Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 4935108, 11 pages
https://doi.org/10.1155/2021/4935108

Research Article

Classification of Markov Encrypted Traffic on Gaussian Mixture Model Constrained Clustering

Junkai Yi,[1] **Guanglin Gong**,[1] **Zeyu Liu**,[1] **and Yacong Zhang**[2]

[1]*College of Automation, Beijing Information Science and Technology University, Beijing 100192, China*
[2]*College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100015, China*

Correspondence should be addressed to Guanglin Gong; gguanglin@126.com

Received 19 August 2021; Revised 8 September 2021; Accepted 16 September 2021; Published 7 October 2021

Academic Editor: Chi-Hua Chen

In order to solve the problem that traditional analysis approaches of encrypted traffic in encryption transmission of network application only consider the traffic classification in the complete communication process with ignoring traffic classification in the simplified communication process, and there are a lot of duplication problems in application fingerprints during state transition, a new classification approach of encrypted traffic is proposed. The article applies the Gaussian mixture model (GMM) to analyze the length of the message, and the model is established to solve the problem of application fingerprint duplication. The fingerprints with similar lengths of the same application are divided into as few clusters as possible by constrained clustering approach, which speeds up convergence speed and improves the clustering effect. The experimental results show that compared with the other encryption traffic classification approaches, the proposed approach has 11.7%, 19.8%, 6.86%, and 5.36% improvement in TPR, FPR, Precision, and Recall, respectively, and the classification effect of encrypted traffic is significantly improved.

1. Introduction

Network traffic is the data transmitted in the network. Analyzing and monitoring network traffic can allow managers to clearly know the situation of data transmission in the network [1]. In addition to providing references for network control and services, traffic behavior analysis is the basic premise of network security analysis. Therefore, traffic behavior analysis is also called "network visualization." For the needs of transmission protection and user privacy, the encryption ratio of data packets has increased sharply. This change makes traffic analysis and identification very difficult. Before further analysis of network traffic, determining the type of traffic is the basis [2].

Traffic classification is a problem that has been studied very early. In the early reference, most of the traffic was roughly divided into secure shell (SSH), virtual private network (VPN), secure socket layer (SSL), encrypted peer to peer (P2P), voice over internet protocol (VoIP), and other categories [3]. In fact, this level of classification does not have much meaning for further analysis and research in the future. Although traffic classification has been studied very early, due to the characteristics of the traffic itself and the use of encryption technology, traffic classification also faces many problems. First of all, in the network environment, the message exchange between the two parties in the traffic classification feature extraction is a continuous process. The data generated by this process does not have a typical feature description. Moreover, the type, length, and IP address of the message cannot be used directly. Traditional machine learning-based methods cannot analyze network traffic. The second is the scale of network traffic. The specific form of network traffic in the network is data packets. A data packet has a limited size and the limited data it carries. The number of data packets in the network will be very large. Therefore, analyzing and marking network traffic faces a huge challenge.

Encrypted traffic uses a special algorithm to change the original data. Even if it is intercepted during transmission, the content cannot be obtained. Before realizing data transmission in secure socket layer/transport layer security (TLS/SSL), both parties in communication need to go

through cipher and certificate exchange and user data transmission. This process is more complicated, so in the actual process, the communication process is usually simplified to improve communication efficiency. After experiments, the process is divided into three categories:

(1) The first complete communication process generally occurs when the application first applies for communication with the server. The server needs to transmit ciphers and certificates. This process is the most complete and is also the main goal of previous reference behavior analysis

(2) The simplified communication process is used in the session ID reuse phase. This process is suitable for applications to connect again in a short time. The server will reserve resources. When the application requests data, the server will directly obtain the relevant parameters from the reserved resources and will not perform the first type of complete communication. In order to ensure the reliability of the transmission, it will use a ChangeCipherSpec message to transmit the new cipher

(3) The data transmission process is mainly based on the application transmitting user data. Most applications only use transmission control protocol (TCP) message that is used to maintain the connection and Application Data message to complete the task

The existing literatures only consider the first type of situation in the communication process analysis stage and take the message type as the main analysis target. A small amount of literatures consider the relationship between message lengths. The selection of the appropriate analysis object affects the classification efficiency of the classification method.

Encrypted traffic classification methods are constantly changing, and new research results are constantly being produced. Early traffic classification mainly used protocol ports to identify traffic, such as port 21 for file transfer protocol (FTP) and port 80 for hypertext transfer protocol (HTTP). However, many applications currently use the dynamic port negotiation mechanism, which makes the port method no longer applicable. The method based on deep packet inspection (DPI) is considered to be effective and reliable for unencrypted traffic. However, with the widespread use of encryption technology, the amount of encrypted traffic has also increased significantly. Many applications are using protocol encapsulation or obfuscation techniques to circumvent network monitoring. Therefore, DPI-based methods are no longer applicable [4].

