Processing and classifying IP packet data on the Internet based on machine learning

Vuong Xuan Chi*, Nguyen Kim Quoc**

Faculty of Information Technology, Nguyen Tat Thanh University
*vxchi@ntt.edu.vn, **nkquoc@ntt.edu.vn

Abstract

Nowadays, the continuous development of information technology, communication over the Internet is increasing rapidly, and network congestion has become an alarming issue. To develop communication network infrastructure in a large city, a country, or globally, streamlining and controlling network data flow to optimize communication processes and minimize network congestion is crucial and necessary. In this study, the authors analyze and process data according to the delay of Internet Protocol (IP) packets, using machine learning models with the Random Forest (RF) and the Support Vector Machines (SVM) method to classify IP packets. The primary goal of classifying packets by delay is to optimize network performance by prioritizing processing of low-delay packets, ensuring stable and uninterrupted online services such as video streaming and voice calls. Furthermore, it is easy to manage and control packet traffic, hence minimizing network congestion at the router.

Received 10/03/2024
Accepted 05/05/2024
Published 20/06/2024

Keywords

IP packet classification, IP network, network congestion, machine learning, random forest

1 Introduction

Classifying IP packet stream data in Internet and communication networks is highly important. Packet classification in IP networks has numerous common applications such as traffic control, bandwidth management, intrusion detection, traffic analysis, and many others. Accurate and efficient packet classification in IP networks plays a significant role in designing and operating communication network systems. In machine learning, some network classification techniques involve statistical analysis of attributes of IP data streams and use unsupervised learning mechanisms to cluster streams into groups based on similarity [1]. Additionally, algorithms and methods such as Artificial Neural Networks (ANN), Perceptron (PLA), K-Nearest Neighbor (KNN) method, Decision Tree (DT) method, Random Forest (RF) method, and Support Vector Machine (SVM) method are applied to determine data features and can classify data into separate groups based on the attribute information of the data service [2-4].

The IP network is distributed and complex, potentially millions of packets transmitted through the network per a second. To provide fast responses to network devices within the system, latency becomes one of the critical factors. Among various types of attributes, queue delay has a more significant impact on the network than other types of delays [5]. Additionally, measuring traffic control, load balancing techniques, routing, and anomaly detection all identify causes of high-latency packets, posing challenges for network management and operation in the future [6].

The rest of the paper is structured as follows. Section 2 presents the research methodology. In section 2.1, the authors study the structure of IP packets. Approach DT and RF machine learning models for applying to IP packet classification in section 2.2. Next, in section 2.3,

https://doi.org/10.55401/2xyvkg06

analyze and process IP packet data, monitor the proportions of incoming and outgoing data streams from source and destination addresses. In section 2.4, the paper analyzes and extracts data related to delay and in section 2.5, the authors calculate the average delay of IP packets. Subsequently, in section 3, experimenting with machine learning models, particularly SVM and RF models for classifying packets based on delay. Comparing the classification results of DT, RF, SVM, KNN models, and find that the RF model achieves the highest accuracy in the evaluation results. Finally, in section 4, the authors conclude the paper.

2 Research Methods

2.1 IP packet structure and IP packet classification

Packet data consists of small units of data encapsulated into packets for transmission over a network. These packets contain information about the source, destination, network protocol, IP address, technical parameters of the transmission path, actual data to be transmitted, and other relevant information.

The IP packet structure is a format used in various network protocols to transmit packets across the Internet. The IP packet structure consists of a packet header and packet data. The specific fields of the IP header include 12 mandatory attributes, with a total length of 20 bytes (excluding Data and Options). Refer to Table 1 for a description of the structure of the IPv4 Header.

Table 1 Structure of IP Header (Version 4)

<table>
<tr><td>Version</td><td>IHL</td><td>Type of Service</td><td colspan="2">Total Length</td></tr>
<tr><td colspan="3">Identification</td><td>Flags</td><td>Fragment Offset</td></tr>
<tr><td>Time to Live</td><td colspan="2">Protocol</td><td colspan="2">Header Checksum</td></tr>
<tr><td colspan="5">Source IP Address</td></tr>
<tr><td colspan="5">Destination IP Address</td></tr>
<tr><td colspan="4">IP Options</td><td>Padding</td></tr>
<tr><td colspan="5">Data</td></tr>
</table>

The fields are described as follows:

Version: The version of the IP protocol, typically IPv4 or IPv6; Internet Header Length (IHL): The length of the IP header, measured in 32-bit words. The minimum value of IHL is 5, and the maximum is 15; Type of Service (TOS indicates how to process the data packet, whether there is priority or not, and the allowable delay of the data packet. This field is often used to perform network service quality management; Total Length: The total size of the IP packet, measured in bytes; Identification: Used to identify related packets during fragmentation and packet reassembly; Flags: The first 3 bits of the 16-bit field, used to determine whether a packet has been fragmented or if it is the last packet in the fragmentation process; Fragment Offset: The position of each fragment within the packet after fragmentation; Time to Live (TTL): The lifespan of the IP packet, measured in seconds. When passing through a router, the TTL value decreases by one unit; Protocol: Indicates which protocol of the upper layer (Transport layer) will receive the data after IP diagram processing at the Network layer is completed or indicates which protocol of the upper layer sends the segment to the lower layer. Network packaged into IP Diagram, each protocol has 1 code (06: TCP, 17: UDP, 01: ICMP...); Header Checksum: The integrity of the IP header; Source Address: Identifies the source IP address of the packet. Destination Address: Identifies the destination IP address of the packet; IP Options: An optional field that may or may not be present in the IP header is used for various purposes such as measuring latency and measuring service quality in the IP network. Finally, Padding: Zeros are added to this field to ensure the IP Header is always a multiple of 32 bits [7].

