Research on QoS Classification of Network Encrypted Traffic Behavior Based on Machine Learning

Yung-Fa Huang 1, Chuan-Bi Lin 1, Chien-Min Chung 1 and Ching-Mu Chen 2, *

1 Department of Information and Communication Engineering, Chaoyang University of Technology, Taichung 413310, Taiwan; yfhuang@cyut.edu.tw (Y.-F.H.); cblin@cyut.edu.tw (C.-B.L.); s10730612@gm.cyut.edu.tw (C.-M.C.)
2 School of Engineering, Chung Chou University of Science and Technology, Changhua County 51003, Taiwan
* Correspondence: 925633@gmail.com; Tel.: +886-4-835-90-00

Abstract: In recent years, privacy awareness is concerned due to many Internet services have chosen to use encrypted agreements. In order to improve the quality of service (QoS), the network encrypted traffic behaviors are classified based on machine learning discussed in this paper. However, the traditional traffic classification methods, such as IP/ASN (Autonomous System Number) analysis, Port-based and deep packet inspection, etc., can classify traffic behavior, but cannot effectively handle encrypted traffic. Thus, this paper proposed a hybrid traffic classification (HTC) method based on machine learning and combined with IP/ASN analysis with deep packet inspection. Moreover, the majority voting method was also used to quickly classify different QoS traffic accurately. Experimental results show that the proposed HTC method can effectively classify different encrypted traffic. The classification accuracy can be further improved by 10% with majority voting as K = 13. Especially when the networking data are using the same protocol, the proposed HTC can effectively classify the traffic data with different behaviors with the differentiated services code point (DSCP) mark.

Keywords: quality of service; machine learning; differentiated service; traffic classification

1. Introduction

With the vigorous development of communication technology and the Internet, All-IP networks have become mainstream [1], and many services built on IP networks have been born as a result, such as OTT [2], VoIP [3], etc. Therefore, how to provide better quality of service to users under limited network bandwidth and the needs of different applications is necessary to provide different QoS (Quality of Service) for different needs at the edge of the network [4] for different applications. Moreover, the QoS at the edge of the network necessary that is able to identify traffic has become a major challenge for Internet service providers (ISPs).

There are three commonly used traffic classification methods now based on packet header (Header) judgment, deep packet inspection (Deep Packet Inspection, DPI) [5], and machine learning (ML) methods [6]. The method based on Header judgment is divided into the IP and Autonomous System Number (ASN) detection method that uses the server’s IP to obtain registrant information, and the stateful firewall recognized the port and connection status. The advantage to the identification speed is faster, but the accuracy is poor. The IP registrant information or the application corresponding to the port number must be known in advance during analysis. The advantage of using DPI is more accurate, but it takes more time to parse the packet content, so the comparison time is longer and the identification rules are more complicated [7]. However, the training data is pre-processed before training and the training process needs more computing resources.

Furthermore, the awareness of information security and privacy has been increased in recent years. Users transmit data with data encryption [8]. Using the method based on
packet header judgment and DPI will encounter data encryption as detecting traffic and the data must be decrypted through an intermediary or a shared key to perform traffic identification [9]. However, the machine learning method has been proved to be effective in processing and identifying encrypted traffic [10–22].

This paper proposes a hybrid traffic classification (HTC) that uses a machine learning method to determine and classify the traffic behavior of users to reduce the types and targets of classification, and maintain the number of tags for training preprocessing. It combines IP/ASN and deep packet inspection to supplement the classification results of the machine learning model, and adds a Majority Voting strategy to get more accurate traffic classification results. In addition, this paper also attempts to obtain different degrees of packet header information as identification features during the training process of the machine learning model to observe the impact on the classification effect, and simulate the actual traffic to verify the classification accuracy.

Quality of service (QoS) is a mechanism to control network resource allocation [4] that provides corresponding priority for different users or applications, and achieve the guaranteed network performance. Network performance can be evaluated by the transmission bandwidth, delay and jitter of the network. Different types of applications have their own performance requirements. The transmission mode of the original IP network design is Best Effort (BE) mode. When network resources are limited, each user and application must grab resources caused unfair network resource allocation. Some network users or applications occupies many resources that will affect other services. Therefore, with limited resources, QoS is needed to distinguish applications with different requirements or guarantee their resources.

In order to meet the needs of different applications in the network, the IETF (Internet Engineering Task Force) has customized two different QoS standards [11], namely Integrated Services (IntServ) [12] and Differential Services (Differentiated Services, Diff-Serv) [13], the following describes the two QoS methods and best-effort transmission of IP networks without QoS.

The main operation mode of the Integrated Service [12] is to ensure the QoS of network transmission by pre-reserving resources. Its mode of operation is shown in Figure 1. When users want to send traffic, they must first request resource reservation through the Resource Reservation Protocol (RSVP) [14].

![Figure 1. IntServ Operational Architecture.](image-url)

The sender first sends the RSVP path (PATH) message to the receiver, that advertises each node or router on the path, records and establishes the path status along the way, and reserves resources. The reservation message (RESV) will be sent back to the sender
by the receiver along the opposite path and the required resources are determined. When sender receives the RESV message, the path is established to confirm that the resource reservation on the entire path is completed, and establishes a virtual circuit between nodes, as shown in Figure 2. Packet streams with the same source and destination IP and port are transmitted through the virtual channel with guaranteed quality of service.

![Virtual channel established by RSVP.](image)

Compared with the overall service, the design of differentiated service [13] is simpler in implementation. Mainly through the implementation of traffic classification and admission control at the edge of the network, it provides a different method from IntServ to ensure the quality of the network service. Its architecture is shown in Figure 3a. The nodes on the entire DiffServ network will be composed of two roles that are Edge Router and Core Router. When the packet enters the network, Edge Router will classify the packet and adjust the flow rate. The operation process is shown in Figure 3b.