At this stage, the neural network is a research hotspot. The combination of encrypted traffic classification and the neural network has become a relatively common method. For example, Liu et al. proposed a novel traffic classification method named High Entropy DistinGuishEr (HEDGE) to distinguish between compressed and encrypted traffic [5]. Ren et al. proposed a tree-structured recurrent neural network (tree-RNN) to classify encrypted traffic, using the tree structure to divide large categories into small categories [6]. Aceto et al. proposed a novel multimodal multitask deep learning approach for traffic classification based on deep learning [7]. However, the above methods are either complicated in feature extraction or difficult in model training.

In 2014, Korczynski and Duda introduced the concept of Markov chain fingerprint recognition for the first time, using hidden Markov models for traffic classification and recognition [8]. They think that the message sequence of communication is a Markov random process, and the current state depends on the previous state. They take advantage of a sequence of message types in the SSL/TLS headers of a given application, which appears in a single-direction flow from a server to a client, to build the first-order Markov chain as a statistical fingerprint for that application. Information embedded in SSL/TLS sessions naturally forms a sequence with time-varying message types, which is analogous to the state transitions in the Markov chain. Therefore, it is reasonable to apply the Markov chain to the construction of application fingerprints. Through Markov modeling the SSL/TLS message sequence, the whole process is divided into three parts: enter probability, transition probability matrix, and exit probability.

Shen et al. [9] believed that there were some shortcomings in the research of Korczynski and Duda [8]. The first is the problem of session ID reuse. The communication between the client and the server will not reestablish the connection within a certain period of time, and there is no complete cipher and certificate exchange process. The second is the problem of the Application Data message. The existing literature only analyzes the encryption process with ignoring the transmission of the Application Data message in the network. From these two points, Shen et al. proposed the classification method of encrypted traffic with second-order Markov chains and application attribute bigrams [9]. However, there are still major limitations in their method: (1) The types of applications are gradually increasing, most of those carry out cipher and certificate interactions in accordance with the protocol, and feature fingerprints have extremely high repetitions. (2) The Certificate message occupies only a small part of the whole message, and most of the message length information is not considered. The length of information of these messages is also an important feature. (3) In the session ID multiplexing stage, the Certificate message appears rarely. Most communication is to keep the connection through TC messages and then send a large number of Application Data messages.

Chen et al. considered that the application had the problem of message duplication that affected the classification effect and proposed a multiattribute-based encrypted traffic classification system named multiattribute associated fingerprint (MAAF) [10]. Liu et al. introduced the concept of length block sequence and proposed a method named multi-attribute Markov probability fingerprints (MaMPFs) [11]. However, this method is still based on statistics and belongs to the category of machine learning. It relies heavily on feature selection and usually cannot find exact features. Chen et al. considered the differences among encryption network protocol stacks and proposed a method of encrypted

traffic service classification combining with capsule neural network in a multiprotocol environment by using multiprotocol data unit (PDU) lengths as the features, making full use of Markov property between PDU length sequences [12]. Although the feature extraction time is improved compared with the literature [11], it requires a lot of observation and analysis work in the early stage. Therefore, Yao et al. introduced the GMM and proposed a new traffic classification model based on GMM and hidden Markov models (MGHMMs) [4], which only required fewer features of interpacket time (IPT) and packet size (PS) and calculations to classify traffic.

In order to solve some of the above problems (a summary is presented in Table 1), this paper proposes the method of Classification of Markov Encrypted Traffic on Gaussian Mixture Model Constrained Clustering (MET-GCC). First, the Markov model is established by calculating the initial probability, the completion probability, and the state transition matrix to take fingerprints as a feature of traffic classification. Aiming at the problem of fingerprint duplication and neglect of a large number of messages, an N-gram model [13] based on message length is established. On the basis of the considered Certificate message, the length of other related messages is also taken as an important feature, and the GMM [4] is used to model the packet length. Finally, the method of constrained clustering [14] is proposed, and constrained conditions are added to the clustering parameters to divide the application fingerprints with similar lengths in the same application into a cluster as much as possible and calculate the distribution probability of the application fingerprints.

The MET-GCC method digs out fine-grained features from encrypted traffic, that is, the length of other related messages is also an important feature. The method solves the problem that the existing Markov network traffic classification method only analyzes the traffic classification in the complete stage of communication establishment and ignores the traffic classification in the simplified stage of communication maintenance, and there is a lot of duplication of fingerprints in the network. Through the analysis of the three types of communication processes, the message length is used as the analysis object. The GMM and the constrained clustering method are used to establish the message length model to improve classification efficiency through the distribution probability of fingerprints and realize the classification of traffic in different states.