When an IP packet is generated, it contains information about the source, destination, and content of the packet. IP packet classification clearly defines how packets are processed and forwarded in the network. The primary purposes of IP packet classification include Routing: IP packets are classified to optimize routing paths from source to destination; Network Management: Assists in network control and management; Packet information can be used for monitoring and analyzing network traffic [8], ensuring quality of services and implement security and confidentiality measures.

Additionally, it can be classified to prioritize resource allocation within the network, for example, in a Quality of Service (QoS) model, packets are classified for prioritization based on bandwidth requirements, latency, or priority levels, aiding in the enforcement of specific network policies. Furthermore, there are various network classifications utilizing Software-Defined Networking (SDN), including network access control, application of security regulations, or prioritization of services for specific applications [9].

Therefore, the processing and classification of IP packets play a significant role in handling packets across the Internet, ensuring efficient communication, and enforcing defined network requirements and policies. With the information provided in the IP Header, one can establish rules and models for classifying IP packet data based on their attributes and characteristics [10].

2.2 Machine learning model using Random Forest method to classify IP packet data

2.2.1 The model applies Decision Tree

A model using a Decision Tree [11]. can be used to classify IP packets based on characteristics. In particular, the machine learning method uses the Decision Tree algorithm to classify IP packets as follows:

Data preparation

+ Each IP packet will be described by a characteristic vector, including attributes such as source address, destination address, protocol usage, packet size, and many other related attributes.

+ Class labels of IP packets need to be assigned, for example, classes according to different protocols...

Building a Decision Tree

+ Define branching rules based on characteristics and their values to classify IP packets.

+ Algorithm of decision tree machine learning method:

```
if (feature_1 > threshold_1):
    if (feature_2 < threshold _2):
        ... # Subsequent branches
    else
        ... # ...
else
    ...#...
```

+ Decision Tree setup is implemented by selecting optimal features and thresholds to maximize classification accuracy.

Prediction

+ For each new IP packet, the packet will traverse the decision tree by examining each node and following the branching rules until reaching a leaf node.

+ Leaf nodes represent the final classification result for the IP packet.

The process of constructing trees and finding optimal branching rules is carried out through various algorithms such as ID3, C4.5, and CART [12].

2.2.2 Classification method using Random Forest machine learning model

The classification method uses a Random Forest is an ensemble of multiple Decision Trees, with each Decision Tree trained independently [13].

A general way to represent Random Forests:

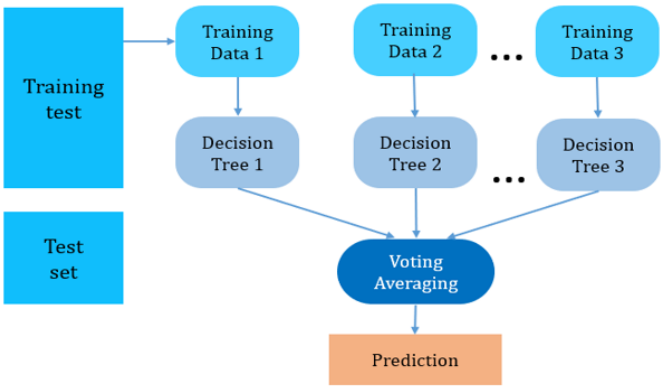

Figure 1 Random Forest algorithm diagram

Decision tree training

- Randomly select a subset of the training set and random features.
- Build Decision Trees on the subset of data. Decisions at each node in the tree are based on the value of a feature and a threshold.
- Continue building the tree based on branching rules using features and thresholds to optimize classification criteria (e.g., entropy) [14].
- Repeat the above process to set up multiple separate Decision Trees.

Random Forest prediction

For each new data point, it passes through all Decision Trees in the Random Forest. For each Decision Tree, the prediction result is obtained and recorded from the tree. Then, the majority vote is calculated from all predictions of the trees to determine the final classification result for the data classification model as illustrated in Figure 1.

2.2.3 Applying Random Forest for IP packet data classification.

Considering the IP packet data as the training set of the model $\mathcal{A} = \{(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)\}$ (1) include $\mathcal{K}$ observations. The Random Forest algorithm utilizes the method of sampling IP packet data to reconstruct into subsets B (initially is empty) has IP packets. Repeatedly sampling from the training set $\mathcal{A}$, which means that $\mathcal{H}$ repeated samplings using the method of putting IP packets into bags (bagging) from observations to create a packet data set IP $\mathcal{B}_i$ (2):

$$\mathcal{B}_i = \left\{\left(x_1^{(i)}, y_1^{(i)}\right), \left(x_2^{(i)}, y_2^{(i)}\right), \ldots, \left(x_H^{(i)}, y_H^{(i)}\right)\right\} \quad (2)$$

Out-of-bag IP packets are observations that are not included in the sampling process of a subset $\mathcal{B}_i$. Each set of packets IP $\mathcal{B}_i$ will build a Decision Tree model and the returned result is $p(\hat{y}_j^{(i)}) = f_i(x_j)$, this is a predictive model.