![DiffServ operation architecture (a) Architecture diagram (b) Operation process.](image)

In order to provide QoS, DiffServ must classify traffic and add DSCP tags. In order not to affect the format of IP packets, DiffServ writes DSCP tags in the ToS (type of service) field of IP packets. The ToS field is planned in the IP network to distinguish the priority of packets. Due to the insufficient definition of ToS, it will be replaced by the DSCP field in RFC2474, as shown in Figure 4. The DSCP field has a total length of 6 bits that can be classified into 26 and a total amount of 64 categories. The remaining 2 bits are used as ECN (Explicit Congestion Notification), which has nothing to do with DSCP.

![DSCP field format.](image)
Assured Forwarding (AF) guarantees the traffic not exceed a certain rate that can be subdivided into 12 categories according to the priority. The acceptable degree of packet drops [15], where the first three bits represent the priority, the second and three bits indicate the acceptance level of packet dropping, as shown in Table 1. The Expedited Forwarding (EF) is the highest priority, as far as possible to reduce the delay and jitter of the network. In addition, a ToS-compatible IP Precedence (IPP) Class Selector (CS) [16] was ordered, the last three bits are fixed to 0, and the first three bits are the same as IPP, as shown in Table 2. The use of DSCP classification can refer to Cisco’s QoS Baseline recommendations [17], as shown in Table 3.

Table 1. DSCP AF category.

| Acceptance | Priority | Class1 (001) | Class2 (010) | Class3 (011) | Class4 (100) |
|------------|----------|--------------|--------------|--------------|--------------|
| Low (010)  | AF11     | AF21         | AF31         | AF41         |
| Medium (100)| AF12     | AF22         | AF32         | AF42         |
| High (110) | AF13     | AF23         | AF33         | AF43         |

Table 2. DSCP CS category.

| Category   | DSCPValue | IP Precedence |
|------------|-----------|---------------|
| CS0/DF     | 0 (000 000)_b | Routine |
| CS1        | 8 (001 000)_b | Priority |
| CS2        | 16 (010 000)_b | Immediate |
| CS3        | 24 (011 000)_b | Flash |
| CS4        | 32 (100 000)_b | Flash Override |
| CS5        | 40 (101 000)_b | Critical |
| CS6        | 48 (110 000)_b | Internetwork Control |
| CS7        | 56 (111 000)_b | Network Control |

Table 3. Cisco’s QoS Baseline recommendations.

| DSCP Category | Application Area |
|---------------|------------------|
| DF/CS0        | Best Effort      |
| CS1           | Scavenger        |
| AF11          | Bulk Data        |
| CS2           | Network Management |
| AF21          | Transactional Data |
| CS3           | Call-Signaling |
| AF31          | Mission-Critical Data |
| CS4           | Streaming-Video |
| AF41          | Interactive Video |
| EF (ID:46)    | Voice            |
| CS6           | IP Routing       |
The classification of transmission ports is through the well-known port numbers of TCP and UDP [18] to identify possible network services on the application layer. The well-known ports on TCP and UDP are allocated by IANA, and its common services are shown in Table 4.

Table 4. Common services.

| Port No. | Application Protocol | Transmission Protocol |
|----------|----------------------|-----------------------|
| 20       | FTP Data             | TCP                   |
| 21       | FTP Control          | TCP                   |
| 22       | SSH                  | TCP                   |
| 23       | Telnet               | TCP                   |
| 53       | DNS                  | TCP/UDP               |
| 67, 68   | DHCP                 | UDP                   |
| 80       | HTTP                 | TCP                   |
| 143      | IMAP                 | TCP                   |
| 443      | HTTPS                | TCP                   |

DPI is a technology used to analyze network packets. Compared with the method of judging the type of traffic only by the port number, DPI has better accuracy. The detection method is to mine the payload data through regular expressions or string search, such as the Aho-Corasick [19] algorithm, etc., to compare the characteristic string or data state in the network packet. The following is to use Wireshark [20] with the open source DPI project nDPI [21] to analyze the packet record file. It can be observed that the application layer protocol can be resolved as the RTP protocol when DPI is used. However, for traffic using encryption protocols, because the data in the packet are encrypted, DPI cannot directly decrypt the data, resulting in inaccurate interpretation. Data must be decrypted through an intermediary or a shared key [8] in order to perform traffic identification that poses hidden dangers to data privacy and security.

Machine learning provides a large amount of training data to the computer, allowing the computer to learn by simulating the learning method of the human brain [5]. The model can calculate the loss value (loss) between the predicted result and the target result through the optimization algorithm and Loss Function. It updates the weight and weight of the entire model through the backpropagation algorithm (Backpropagation, BP) [22] Error value, and can gradually accumulate learning and improve accuracy. Therefore, the machine learning model can mine and count the features in a large number of packet data, and use the features to achieve accurate traffic identification and classification.

For encrypted traffic, References [10,23] showed that machine learning can accurately identify traffic especially in the Datanet proposed by Reference [23]. Three models of multi-layer perceptron (MLP), Stacked Auto Encoder (SAE) [24] and convolutional neural network (CNN) [25] are proposed for traffic identification, and use the ISCX VPN-nonVPN [26] public data set for training. Experimental results show that machine learning methods can indeed effectively identify and classify traffic.

In terms of accuracy, CNN and SAE models can achieve the best recognition results, and the accuracy of individual classifications can reach more than 90% on average. The worst effect is the MLP model that the classification accuracy can reach more than 80%.

When a different flow rate of the training data set is used for verification, the accuracy will be greatly reduced. Machine learning allows the machine to absorb experience and find rules in the learning process, and use the model to predict or classify [5].

According to different training methods, machine learning can be divided into four categories: supervised learning [27], unsupervised learning [28], semi-supervised learning [29], and reinforcement learning [30].
The paper revealed that the machine learning method used is based on the data flow of orderly traffic over a period of time to identify and filter inactive data (3 Kbyte threshold) [31]. In order to be able to distinguish the type of a single packet in a short time, the MLP method to identify each packet, so the accuracy will be affected by the following types of packets in the traffic, such as ACK, SYN, and RST, for empty packets or packets with a small amount of data. However, the use of majority voting method proposed in this paper can improve this issue, and the accuracy was improved by 10%. The multilayer perceptron (MLP) is a forward-passing machine learning architecture. It must be composed of more than three layers of perceptions, which includes input, output, and at least one hidden layer, as shown in Figure 5.

