netze, Transport Layer Security (TLS) Extensions: Extension Definitions, 2011.
[145] B.C. Park, Y.J. Won, M.S. Kim, J.W. Hong, Towards automated application signature generation for traffic identification, in: Proceedings of Network Operations and Management Symposium, 2008, pp. 160–167.
[146] T. Ishioka, Extended K-means with an efficient estimation of the number of clusters, 1983 (2000) 17–22.
[147] S. Mayank, H. Neminath, Rdclass: On using relative distance of keywords for accurate network traffic classification, IET Netw. 7 (4) (2018) 273–279.
[148] P. Khandait, N. Hubballi, B. Mazumdar, Efficient keyword matching for deep packet inspection based network traffic classification, in: Proceedings of International Conference on Communication Systems and Networks, COMSNETS, 2020, pp. 567–570.
[149] G. Marín, P. Casas, G. Capdehourat, Deep in the dark-deep learning-based malware traffic detection without expert knowledge, in: Proceedings of IEEE Security and Privacy Workshops, SPW, 2019, pp. 36–42.
[150] S. Garcia, M. Grill, J. Stiborek, A. Zunino, An empirical comparison of botnet detection methods, Comput. Secur. 45 (2014) 100–123.
[151] Y.D. Lin, C.N. Lu, Y.C. Lai, W.H. Peng, P.C. Lin, Application classification using packet size distribution and port association, J. Netw. Comput. Appl. 32 (5) (2009) 1023–1030.
[152] B. Anderson, D. McGrew, 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, 2017, pp. 1723–1732.
[153] G. Szabó, D. Orincsay, S. Malomsoky, I. Szabó, On the validation of traffic classification algorithms, in: Proceedings of International Conference on Passive and Active Network Measurement, 2008, pp. 72–81.
[154] M. Canini, W. Li, A.W. Moore, R. Bolla, GTVS: Boosting the Collection of Application Traffic Ground Truth, Springer Berlin Heidelberg, 2009.
[155] M. Rosario Oliveira, J. Neves, R. Valadas, P. Salvador, Do we need a perfect ground-truth for benchmarking Internet traffic classifiers? in: Proceedings of IEEE Conference on Computer Communications, INFOCOM, 2015, pp. 2452–2460.
[156] Y. Fu, H. Chen, Q. Zheng, Z. Yan, R. Kantola, J. Jing, H. Li, An adaptive security data collection and composition recognition method for security measurement over LTE/LTE-A networks, J. Netw. Comput. Appl. 155 (2020).
[157] R. Bost, R.A. Popa, S. Tu, S. Goldwasser, Machine learning classification over encrypted data, in: Proceedings of Network and Distributed System Security Symposium, NDSS, 2015, pp. 4325.
[158] Q. Zheng, J. Liu, H. Zeng, Z. Guo, W. Bei, B. Wei, Knowledge forest: a novel model to organize knowledge fragments, Sci. China (Inform. Sci.) (2019).
[159] Y. Chen, Q. Zheng, W. Zhang, Omni-word feature and soft constraint for Chinese relation extraction, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2014, pp. 572–581.
[160] X. Jing, Z. Yan, Y. Shen, W. Pedrycz, J. Yang, A group-based distance learning method for semisupervised fuzzy clustering, IEEE Trans. Cybern. (2020) 1–14.