Where $p(\hat{y}_j^{(i)})$: the j observation (prediction) from the i IP packet model.

x_j: vector (input value).

$f_i(.)$: prediction model for i IP packet (prediction function).

At this point, from the decision tree, the prediction is the average value (3) or vote (4) of B decision trees.

- From the submodel, we can calculate the average value of the predictions (this is the prediction model):

$$\hat{y}_j = \frac{1}{B}\left(\frac{1}{B}\sum_{i=1}^{B} \hat{y}_j^{(i)}\right) \quad (3)$$

- From the submodels, voting can be performed to select the prediction labels based on the highest frequency (this is a classification model):

$$\hat{y}_j = \arg\max \sum_{i=1}^{B} = p(\hat{y}_j^{(i)}) \quad (4)$$

In the case of a prediction model, there will be a variance (5) of:

$$\sigma_{\hat{y}}^2 = \text{var}\left(\frac{1}{B}\sum_{i=1}^{B} \hat{y}^{(i)}\right) = \frac{1}{B^2}\left[\sum_{i=1}^{B} \text{var}\left(\hat{y}^{(i)}\right) + 2\sum_{1\le h<k\le B} \text{cov}\left(y^{(h)}, y^{(k)}\right)\right] \quad (5)$$

Because the result of one sub-model is independent and has no influence on another sub-model B, So from the model prediction results, it is likely that they will be independent of each other. That is, Covariance $\text{cov}\left(y^{(h)}, y^{(k)}\right) = 0, \forall 1 \le h < k \le B$. Besides, it is assumed that the models have uniform quality, specified by the variance being identical, so $\text{var}\left(\hat{y}^{(i)}\right) = \sigma^2, \forall i = \overline{1,B}$. Therefore, it follows that (6):

$$\sigma_{\hat{y}}^2 = \frac{1}{B^2}\left[\sum_{i=1}^{B} \text{var}\left(\hat{y}^{(i)}\right)\right] = \frac{1}{B^2} B\sigma^2 = \frac{1}{B}\sigma^2 \quad (6)$$

In a Random Forest model, there are a large number of decision trees. As a result, the combined prediction's mean and variance will decrease by a factor of B compared to solely using a single model [15, 16].Therefore, the variance can decrease multiple times from the ensemble prediction model, resulting in a better prediction.

2.3 Data analysis and processing

2.3.1 Data and data processing

Network Data Dataset IP packet contains 3.577.296 instances on single CSV file [17]. The data was experimented with on Google Colaboratory Pro using the Scikit-learn library. The author used machine learning methods, especially the Random Forest machine learning method. The paper extracts data attributes related to the transmission time of IP packets, focusing on packet latency and reorganizing them according to latency. The attributes of the dataset are described in Table 1 as follows:

Table 1 Dataset attributes and description.

Groups of attributes	Attributes	Description
Network identifiers	FlowID; Source IP; Source Port; Destination IP; Destination Port; Protocol; Timestamp	The attributes contain all information related to the source and destination of the Internet stream, namely IP addresses, transport layer protocols, and ports.
Inter arrival times	Flow Duration; Flow IAT Mean; Flow IAT std; Flow IAT Max; Flow IAT Min; Bwd IAT Total; Bwd IAT Mean; Bwd IAT Std; Bwd IAT Max; Bwd IAT Min; Fwd IAT Total; Fwd IAT Mean; Fwd IAT Std; Fwd IAT Max; Fwd IAT Min.	The attributes contain all information related to the time between arrivals.
Flag features	Fwd PSH flags; Bwd PSH flags; Fwd URG flags; Bwd URG flags; CWE Flag Count; ECE Flag Count FIN Flag Count; SYN Flag Count; RST Flag Count; PSH Flag Count; ACK Flag Count; URG Flag Count.	The attributes display information related to all flags present in the packet header, such as Push Flag,

		Urgent Flag, Fin Flag, and other flags.
Flow descriptors	Total Fwd Packets; Total Bwd Packets; Total Length of Fwd Packets; Total Length of Bwd Packets; Fwd Packet Length Max; Fwd Packet Length Max; Fwd Packet Length Min; Fwd Packet Length Mean; Fwd Packet Length Std; Bwd Packet Length Max; Bwd Packet Length Min; Bwd Packet Length Mean; Bwd Packet Length Std; Flow Bytes S; Flow Packets S; Min Packet Length; Max Packet Length; Packet Length Mean; Packet Length Std; Packet Length Variance; Down Up Ratio; Init Win bytes forward; Init Win bytes backward; act data pkt fwd; min seg size forward; Label; L7Protocol; ProtocolName; Avg Fwd Segment Size; Avg Bwd Segment Size; Fwd Avg Bytes Bulk; Fwd Avg Packets Bulk; Fwd Avg Bulk Rate; Bwd Avg Bytes Bulk; Bwd Avg Packets Bulk; Bwd Avg Bulk Rate.	The attributes encompass all information related to the Internet stream, including packet count, volume, and standard deviation among other details, in both forward and backward directions.