![Multilayer perceptron MLP architecture.](image)

The Stacked AutoEncoder (SAE) [23] was improved from AutoEncoder (AE) [32]. The architecture of the autoencoder is shown in Figure 6. The number of nodes between the input layer and the output layer is the same, and the number of nodes in the hidden layer is less, causing the data to lose part of the data due to compression when passing through the hidden layer. The training target of AE is the input target, so the compressed data will be restored to the input state in a way. Through this training method, the final hidden layer can retain the important features of the data. After that, the trained AE hidden layer is taken out as the input layer of another AE and trained in the same way. Through this hierarchical training method, multiple hidden layers that can obtain data features can be obtained. Finally, the multiple completed training AEs are hidden into a combination and then formed together to form an SAE, as shown in Figure 7. Finally, the training is carried out by using the label data. Since the hidden layer is trained for the first time by AE, the initial weight was adjusted to a suitable state, so a very good classifier can be obtained in the end.
Convolutional Neural Network (CNN) [24], as shown in Figure 8. It is composed of Convolution Layer, Pooling Layer and Fully-Connected Layer. The input method of CNN is different from the general neural network. It uses a set of two-dimensional matrices as input. After a series of convolutional and pooling operations, local features are extracted and retained in the original two-dimensional data to reduce matrix and the computational complexity of the data. Finally, the data obtained after the feature are flattened and transformed into one-dimensional data. Then input of the fully connected layer for learning, and the fully connected layer is usually composed of MLP and other networks.
The main function of the activation function is to introduce non-linearity between the output of the upper layer and the input of the lower layer, so that it can be separated from the linear relationship, and the learning can achieve better results. The commonly used activation functions in neural networks are Sigmoid, Hyperbolic Tangent (TanH), and Rectified Linear Unit (ReLU) [33,34]. Among them, ReLU is the most widely used excitation function in machine learning. It is used to prevent the problem of gradient disappearance, and has a small amount of calculation and does not require complex calculations. Its function is mainly to adjust the value less than 0 to 0, which can be expressed as follows:

\[ f(x) = \max(0, x); \quad x \in \mathbb{R} \] (1)

In the machine learning model for classification, the output layer usually chooses to use the Softmax function. Softmax normalizes the values between all multiple nodes to [0, 1], and the sum is equal to 1, which can show the probability of each classification result, and can implement multi-category classification [31]. The probability that the sample \( z_i \) belongs to the category \( i \) can be defined by Softmax, expressed as follows:

\[ q(z)_i = \frac{e^{z_i}}{\sum_{c=1}^{C} e^{z_c}}, \quad i = 1, \ldots, C \] (2)

where \( z \) is the input of the previous layer, and \( C \) is the total number of categories.

The optimization algorithm is the method used by machine learning to find the best solution. The most well-known one is Gradient Descent, as shown in Figure 9. It can find the local optimal solution to the location with a small loss through a step-by-step search method. The gradient descent formula is defined as follows:

\[ W^{t+1} = W^t - \eta \frac{\partial L}{\partial W} \] (3)

where \( W \) is the weight, \( L \) is the loss value, and \( \eta \) is the learning rate. Allow the model to find the updated weight value \( W^{(t+1)} \) to the lower loss value.
Loss Function is a machine learning model used to estimate the gap with the target value. The smaller the calculated loss value, the closer the distance to the target value and the higher the accuracy. In the classification problem, cross entropy is usually used to calculate the loss value, and the cross entropy can be expressed as follows:

\[ CE(p, q) = - \sum_{i=1}^{C} p_i \log_2(q_i) \]  

(4)

where \( p_i \) is the expected probability corresponding to the \( i \)-th category in the label \( p \), and \( q_i \) is the probability that the model predicts the \( i \)-th category. However, during the training process of the model, the training speed is usually accelerated through batch training. Therefore, the final loss value \( L \) is the mean value of the cross entropy of all test samples in the batch, which can be expressed as follows:

\[ L = \frac{1}{B} \sum_{b=1}^{B} CE(p_{b}, q_{b}) \]  

(5)

where \( B \) is the batch size.

As shown in Figure 10, the relationship between the predicted class and the actual class can be clearly seen by the confusion matrix. When the prediction is the same as the actual classification, that is, the classification result is True, otherwise it is False. Therefore, there are four conditions, True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN).
Commonly used indicators are accuracy, precision, recall and F1-Measure or F1-Score [33]. The closer the value is to 1, the better the model performance. Accuracy $A$ is used to evaluate the proportion of correct predictions of True and False under all conditions, which can be expressed as follows:

$$A = \frac{TP + TN}{TP + TN + FP + FN}$$  \hspace{1cm} (6)

The accuracy rate $P$ is used to evaluate how many of the cases predicted to be positive ($TP$) are true positives ($TP$), which can be expressed as follows:

$$P = \frac{TP}{TP + FP}$$  \hspace{1cm} (7)

The recall rate, $R$, is used to evaluate how many of all positive categories are correctly predicted as true positives ($TP$), which can be expressed as follows:

$$R = \frac{TP}{TP + FN}$$  \hspace{1cm} (8)

F1-Score $F_1$ is the harmonic average of precision rate and recall rate. It can comprehensively consider the two values to avoid extreme conditions. It can be expressed as follows:

$$F_1 = \frac{2 \times P \times R}{P + R}$$  \hspace{1cm} (9)

Suppose that if the model predicts all categories as positive, the recall rate will be 1, but the accuracy may be very low (for example, close to 0). Generally, the average calculation result is 0.5, while $F_1$-Score is 0.