We briefly summarize our main contributions as follows:

(1) We propose a new type of encrypted traffic classification approach—MET-GCC. By constrained clustering by Gaussian mixture model of message length, the traffic classification of different states is realized. Add constrained conditions to clustering and analyze the probability distribution of packets of different lengths in the same application to improve accuracy. Realize the analysis of the message length through constrained clustering, and improve the classification efficiency by analyzing the distribution of the length of each message

(2) MET-GCC solves not only the traditional Markov model-based network traffic classification method only analyzes the traffic classification in the complete stage of communication establishment with ignoring the traffic classification in the simplified stage of communication maintenance but also duplication problems in application fingerprints during state transition. GMM and clustering methods are introduced on the existing basis to realize the classification of traffic in different states, which effectively improves the classification effect

(3) We compared the classification performance of MET-GCC and related algorithms MCF and SOM. We also compared the classification performance of MET-GCC and the latest Markov and GMM-based MGHMM algorithm on average Precision and Recall. The experimental results show the superiority of the algorithm

The rest of the paper is organized as follows. Section 2 introduces the Markov process and message state transition. Section 3 introduces the derivation of the MET-GCC algorithm in detail. Section 4 verifies the effectiveness of MET-GCC in encrypted traffic classification through a large number of experiments and compares it with traditional and latest classification algorithms. Section 5 gives the conclusion of this article.

2. Markov Process and Message State Transition

There is a certain probability of data packet conversion in the network. This probability is related to the application. Such a random process can be described by the Markov process. The core principle of the Markov process is to include a collection of multiple states. A state transition matrix can describe the transition process. The current state is only related to the previous state and has nothing to do with other states. This is the Markov property. This process of state transition is called the Markov process.

The Markov process is an extremely ideal process, which is a high abstraction of reality, but there are two basic conditions that need to be met:

(1) The state at the current time t is only related to the previous state X_{t-1}, not related to the earlier state

(2) Markov contains at least three parts: state set, transition matrix, and initial state distribution

A set of states of the Markov process is $X_1, X_2, \cdots, X_t$, and the probability of the next moment $(t+1)$ is shown in Equation (1):

$$P(X_{t+1} = x \mid X_1, X_2, \cdots X_t) = P(X_{t+1} = x \mid X_t) = p_{t+1}. \quad (1)$$

The mobile terminal contains a large number of applications and sends a large number of data packets at all times. The acquired features are limited, and these features are not typical features. The classification algorithm in

Table 1: The comparison of existing recent traffic classification algorithms.

Algorithm	Advantages	Disadvantages
HEDGE [5]	High accuracy	Difficult in feature extraction
Tree-RNN [6]	Small classification	Difficult in feature extraction and model training
DISTILLER [7]	Overcome performance limitations of single-modality DL-based TC proposals	Difficult in feature extraction
SOM [9]	Enrich the state transitions in the Markov chain and construct more distinctive application fingerprints	Ignore the state transition of network communication and have duplicate fingerprints
MAAF [10]	Can accurately classify the applications of the same developer	Poor classification to applications with different developers but similar certificates
MaMPF [11]	High accuracy in real networks	Difficult in feature extraction
LS-CapsNet [12]	Solve difficult in feature extraction	High computation overhead
MGHMM [4]	Need fewer features and computational overhead	Constrained by encryption methods

traditional machine learning cannot be used to achieve classification, so the Markov process is a feasible method.

Different types of messages in the network can be defined as the state space in the Markov model. The current state is X_t, and the probability of the next state X_{t+1} is defined as follows:

$$P(X_{t+1} \mid X_t) = p_{t\sim(t+1)}, \tag{2}$$

$$P = \begin{bmatrix} p_{1\sim1} & p_{1\sim2} & \cdots & p_{1\sim n} \\ p_{2\sim1} & p_{2\sim2} & \cdots & p_{2\sim n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m\sim1} & p_{m\sim2} & \vdots & p_{m\sim n} \end{bmatrix}, \tag{3}$$

where $p_{i\sim j}(i, j \in T)$ is the transition probability.