Visualizing the experimental IP packet dataset helps to understand the attributes of a packet, thereby aiding in detecting relationships and trends within the data and packet attributes such as source and destination IP addresses, source and destination ports, protocol information, and data content. Figure 2 illustrates the attributes of the packet data.

	Flow.ID	Source.IP	Source.Port	Destination.IP	Destination.Port	Protocol	Timestamp	Flow.Duration	Total.Fwd.Packets	Total.Backward.Packets	.
0	172.19.1.46-10.200.7.7-52422-3128-6	172.19.1.46	52422	10.200.7.7	3128	6	26/04/201711:11:17	45523	22	55	
1	172.19.1.46-10.200.7.7-52422-3128-6	10.200.7.7	3128	172.19.1.46	52422	6	26/04/201711:11:17	1	2	0	
2	10.200.7.217-50.31.185.39-38848-80-6	50.31.185.39	80	10.200.7.217	38848	6	26/04/201711:11:17	1	3	0	
3	10.200.7.217-50.31.185.39-38848-80-6	50.31.185.39	80	10.200.7.217	38848	6	26/04/201711:11:17	217	1	3	
4	192.168.72.43-10.200.7.7-55961-3128-6	192.168.72.43	55961	10.200.7.7	3128	6	26/04/201711:11:17	78068	5	0	

5 rows × 87 columns

Figure 2 Experimental dataset - IP packet data

The data consists of numerous features grouped into categories such as Network identifiers, Interarrival times, Flag features, Flow descriptors, etc. These features can be selected to create attributes for labeling purposes. The attributes in the dataset are visually represented as shown in Figure 3.

	Source.Port	Destination.Port	Protocol	Flow.Duration	Total.Fwd.Packets	Total.Backward.Packets	Total.Length.of.Fwd.Packets	Total.L
count	415338.000000	415338.000000	415338.000000	4.153380e+05	415338.000000	415338.000000	4.153380e+05	
mean	37161.925263	12535.885267	6.006270	2.105026e+07	55.272852	63.188695	3.853647e+04	
std	22211.921069	21033.477837	0.345001	3.700406e+07	826.936626	1398.362427	1.523026e+06	
min	0.000000	0.000000	0.000000	1.000000e+00	1.000000	0.000000	0.000000e+00	
25%	3128.000000	443.000000	6.000000	5.600000e+02	2.000000	1.000000	6.000000e+00	
50%	48105.000000	3128.000000	6.000000	3.141205e+05	5.000000	4.000000	3.060000e+02	
75%	54070.000000	3128.000000	6.000000	1.992307e+07	15.000000	14.000000	1.688000e+03	
max	65534.000000	65534.000000	17.000000	1.200000e+08	268376.000000	542196.000000	5.006735e+08	

8 rows × 81 columns

Figure 3 Description of IP packet properties

2.3.2 Preprocessing IP packet data

Data normalization in the preprocessing stage involves preparing packet data prior to applying classification methods or analyzing other data [18]. The aim of the preprocessing stage is to eliminate unnecessary information,

handle missing, inaccurate, or noisy values, ensure that the data is informative enough, and format it appropriately for use in machine learning models or data analysis [19].

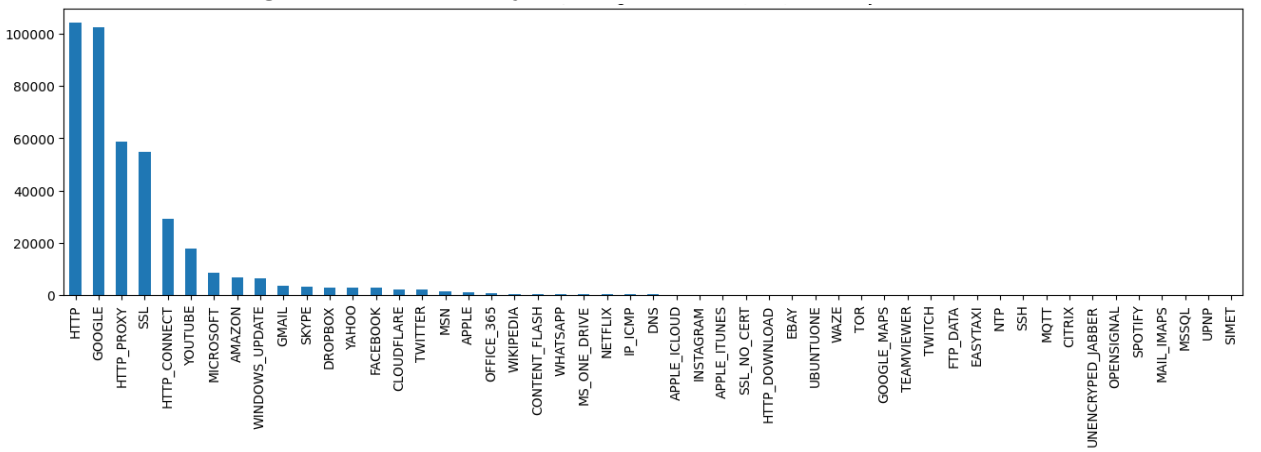

Figure 4 Frequency of occurrence of unprocessed packet data services

Before using data, is important to assess its quality. This involves checking for completeness, accuracy, and suitability for the intended purpose. Quality assurance measures include checking for duplicate data, examining outliers, and comparing it to the original data source (if available). For large-scale data processing systems, optimizing performance can be a crucial factor. Measures such as optimizing database queries, enhancing hardware independence, and optimizing algorithms can be applied. Additionally, the data needs to be normalized to ensure that all features operate on the same range or have the same weight. An important part of preprocessing is extracting meaningful features from the original data, which may include selecting important variables, reducing data dimensionality, or transforming the data to create features. new display. The unprocessed data shown in Figure 4 and Figure 5 are processed data.