2. Materials and Methods

The architecture of the traffic classification system is shown in Figure 11. The entire network is composed of Mounting Client, Mark Forward Server and Edge Router. The main purpose is to make traffic through the use of Mark Forward Server. Using the hybrid traffic classification method distinguishes the QoS category of the traffic sent by the network client according to the usage behavior and mark and forward the traffic to realize the Classifier, Meter and Marker functions in DiffServ. It enables Edge Router to implement network QoS through the packet scheduling corresponding to the DSCP label, and forward the flow to the DiffServ network domain.
When the traffic passes through the Mark Forward Server, the mixed traffic classification will be performed through the flowchart as shown in Figure 12. First, the system will treat packets with the same source destination IP and port as the same flow. When the packet comes in, it will first confirm whether the flow has been classified, find the classification result, directly mark the corresponding DSCP value; and forward the flow. When it is found that the traffic has not been classified, it will confirm whether it is a new traffic, and judge whether there has been a preliminary DPI test. If the traffic is new, DPI will be used for preliminary judgment, and IP registration information query will be completed.

Figure 11. Architecture of traffic classification system.

Figure 12. Hybrid traffic classification process.
After the DPI detection is completed, if the result is not encrypted or unknown traffic, the classification detection result will be recorded. When the same data flow enters again, the classification results can be immediately queried, and the DSCP value is forwardly marked. However, when the data are encrypted or there is unknown traffic, the P-MLP model is used for traffic classification, and corresponding data preprocessing and classification are performed. After the classification is completed, the classification result will be recorded. It confirms whether or not the classification has been completed with a certain number of times that is the classification has been completed $K$ times. When the number of classifications reaches $K$, the majority voting method is used for calculation according to the classification results. After counting the most categories in multiple classifications, it determines the classification of the data flow and record the classification detection results. If it is still not achieved, the preset DSCP is used for classification results as the output.

Network traffic should be classified in units of traffic during detection, and it will be pre-communication operations when the connection is established, such as TCP 3-Way handshake. This type of packet is usually empty or carries data with no identifying characteristics. If only the first packet of each traffic is classified by machine learning, it may cause lower accuracy. Therefore, this paper adopts the majority voting method that records a traffic $K$ classification results and votes. It uses the classification results of $K$ packets as statistics to reduce the occurrence of identification errors and improve the classification accuracy. The packet data used for majority voting is in order and the traffic is collected from the beginning of the connection. During the preprocessing, the order of the packets will not be disrupted.

In order to compare the influence of the $K$ value on the results of the majority voting method, five different results, namely 1 (no majority voting), 5, 9, 13 and 17 were tested. At the same time, in order to verify the influence of the empty packets generated by TCP in the 3-way handshake on the results of the majority voting method, the voting method not excluded and excluded the first three packets was used for comparison. It can be observed from Figure 13. However, the recall rate of the streaming category is low. When $K$ increases, the accuracy of the streaming category is shown in Figure 13a and the recall rate in Figure 13b is slightly improved, while the F1-Score rises to 0.6 for the row traffic classification results.

The majority vote is performed by excluding the first three packets as shown in Figure 14. With the case of handshake packets, the classification accuracy can be increased by about 10%. The F1-Score score can reach as high as 0.7. Therefore, the method of removing the first three packets can eliminate the problem of insufficient classification characteristics of lacking payload in the packet when the TCP connection is established, and finally to improve the accuracy of majority voting.

Figure 15 shows the Packet MLP (P-MLP) model. The P-MLP is composed of two MLP networks, which process payload and header information separately, and finally aggregate and use Softmax to output, and use the highest probability classification as the output result. The MLP network that processes packet data has an input layer of $N$ nodes and 3 hidden layers. The value of $N$ needs to be defined according to the maximum length of a single TCP/UDP packet data. Therefore, the value of $N$ is set to 1472 according to the maximum length of a single UDP packet data under the Maximum Transmission Unit (MTU) in common Ethernet. The MLP network that processes the header has an input layer of 12 nodes and a hidden layer. ReLU is set as the excitation function between the hidden layers. P-MLP divides traffic into four categories: bulk data, browsing, streaming and real-time based on usage behavior.
Figure 13. Results of majority voting method (a) accuracy of each category, (b) recall rate of each category, (c) F1-Score of each category, and (d) average of each category index.

Figure 14. The results of majority voting after removing the first three packets (a) accuracy of each category, (b) recall rate of each category, (c) F1-Score of each category, and (d) Average of each category index.
Figure 15. P-MLP model architecture.

The training data set used is obtained from the ISCX VPN/non-VPN [25] data set by selecting FTPS, SCP and Email program packets and Wireshark snippet packets, as shown in Table 5.

Table 5. Training data set and assigned labels.

| ISCX VPN-NonVPN   | Quantity | Category       |
|-------------------|----------|----------------|
| FTPS              | 6000     | Bulk Data      |
| SCP Download      | 4500     | Browsing       |
| E-Mail            | 1750     |                |

| Wireshark         | Quantity | Category       |
|-------------------|----------|----------------|
| Discord Text      | 1750     | Browsing       |
| Line Text         | 1750     |                |
| Messenger Text    | 1750     |                |
| Facebook          | 1750     |                |
| Google Search     | 1750     |                |
| Spotify           | 2100     | Streaming      |
| Apple Music       | 2100     |                |
| YouTube           | 2100     |                |
| Facebook Video    | 2100     |                |
| Netflix           | 2100     |                |
| Discord Screen Share/Video | 1750 | Real-Time |
| Discord Voice     | 1750     |                |
| Line Voice        | 1750     |                |
| Line Video        | 1750     |                |
| Messenger Voice   | 1750     |                |
| Messenger Video   | 1750     |                |
In total, 80% are used as training data sets, and 20% are used as test data sets. In each round (Epochs) training, 80% of the training data set is randomly selected as the training data for this round, and 20% is used as the verification data for this round. To ensure that the test data set will not participate in the training process and affect the accuracy of the P-MLP model. Each round of training will go through multiple iterations (Iterate), each iteration will sequentially take out a fixed batch of training data, and do back propagation to update the weights, until all training data sets are taken out and training is completed. The training process of the P-MLP model adopts supervised learning, and the data are classified into 4 categories to distinguish the demand traffic behaviors of different QoS. The four types of tags are real-time, streaming, browsing and bulk data from high priority to low priority. The real-time tag is the highest priority traffic, corresponding to real-time multimedia streaming, voice calls, etc., for applications that require high network bandwidth and delay; and that are jitter-sensitive applications.

Streaming is a multimedia streaming service, with the second priority. Although multimedia streaming requires a higher bandwidth, it has a buffer mechanism in non-real-time multimedia streaming, which can accept problems such as sudden packet drop and bandwidth reduction to a limited extent. Browsing corresponds to services such as web browsing and e-mail, which are generally less demanding on the Internet. The last is bulk data, which corresponds to file transfer service. The corresponding label category and DSCP category are shown in Table 6. Under the QoS Baseline architecture proposed by Cisco [17], the priority of bulk data should be higher than that of web browsing. However, when considering the flow of large-volume file transfer services but not restricting it, it is easy to cause network resource bottlenecks and affect other networks. Moreover, during file transfer, it is more acceptable to users if there is a sudden decrease in the transfer speed. However, when the web service cannot be browsed normally, the user experience is poor.

Table 6. Classification category and DSCP tag correspondence.

| Classification Label | DSCP |
|----------------------|------|
| Bulk Data            | BE   |
| Browsing             | AF11 |
| Streaming            | CS4  |
| Real-Time            | EF, AF |

The original packet data cannot be directly input into the P-MLP model for classification. The packet data needs to be preprocessed first, as shown in Figure 16. The preprocessing will take the packet header and payload separately. The IP proto field, TCP/UDP port, and TCP flags information will be extracted from the header as the data for identifying the traffic type of the transport layer. In addition, the TCP/UDP port will determine the source port or the destination port according to the traffic direction. In addition, determine whether the length of the payload data is greater than N Bytes, and decide to truncate or padded to N Bytes. After obtaining the data, normalize the Byte of the data, limit the Byte value to [0, 1], speed up the model training speed, and input the data into the P-MLP model in the number of Bytes.
Before actual training, effective features must be selected from the packet data for model training and use. The network packet is completed through multi-layer encapsulation. The application-layer data will be encapsulated in sequence from top to bottom through the transport layer, network layer and data link layer, and finally become a transmittable frame. However, in the encrypted data, since the original data were encrypted at the application layer, it is impossible to manually find the characteristics in the data. Therefore, the payload data in the packet can only be obtained by the model for self-learning. However, in the process of each layer of encapsulation, the header is added to the data, and the information on the header is not encrypted, so the header information can be selected as features for the model to use, so that the model can obtain more classification features. Therefore, some fields will be selected in the header of the transport layer and the network layer as the features of P-MLP learning.

In the network layer, the available features are the protocol field and the address field. The protocol field will record the protocol used by the transport layer. The address field is divided into source and destination. It is necessary to find out the IP address of the connecting server according to the direction of traffic transmission, and use this address as the characteristic value. In the transport layer, it is mainly divided into UDP and TCP. Therefore, the common port field of the two is selected as a feature. This field is also divided into source and destination. Therefore, the port address of the connection server must also be found according to the direction of traffic transmission. However, TCP has more flags fields in the header than UDP, which is used to control the transmission status, so it is also very suitable for use as a characteristic value. When the packet is UDP, it is replaced by zeros. The following will add each header feature to test the impact of using different header features and accuracy. The training parameters are shown in Table 7.
Table 7. P-MLP model feature evaluation training parameters.

| Parameters                | Values  |
|---------------------------|---------|
| Optimizer                 | SGDM    |
| Learning rate ($\eta$)    | 0.0025  |
| Number of Training (Epochs) | 200     |
| Number of Batches (Batch) | 128     |

In order to test the impact of different header features on the accuracy of training, the characteristics of different headers are taken and divided into cases I to V. In I, only the IP protocol field is used to identify the packet as TCP or UDP. In II, the IP protocol and TCP flags fields are used. In addition to identifying the packet type, the packet transmission status is also taken into consideration. Using the IP protocol and port fields in III to take the server’s port into consideration. Use the IP protocol, TCP flags, and port fields in the IV are considered to the server’s port and transmission status together. Finally, the IP protocol, IP address, TCP flags and port fields are used in V to add all the features together, and even the server IP is also considered.

The results can be shown in Figure 17. It can be seen that the more header feature values are added, the higher the accuracy, recall rate and F1-Score will be. When using the IP protocol and adding the TCP flags field, although the accuracy can reach 0.75, the recall rate is low, so the F1-Score is only about 0.7.

Figure 17. The impact of header features on classification performance.

The classification results can be seen in Figure 18a,b. Both browsing and bulk data have serious classification failures. However, after adding the port feature, it can be seen from Figure 18c,d that the failure of bulk data classification has improved, so the recall rate has increased, and F1-Score has risen to about 0.83. After adding the server IP address, it can be seen from Figure 18e that the classification accuracy has slightly increased, so the F1-Score increased to 0.84, which is the best result in the test.
Through the evaluation of the features used in the training of the P-MLP model architecture, the header features are selected as the IP protocol, TCP flags, and port fields. In the formal training, the training parameters of the P-MLP model are shown in Table 8.

Table 8. P-MLP model training parameters.

| Parameters               | Values |
|--------------------------|--------|
| Optimizer                | SGDM   |
| Learning Rate ($\eta$)   | 0.0025 |
| Simulation Time (Epochs) | 1000   |
| Batches                  | 64     |

The accuracy during training is shown in Figure 19. As the number of training increases, the accuracy gradually increases. After about 200 rounds, the increase in the verification accuracy can be seen to slow down. When it reaches about 250 rounds, the model accuracy is about 0.85 and tends to be stable, and the accuracy no longer increases.
3. Results

After the training is completed, the accuracy and classification results of the P-MLP are verified by using 20% test data not involved in the training. The results are shown in Table 9. It can be observed that the average F1-Score score can reach 0.82, indicating that it can accurately predict more than 80% of the traffic to the correct classification. Moreover, in the two categories of bulk data and real-time, its F1-Score score can reach 0.92 and 0.98, and the classification effect is good. However, it can be seen that in the classification of streaming and browsing, F1-Score tends to be low, but the test results can reach 0.68 and above 0.73.