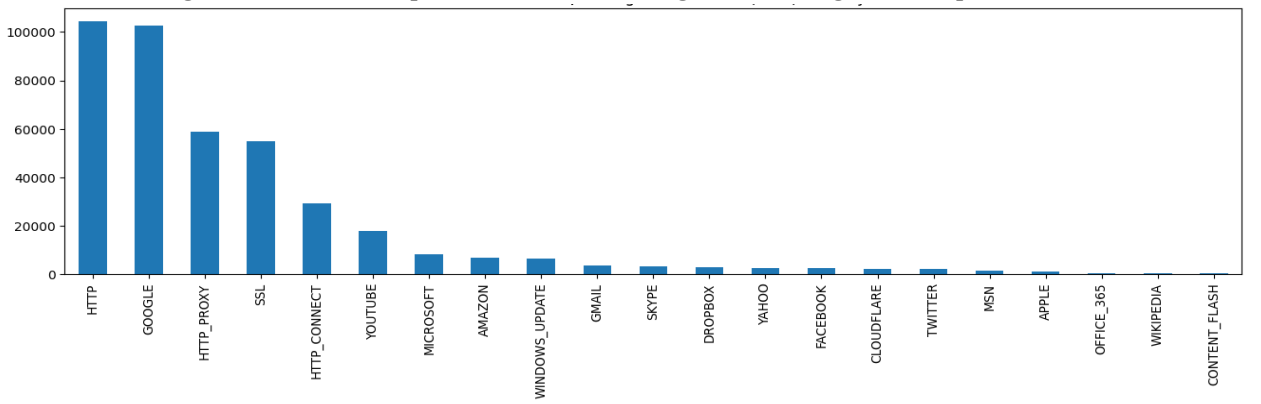

Figure 5 Frequency of occurrence of processed packet data services

In the Figure 5 chart, the data has been stripped of service fields that are rare and appear only once. The most common service attributes such as HTTP, Google, SSL, Youtube, Microsoft, Amazon, Gmail, Skype, Dropbox, Yahoo, Facebook, Windows_update, HTTP_Proxy, HTTP_Connect, etc.

2.3.3. Analyzing IP packets according to IP flows

By tracking the classification ratio of data streams to and from IP addresses, security experts can detect anomalous activities, such as attacks from specific IP addresses or suspicious activity patterns from a particular IP range. Additionally, distinguishing between data streams from source and destination IP addresses can help organizations gain a better understanding of how computer networks operate and optimize network configurations to improve performance and security. Furthermore, by monitoring the classification ratio of data streams, organizations can quickly detect network issues, such as connectivity or security incidents, and take preventive or corrective

actions. Analyzing IP packet data to identify data streams from source IP addresses is illustrated in Figure 6 and destination IP addresses in Figure 7.

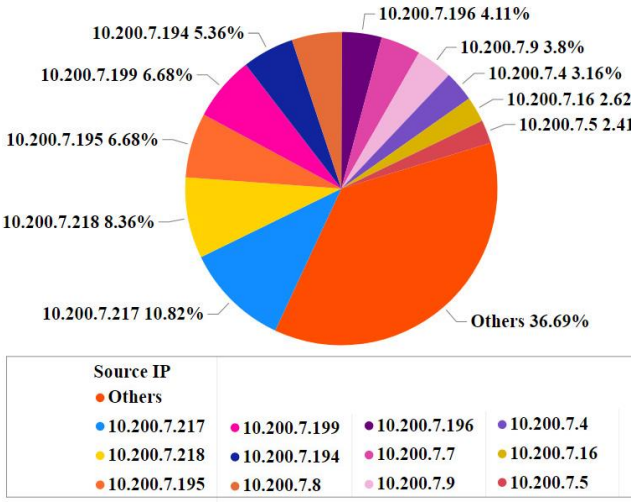

Figure 6 Percentage of IP flows by source address (Source. IP)

In the pie chart Figure 6, other types of IP account for the most proportion, while the source IP is 10.200.7.217 in the data set, accounting for a fairly high proportion of about 10.8 % of the data set. At least the packets have a source IP address of 10.200.7.5. Similarly, as shown Figure 7, IP packets with other IP addresses have a large number of 59.8 %, accounting for more than 50 %. Packets with destination IP address 179.1.4.251 account for a relatively small proportion of 1.01 %, and the largest proportion of packets with destination IP address is 10.200.7.8 has a high proportion of 7.8 %.

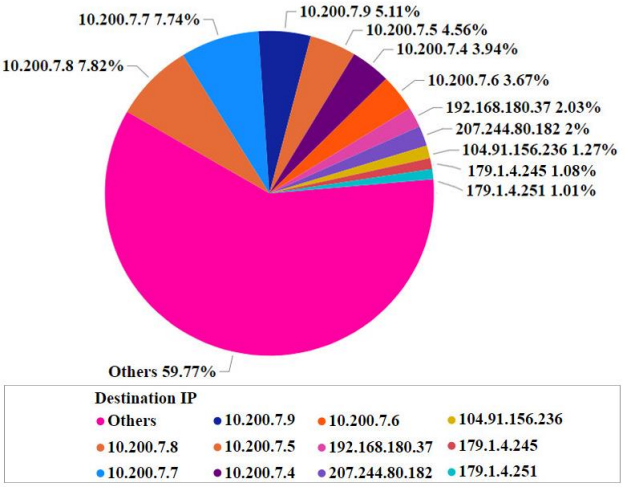

Figure 7 Percentage of IP flows by destination address (Destination.IP)

2.4 Extracting data related to delay

Unnecessary or irrelevant attributes in the classification process can be eliminated, such as MAC addresses, source/destination port numbers unrelated to the classification process. Missing values in IP packet data are cross-checked and handled accordingly. Noise handling methods may involve discarding invalid packets and correcting outlier values. Ensuring that IP packet data converts into an appropriate format for use in machine learning classification models.

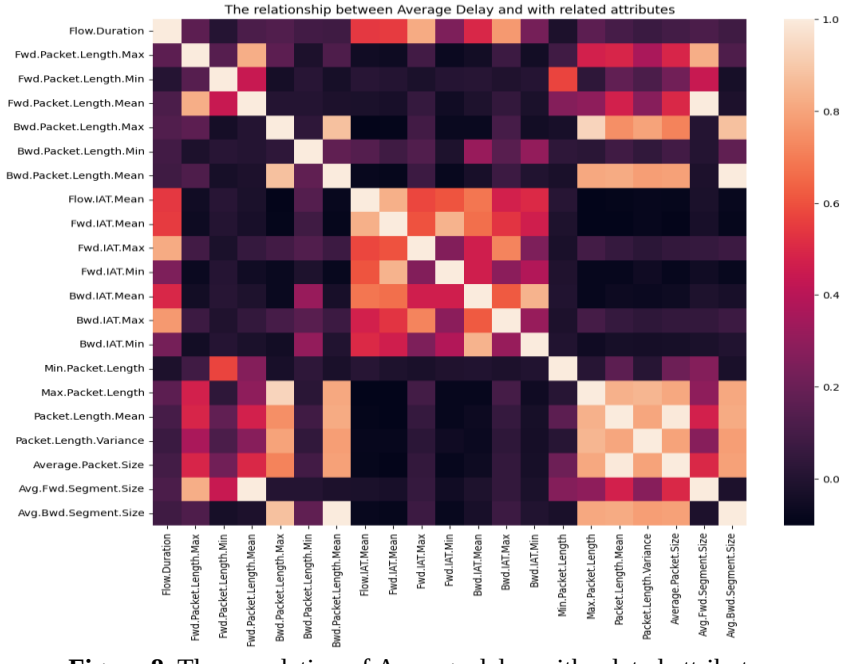

Figure 8 The correlation of Average delay with related attributes.

The attributes in the experimental dataset that have an impact and correlation with delay include Inter Arrival Time (IAT) attributes, such as 'Flow.IAT.Mean', 'Fwd.IAT.Mean', 'Bwd.IAT.Mean', 'Fwd.Packet.Length.Mean', 'Bwd.Packet.Length.Mean', 'Packet.Length.Mean'… Figure 8 illustrates the relationship between delay-related attributes.

Attributes correlating with IP packet delay need calculating the average delay, as shown in Table 3.

Table 3 Description of key attributes related to delay.

Attributes	Explain
Flow.Duration	Time from start to finish of a packet flow.
Flow.IAT.Mean	Time from receiving one packet to receiving the next packet. in a data stream.
Fwd.IAT.Mean	Average time between forward packets (from source to destination) on the data stream
Bwd.IAT.Mean	Average time between reverse packets (destination to source) on the data stream.
Packet. Length. Mean	Average length measures the average size of all packets.
Packet. Length. Mean	Describes the average length of forward packets in a particular data stream.
Fwd.Packet. Length.Mean	Describes the average length of forward packets in a particular data stream.
Bwd.Packet. Length.Mean	Describes the average length of reverse packets in a particular data stream.
Average. Packet.Size	Average size of packets in a signal stream. a signal stream.

2.5 Average delay

Based on the analysis, calculating the average latency relies on multiple latency-related fields in the data such as selecting the latency-related fields, computing the average latency for each field, and combining the average values to compute the overall average latency across all attributes.

The average latency of all IP packets transmitted from source to destination.

$$\text{Average delay} = \frac{1}{n}\sum_{i=1}^{n} delay_i$$

In which:

- $Average\ delay$ is the average value of the delay.
- $Delay_i$ is the delay value of the i[th] data sample
- n is the number of data samples.

Thus, according to the experiment with 6 average values:

$$\text{Average delay} = \frac{969{,}172.8943774045}{6} = 161{,}528.8157295674 \text{ ms}$$

3 Discussion results

3.1 Support Vector Machines model

Using the SVM method to classify packets based on average delay, each packet is labeled with the corresponding class. The SVM model is used to find the best decision boundary to separate between data classes. From the threshold of average delay, classifying IP packets to identify whether a packet destined to arrive has high or low delay, thereby managing and controlling IP packet traffic effectively.