Table 9. P-MLP model test classification performance.

| Category    | Precision | Recall | F1-Score |
|-------------|-----------|--------|----------|
| Bulk Data   | 0.87      | 0.98   | 0.92     |
| Browsing    | 0.64      | 0.71   | 0.68     |
| Streaming   | 0.86      | 0.64   | 0.73     |
| Real-Time   | 0.94      | 0.96   | 0.95     |
| Average     | 0.83      | 0.82   | 0.82     |
The detailed status of the classification can be viewed through the confusion matrix as shown in Figure 21. It can be observed that the F1-Score of bulk data and real-time classifications, almost all packets can be correctly predicted to the appropriate classification. In the streaming classification, about 800 packets were incorrectly predicted to the browsing classification, which resulted in low accuracy and recall rates of these two classifications, and reduced F1-Score. The problem is that in the packets of the browsing classification and the streaming classification, the data may not be sorted in time series during training, and the data transmitted on the two classifications mostly use the TLS protocol, which results in incorrect predictions. However, the overall F1-Score can still reach above 0.8, so the traffic behavior can still make correct QoS prediction and classification.

![Figure 21. Confusion matrix of P-MLP test.](image)

4. IP/ASN Query Results

Before the final output result, it will cooperate with the result of querying IP/ASN in the previous process to achieve more refined traffic classification. The accuracy is improved through the majority voting method. Finally, $K$ is set to 13, $W_k$ is set to 3, and the output result is used as an example to display the IP/ASN query result. The query results are divided into DNS reverse check and ASN database query methods that are classified according to the expected traffic type. The results of the bulk data category IP/ASN query are shown in Table 10. Since the FTPS and SCP transmission servers are private IP addresses and DNS reverse, ASN queries cannot be performed.

| Service     | Category   | Count | Server IP   | DNS   | ASN   |
|-------------|------------|-------|-------------|-------|-------|
| FTPS Control| Bulk Data  | 5     | 10.10.39.8  | N/A   | N/A   |
| FTPS Data   | Bulk Data  | 5     | 10.10.39.8  | N/A   | N/A   |
| SCP         | Bulk Data  | 5     | 10.10.39.8  | N/A   | N/A   |

Browsing category IP/ASN query results are shown in Table 11. Through DNS reverse query, the SOA and PTR, can be obtained. The PTR record is to set the IP counter-check record, and the SOA record shows that the IP does not have a PTR set where it will still respond to the SOA record. The ASN query result will respond to the ASN of the IP and the registrant information. The response results show that Line, Messenger, Facebook, Google and Mail services can all find the service providers accessed. However, Discord’s query result is CLOUDFLARENET, and Discord uses the CDN acceleration service provided by Cloud Flare. It cannot directly query the connected server behind it, and it cannot judge or identify the application.

![Table 10. Bulk data category IP/ASN query results.](image)
Table 11. Browsing category IP/ASN query results. * stands for all.

| Service     | Category | Count | IP                        | DNS                | ASN         |
|-------------|----------|-------|---------------------------|--------------------|-------------|
| Line Text   | Browsing | 5     | 47.92.249.2               | SOA                | AS 38631    |
|             |          |       | 203.104.153.129           | *.naver.jp.        | LINE Corporation |
|             |          |       | 203.104.150.2             |                    |             |
| Discord Text| Browsing | 1     | 162.159.129.233           | SOA                | AS 13335    |
|             |          |       | 162.159.138.232           | *.cloudflare.com   | CLOUDFLARENET |
| Messenger Text| Browsing | 5     | 31.13.87.9                | PTR                | AS 32934    |
|             |          |       |                           | *.facebook.com.    | FACEBOOK    |
| Facebook    | Browsing | 3     | 31.13.87.36               | PTR                | AS 32934    |
|             |          |       |                           | *.facebook.com.    | FACEBOOK    |
| Google Search| Browsing | 5     | 172.217.160.68            | PTR                | AS 15169    |
|             |          |       | 172.217.24.4              | *.1e100.net.       | GOOGLE      |
| Mail        | Browsing | 4     | 13.107.18.11              | SOA                | AS 8068     |
|             |          |       |                           | *.azure-dns.com.   | MICROSOFT-CORP-MSN-AS-BLOCK |

The results of the browsing category IP/ASN query are shown in Table 12. Since multimedia streaming services require a considerable amount of bandwidth support, enterprises and ISPs will deploy CDN or caching services to cope with the huge traffic demand. Therefore, Facebook, YouTube and the server IPs connected to Netflix, are mostly cache servers deployed by Chunghwa Telecom (HINET) in Taiwan. However, YouTube and Netflix will still connect to the company’s own server if the cache fails. Spotify deploys its server on the cloud service fast, so it is impossible to obtain identifiable results through DNS reverse lookups or ASN lookups. Apple Music uses the company’s own server, so it can successfully inquire by the service Apple provider. However, the query result is difficult to successfully classify the real service provider information to determine the application by using the IP/ASN method when the classification type is streaming.

Real-time IP/ASN query results are shown in Table 13. In real-time voice and video services, P2P and server forwarding are usually connected. Results here show that whether Messenger is on video or voice services, the query result is reflected as the IP of Chunghwa Telecom users. Hence, it can be known that Messenger is connected through P2P connection between users. Line uses the server forwarding method that the query result is displayed as Line Corporation, and the service traffic can be effectively judged. Discord deploys the server on the I3D cloud service and is unable to obtain the result of validly identifying the service.