3.2 Results of IP packet training and Random Forest model

From the feature-extracted data of attributes correlated with packet delay on the transmission path, the Random Forest machine learning model is characterized by constructing and combining decision trees during the training process of IP packets. During prediction, each new IP packet traverses through each decision tree, and the prediction is made based on the decisions at the branches and nodes of the tree. Ultimately, the prediction results of each decision tree are aggregated (through majority voting), ultimately providing the prediction result for that IP packet. In Figure 9, the authors visualize the DT to observe in the RF.

After running the experiments:

- *The number of trees in the Random Forest model (n_estimators) is 104.*
- *The number of layers in the Random Forest model (max_depth) is 50.*

(Min_samples_leaf is 4: the number of samples in each leaf of the Decision Tree. Min_samples_split is 10: the minimum number of samples required to continue splitting a node of the Decision Tree).

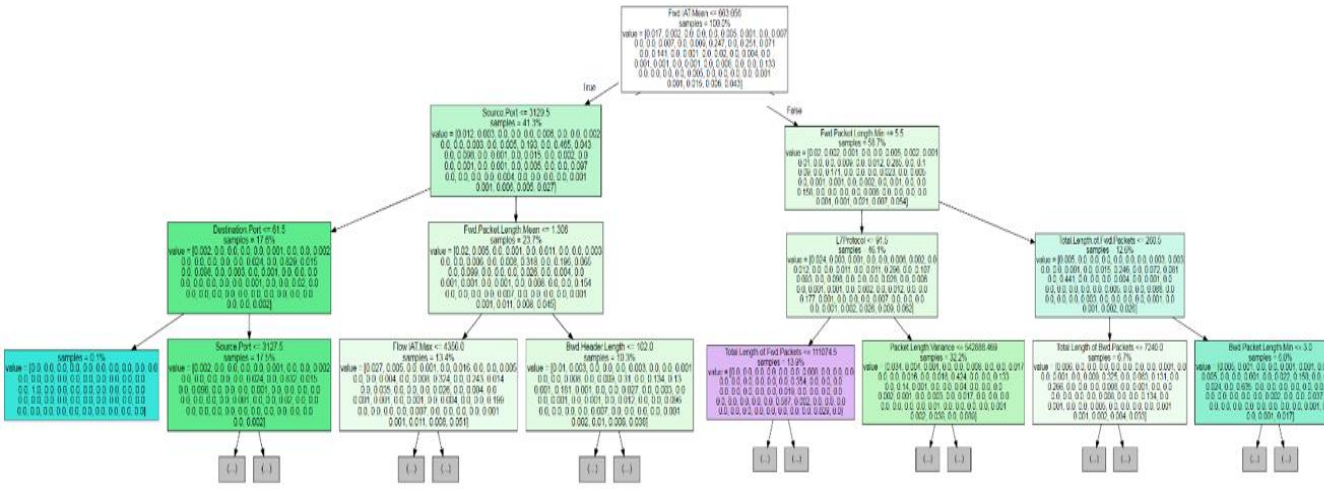

Figure 9 A Decision Tree in the Forest

3.3 Result evaluation

The IP packet classification model using the Random Forest method, the IP packets in the experimental dataset were evaluated for accuracy when comparing the DT, KNN and SVM methods with RF. The accuracy metrics (Accuracy) for IP packet classification on the dataset are as follows Table 3.

Table 3 Test results and comparison of classification methods

Methods	**Accuracy**	**Percentage (%)**
DecisionTree	0.98	98
Random Forest	0.99	99
SVM	0.95	95
KNeighbors	0.74	74

The results of the machine learning process evaluate by machine learning algorithms based on metrics such as Precision, Recall, and F1 score, with the results averaged using a weighted average. Among these algorithms, Random Forest exhibits high accuracy, and shown in Table 4.

Table 4 Metrics are used to evaluate performance

Algorithms	**Precision (%)**	**Recall (%)**	**f1-score (%)**
DecisionTree	98	98	98
Random Forest	99	98	99
SVM	95	95	94
KNeighbors	74	74	73

4 Conclusion

In this paper, the authors investigate the classification of IP packets based on their delay, aiming to optimize network performance by prioritizing the processing of packets with low delay. By utilizing machine learning models, particularly the Random Forest (RF) and Support Vector Machines (SVM) model, the authors analyze and process of IP packets data to make efficient packet classification decisions.

The research aids in improving packet traffic management on the network, minimizing network congestion, and ensuring stable and uninterrupted online services. The results of the study have a wide range of applications in developing Internet communication infrastructure, providing significant benefits for both users and network service providers. Additionally, the paper will contribute to enhancing the scientific research position of Nguyen Tat Thanh University and support students in understanding data science and computer networking.