After verifying the results, the most effective service for querying information through IP/ASN is the service on the Browse classification. Since most of the services classified by browse are web or mail services, the servers on web services and mail usually have a more complete FQDN. Moreover, the service provider will set up corresponding PTR records, and there will be no situations such as direct connection with other users through P2P methods. Therefore, compared with other types of services, it is more effective to provide effective identification service results when querying IP/ASN information. However, when service providers use cloud services, CDN services, or caching, their real server IP locations are hidden, and only cloud or CDN operator information can be found, resulting in limited traffic classification. In addition, when the connection behavior is P2P or the target server is a private IP, the IP/ASN information cannot be obtained. Therefore, the IP/ASN query can only provide auxiliary identification information for the classification results of the P-MLP model, and when the query results are available, the traffic can be further subdivided. When the effective result cannot be queried, the corresponding QoS performance can also be provided through the behavior classification.
Table 12. Streaming IP/ASN query results. * stands for all.

| Service     | Category       | Count | IP              | DNS               | ASN                      |
|-------------|----------------|-------|-----------------|-------------------|--------------------------|
| Facebook    | Browsing       | 2     | 203.74.69.82    | 203.74.69.210     | AS 3462 HINET Data Communication Business Group |
|             | Streaming      | 3     | 203.74.69.146   |                   |                          |
| YouTube     | Browsing       | 2     | 210.61.221.143  | 210.61.221.79     | AS 15169 GOOGLE          |
|             | Streaming      | 3     | 216.58.200.238  | 172.217.160.99    |                          |
| Netflix     | Browsing       | 1     | 203.75.84.4     | 203.75.84.6       | AS 3462 HINET Data Communication Business Group |
|             | Streaming      | 4     | 45.57.58.139    | N/A               | AS 2906 AS-SI (Netflix Streams Services) |
| Spotify     | Streaming      | 5     | 151.101.78.248  |                   | AS 54113 FASTLY          |
| Apple Music | Browsing       | 4     | 17.253.117.202  | N/A               | AS 6185 APPLE-AUSTIN     |
|             | Streaming      | 1     | 17.253.117.203  |                   |                          |

Table 13. Real-time category IP/ASN query results. * stands for all.

| Service       | Category   | Count | IP              | DNS               | ASN                      |
|---------------|------------|-------|-----------------|-------------------|--------------------------|
| Discord       | Screen     | 10    | 43.239.137.237  | 103.194.167.72    | AS 49544 I3DNET          |
| Sound/Send    | Real-Time  | 5     | 43.239.137.237  |                   |                          |
| Messenger     | Real-Time  | 5     | 43.239.137.237  | *emome-ip.hinet.net | AS 17421 EMOME-NET Mobile Business Group |
| Voice         | Real-Time  | 5     | 147.92.130.102  | *naver.jp.        | AS 38631 LINE Corporation |
| Video         | Real-Time  | 5     | 147.92.179.66   | 147.92.179.62     |                          |
| Line          | Real-Time  | 5     | 147.92.147.40   | 147.92.179.77     |                          |
| Voice         | Real-Time  | 5     | 147.92.147.35   |                   |                          |

5. Conclusions

This paper proposes a hybrid traffic classification method based on machine learning using the P-MLP model and majority voting method to classify the encrypted traffic in the network. According to usage behavior and cooperates with information such as IP/ASN, port and DPI, it can be more accurate and faster. The traffic QoS category is classified based on usage requirements. Especially when the same application or protocol has different QoS requirements, the P-MLP model can also accurately identify the classification. Therefore, the network provides the correct QoS according to the actual requirements. Cooperating with the classification results and IP/ASN query, the traffic can be classified in a more fine-grained manner. The experimental results show that, when the behavior
classification method is used, the F1-Score of Datanet MLP can be 4% higher in the test phase of the P-MLP model. When using a test data set to simulate actual traffic, the F1-Score can be higher by 24%, and it is confirmed that the machine learning method can indeed classify encrypted traffic based on usage behavior. When the P-MLP is combined with the majority voting method, the F1-Score score can be increased by 10%, so that the overall classification F1-Score can reach 0.80 or more. Finally, the machine learning method can more effectively classify the traffic and enable the network to smoothly implement differentiated service QoS.

Author Contributions: Conceptualization, C.-B.L., C.-M.C. (Ching-Mu Chen) and Y.-F.H.; Methodology, C.-B.L. and Y.-F.H.; software, C.-M.C. (Chien-Min Chung), C.-M.C. (Ching-Mu Chen); Validation, C.-M.C. (Ching-Mu Chen), C.-B.L., C.-M.C. (Chien-Min Chung) and Y.-F.H.; Formal analysis, C.-M.C. (Ching-Mu Chen), C.-B.L. and Y.-F.H.; Investigation, C.-M.C. (Chien-Min Chung) and Y.-F.H.; Resources, C.-B.L. and Y.-F.H.; Data curation, C.-B.L. and C.-M.C. (Chien-Min Chung); Writing—original draft preparation, C.-M.C. (Ching-Mu Chen) and Y.-F.H.; Writing—review and editing, C.-M.C. (Ching-Mu Chen) and C.-B.L.; Visualization, C.-M.C. (Ching-Mu Chen); Funding acquisition, C.-M.C. (Ching-Mu Chen) and Y.-F.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