References

1. Alina Vladutu, DragosComaneci, Ciprian Dobre. (2016). Internet traffic classification based on flows' statistical properties with machine learning. *John Wiley & Sons.*
2. Ernest Yeboah Boateng, Joseph Otoo, Daniel A. Abaye. (2020). Basic Tenets of Classification Algorithms, K-Nearest-Neighbor, Support Vector Machine, Random Forest and Neural Network: A Review. *Journal of Data Analysis and Information Processing.*
3. Aafa J S, Soja Salim. (2014). A Survey on Network Traffic Classification Technique. *International Journal of Engineering Research & Technology (IJERT), Vols. 3 Issue 3, March 2014*.
4. Ola Salman, Imad H. Elhajj, Ayman Kayssi, Ali Chehab. (2020). A review on machine learning–based approaches for Internet traffic classification. *Annals of Telecommunications, Springer*.
5. Roy, Arnab, Joseph Lalnunfela Pachuau, and Anish Kumar Saha. (2021). An overview of queuing delay and various delay based on algorithms in networks. *Computing 103.10: 2361-2399. Springer.*
6. Chefrour, Djalel. (2021). One-way delay measurement from traditional networks to sdn: A survey. *Association for Computing Machinery, ACM Computing (CSUR) 54.7: 1-35.*
7. Jon Postel. Internet Protocol. (1981). *Information Sciences Institute University of Southern California 4676 Admiralty Way Marina del Rey, California 90291*.
8. Anita P, Manju Devi. (2021). High performance modified bit-vector based packet classification module on low-cost FPGA. *International Journal of Electrical and Computer Engineering (IJECE), Vols. 11, No. 5, pp. 3855~3863.*
9. Mohammad Reza Parsaei, Mohammad Javad Sobouti, Seyed Raouf khayami, Reza Javidan. (2017). Network Traffic Classification using Machine Learning Techniques over Software Defined Networks. *International Journal of Advanced Computer Science and Applications (IJACSA), Vol. No.7.*
10. Eric Liang, Hang Zhu, Ion Stoica. (2019). Neural Packet Classification. *SIGCOMM , ACM ISBN 978-1-4503-5956-6/19/08.*
11. Harsh H. Patel, Purvi Prajapati. (2018). Study and Analysis of Decision Tree Based Classification Algorithms. *International Journal of Computer Sciences and Engineering, Vols. 6(10), E-ISSN: 2347-2693.*
12. Muhammad Shafiq, Xiangzhan Yu, Dawei Wang. (2018). Network Traffic Classification Using Machine Learning Algorithms. *Springer International.*
13. Yanli Liu, Yourong Wang, and Jian Zhang. (2012). New Machine Learning Algorithm: Random Forest. *ICICA, Springer-Verlag Berlin Heidelberg.*
14. Xin Liu, Xiao Liu, Yongxuan Lai, FanYang, Yifeng Zeng. (2019). Random Decision DAG: An Entropy Based Compression Approach for Random Forest. *Springer Nature Switzerland AG*.
15. Marko Ristin, Matthieu Guillaumin, Juergen Gall. (2016). Incremental Learning of Random Forests for Large-Scale Image Classification. *IEEE.*
16. Thomas Y.C. Woo. (2020). A Modular Approach to Packet Classification: Algorithms and Results. *IEEE.*
17. J. S. Rojas, Á. R. Gallón and J. C. Corrales. (2018). Personalized service degradation policies on OTT applications based on the consumption behavior of users. *In Computational Science and Its Applications- ICCSA, Cham, Switzerland:Springer, pp. 543-557.*
18. Brian Malley, Daniele Ramazzotti and Joy Tzung-yu Wu, (2016). Data Pre-processing. *MIT Critical Data, Secondary Analysis of Electronic Health Records, p. Chapter 12.*
19. Salvador García, Sergio Ramírez-Gallego, Julián Luengo, José Manuel Benítez, Francisco Herrera. (2016). Big data preprocessing: methods and prospects. *Open Access - CrossMark.*

Xử lý và phân loại dữ liệu gói tin IP trên mạng Internet dựa vào học máy

Vương Xuân Chí*, Nguyễn Kim Quốc**
Khoa Công nghệ Thông tin, Trường Đại học Nguyễn Tất Thành
*vxchi@ntt.edu.vn, **nkquoc@ntt.edu.vn

Tóm tắt Ngày nay, với sự phát triển không ngừng của công nghệ thông tin, các giao tiếp truyền thông trên mạng Internet ngày càng tăng, tình trạng nghẽn mạng trên đường truyền hay tại các nút mạng là một vấn đề đáng quan tâm. Để phát triển hệ thống hạ tầng mạng truyền thông Internet tại một thành phố lớn, một quốc gia hay trên thế giới, việc phân luồng và điều khiển luồng dữ liệu mạng để tối ưu hóa quá trình giao tiếp và giảm thiểu tình trạng nghẽn mạng là rất quan trọng và cần thiết. Trong nghiên cứu này, nhóm tác giả phân tích, xử lý dữ liệu gói tin Internet (IP), dựa vào độ trễ của gói tin sử dụng mô hình học máy với phương pháp Rừng ngẫu nhiên (RF) và mô hình Support Vector Machines (SVM) để phân loại gói tin IP. Mục tiêu chính của việc phân loại gói tin theo độ trễ để tối ưu hóa hiệu suất của mạng bằng cách ưu tiên xử lý các gói tin có độ trễ thấp, đảm bảo các dịch vụ trực tuyến như video streaming, voice calls được ổn định và ít bị gián đoạn. Hơn nữa, dễ dàng quản lý và điều khiển lưu lượng gói tin, giảm tình trạng tắc nghẽn tại bộ định tuyến mạng.

Từ khóa phân loại gói tin IP, mạng IP, nghẽn mạng, học máy, rừng ngẫu nhiên.