References
1. Olsson, U. Toward the All-IP Vision. *Ericsson Rev.* 2005, 82, 44–53.
2. Sujata, J.; Sohag, S.; Tanu, D.; Chintan, D.; Shubham, P.; Sumit, G. Impact of Over the Top (OTT) Services on Telecom Service Providers. *Indian J. Sci. Technol.* 2015, 8, 145–160. [CrossRef]
3. Flanagan, W.A. VoIP and Unified Communications; Internet Telephony and the Future Voice Network; John Wiley & Sons: Hoboken, NJ, USA, 2012.
4. Aurrecoechea, C.; Campbell, A.T.; Hauw, L. A Survey of QoS Architectures. *Multimed. Syst.* 1998, 6, 138–151. [CrossRef]
5. Dharmapurikar, S.; Krishnamurthi, P.; Sproull, T.; Lockwood, J. Deep Packet Inspection Using Parallel Bloom Filters. In Proceedings of the 11th Symposium on High Performance Interconnects, Stanford, CA, USA, 20–22 August 2003; pp. 44–51.
6. Alpaydin, E. *Introduction to Machine Learning*; MIT Press: Cambridge, MA, USA, 2020.
7. Cascarano, N.; Este, A.; Gringoli, F.; Risso, F.; Salgarelli, L. An Experimental Evaluation of the Computational Cost of a Dpi Traffic Classifier. In Proceedings of the IEEE Globecom, Honolulu, HI, USA, 30 November–4 December 2009; pp. 50–59.
8. Say “Yes” to HTTPS: Chrome Secures the Web, One Site at a Time. Available online: https://www.blog.google/technology/safety-security/say-yes-https-chrome-secures-web-one-site-time/ (accessed on 20 October 2017).
9. Naylor, D.; Schomp, K.; Varvello, M.; Leontiadis, I.; Blackburn, J.; López, D.R.; Papagiannaki, K.; Rodriguez, P.R.; Steenkiste, P. Multi-Context TLS (mcTLS): Enabling Secure In-Network Functionality in TLS. *SIGCOMM Comput. Commun. Rev.* 2015, 45, 199–212. [CrossRef]
10. Lotfollahi, M.; Siavoshani, M.J.; Zade, R.S.H.; Saberian, M. Deep Packet: A Novel Approach for Encrypted Traffic Classification Using Deep Learning. *Soft Comput.* 2020, 24, 1999–2012. [CrossRef]
11. Harju, J.; Kivimaki, P. Co-operation and Comparison of DiffServ and IntServ: Performance Measurements. In Proceedings of the 25th Annual IEEE Conference on Local Computer Networks (LCN 2000), Tampa, FL, USA, 8–10 November 2000; pp. 177–186.
12. Braden, R.; Clark, D.; Shenker, S. RFC1633: Integrated Services in the Internet Architecture: An Overview. *Netw. Work. Group* 1994. [CrossRef]
13. Nichols, K.; Blake, S.; Black, D. RFC 2474: Definition of the Differentiated Services Field (DS field) in the IPv4 and IPv6 Headers. *Netw. Work. Group Internet Soc.* 1998. [CrossRef]
14. Braden, R.; Zhang, L.; Berson, S.; Herzog, S.; Jamin, S. RFC 2205: Resource ReSerVation Protocol (RSVP). *Netw. Work. Group* 1997. [CrossRef]
15. Heinanen, J.; Baker, F.; Weiss, W.; Wroclawski, J. RFC 2597: Assured Forwarding PHB Group. *Netw. Work. Group* 1999. [CrossRef]
16. Babinz, J.; Chan, K.; Baker, F. RFC 4594: Configuration Guidelines for DiffServ Service Classes. *Internet Eng. Task Force (IETF)* 2006. [CrossRef]
17. Szigeti, T.; Hattingh, C.; Barton, R.; Briley, K., Jr. *End-to-End QoS Network Design: Quality of Service for Rich-Media & Cloud Networks*; Cisco Press: Indianapolis, IN, USA, 2013.
18. Reynolds, J. RFC 3232: Assigned Numbers: RFC 1700 is Replaced by an On-line Database. *Netw. Work. Group Internet Soc.* 2002. [CrossRef]
19. Aho, A.V.; Corasick, M.J. Efficient String Matching: An Aid to Bibliographic Search. *Commun. ACM* 1975, 18, 333–340. [CrossRef]
20. Wireshark User’s Guide, Version 3.3.0. Available online: https://www.wireshark.org/download/docs/user-guide.pdf (accessed on 15 March 2020).

21. Deri, L.; Martinelli, M.; Builow, T.; Cardigliano, A. ndpi: Open-source High-speed Deep Packet Inspection. In Proceedings of the 2014 International Wireless Communications and Mobile Computing Conference (IWCMC), Nicosia, Cyprus, 4–8 August 2014; pp. 617-622.

22. Goodfellow, I.; Bengio, Y.; Courville, A. Deep Learning; MIT Press: Cambridge, MA, USA, 2016.

23. Wang, P.; Ye, F.; Chen, X.; Qian, Y. Datanet: Deep Learning Based Encrypted Network Traffic Classification in SDN Home Gateway. IEEE Access 2018, 6, 55380–55391. [CrossRef]

24. Ng, A. Sparse Autoencoder. Available online: https://web.stanford.edu/class/cs294a/sparseAutoencoder.pdf (accessed on 15 March 2020).

25. Lawrence, S.; Giles, C.L.; Tsoi, A.C.; Back, A.D. Face Recognition: A Convolutional Neural-Network Approach. IEEE Trans. Neural Netw. 1997, 8, 98–113. [CrossRef] [PubMed]

26. Gil, G.D.; Lashkari, A.H.; Mamun, M.S.I.; Ghorbani, A.A. Characterization of Encrypted and VPN Traffic Using Time-Related Features. In Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP), Rome, Italy, 19–21 February 2016; pp. 407–414.

27. Russell, S.; Norvig, P. Artificial Intelligence: A Modern Approach; Pearson Education, Inc.: Upper Saddle River, NJ, USA, 2002.

28. Hinton, G.E.; Sejnowski, T.J.; Poggio, T.A. Unsupervised Learning and Map Formation: Foundations of Neural Computation; MIT Press: Cambridge, MA, USA, 1999.

29. Zhu, X.; Goldberg, A.B. Introduction to Semi-Supervised Learning; Lectures on Artificial Intelligence and Machine Learning; Morgan & Claypool Publishers: Morgan, VT, USA, 2009; Volume 3, pp. 1–130.

30. Kaelbling, L.P.; Littman, M.L.; Moore, A.W. Reinforcement Learning: A Survey. J. Artif. Intell. Res. 1996, 4, 237–285. [CrossRef]

31. Labayen, V.; Magaña, E.; Morató, D.; Izal, M. Online classification of user activities using machine learning on network traffic. Comput. Netw. 2020, 181, 107557. [CrossRef]

32. Kramer, M.A. Nonlinear Principal Component Analysis Using Autoassociative Neural Networks. AIChE J. 1991, 37, 233–243. [CrossRef]

33. Sasaki, Y. The truth of the F-measure. Teach. Tutor Mater. 2007, 1, 2007.

34. Nair, V.; Hinton, G.E. Rectified Linear Units Improve Restricted Boltzmann Machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 807–814.