In order to improve the calculation accuracy, the second-order Markov model can be used:

$$P(X_{t+1} = x \mid X_1, X_2, \cdots, X_t) = P(X_{t+1} = \mid X_t, X_{t-1}) = p_{(t-1)\sim t\sim(t+1)}, \tag{4}$$

$$P = \begin{bmatrix} p_{1\sim1\sim1} & p_{1\sim1\sim2} & \cdots & p_{1\sim1\sim n} \\ p_{1\sim2\sim1} & p_{1\sim2\sim2} & \cdots & p_{1\sim2\sim n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m\sim m\sim1} & p_{m\sim m\sim2} & \cdots & p_{m\sim m\sim n} \end{bmatrix}. \tag{5}$$

The initial probability is the probability of occurrence of an unknown sequence, so the initial probability (INIP) is defined as shown in Equation (6):

$$\text{INIP} = [\text{ip}_1, \text{ip}_2, \cdots, \text{ip}_m]. \tag{6}$$

INIP represents the probability of a message appearing, and then, an exit probability (EXTP) needs to be defined, as shown in Equation (7).

$$\text{EXTP} = [\text{ep}_1, \text{ep}_2, \cdots, \text{ep}_n]. \tag{7}$$

Assuming that a data packet sequence $\text{seq}_M = <\text{msg}_1, \text{msg}_2, \cdots, \text{msg}_M>$ is captured, the probability that it belongs to a certain application is calculated as shown in Equation (8):

$$P(<\text{msg}_1, \text{msg}_2, \cdots, \text{msg}_M>) = \text{INIP}_{\text{msg}} \times \prod_{i=2}^{l} p(i-1) \sim i \sim (i+1) \times \text{EXTP}_{\text{msg}}, \tag{8}$$

where INIP_{msg} is the probability that state msg_1 is the initial state and EXTP_{msg} is the probability that state msg_M is the exit state. The transition from the initial state to the exit state is the transition process of the message state.

3. MET-GCC Encrypted Traffic Classification

3.1. Utility Calculations of Application Fingerprints. Obtaining the most typical cipher exchange process in the communication process can more accurately describe the communication process. After a long period of experimental observation, it is found that the number of each type of message in the network is very different. The TCP three-way handshake protocol is widely available in the network, accounting for more than 80% of the total number of messages. Therefore, in addition to the difference in length, it is difficult to conduct behavior analysis through the messages generated by the TCP three times handshake protocol. Although the number of TLS-related messages is about 10%, each time an encrypted communication process is established, the communication between client and server requires a complete cipher and certificate exchange process, which has a greater impact on the analysis of communication behavior. That is, a small number of message types have strong behavior analysis capabilities, so the ability of this message to distinguish data streams is defined as utility, which can be understood as the value of the message itself. It is similar to natural language processing. There are some very frequently used words in natural language, which are used in almost every text, such as "you," "me," and "yes." Although these words have a high frequency of occurrence, they are not helpful for the next step of the algorithm,

because such words are used too often. Therefore, combine with the tf-idf algorithm [15] of evaluating the importance of words in natural language processing to define utility.

Definition 1 (Utility). In the sequence representation $s = <p_l, p_m, \cdots, p_n>$ of an N-gram model, the utility of an item p_m is defined as follows:

$$u(p_m) = \phi\left(n_{p_m}, l\right) \times \varphi\left(n_d, n_d^{p_m}\right), \tag{9}$$

where the number of messages p_m in the sequence s is n_{p_m}, the length of s is l, and the proportion of p_m in the sequence s is $\varphi(n_{p_m}, l)$, which is defined as follows:

$$\phi\left(n_{p_m}, n\right) = \frac{n_{p_m}}{n}. \tag{10}$$

The total number of fingerprints isn_d, $n_d^{p_m}$ is the fingerprint containing message p_m, and $\varphi(n_d, n_d^{p_m})$ is defined as follows:

$$\varphi\left(n_d, n_d^{p_m}\right) = \ln \frac{n_d}{n_d^{p_m} + 1}. \tag{11}$$

Suppose a fingerprint sequence is <11 : 01,11 : 03,11 : 02,23 : 2,11 : 02,23 : 3,23 : 5,23 : 8>, where the utility of the message 23 : 2:

$$u(23 : 2) = \phi(n_{23:2}, n) \times \varphi\left(n_d, n_d^{23:2}\right) = \frac{1}{8} \times \ln \frac{8}{5} = 0.575. \tag{12}$$

Definition 2 (Average utility). The average utility of the sequence $s = <p_l, p_m, \cdots, p_n>$ calculates the average utility of the entire fingerprint based on the utility of the message and is defined as follows:

$$au(s) = \frac{\sum X \in (l, m, \cdots, n) u(p_X)}{l(p_X)}. \tag{13}$$

3.2. Communication State Transition Process. The transmission of network data is a very complicated process that has a large amount of data, many redundancies, and few attributes. The analysis of encrypted traffic has always been a difficult point in research. Literature [9] analyzes the length of the Certificate and Application Data in the message to improve the classification effect and proposes bigram clustering to describe the relationship between them. Figure 1 shows the process of communication state transition described by it.

Figure 1 shows that state X represents the interaction process before Certificate, and state Y represents other types of messages between Certificate and Application Data. The Application Data message only has a difference in length. The probability distribution of this message length is determined by its bigram clustering, and the probability of the Application Data message should also be considered when calculating the probability. However, according to long-term experimental observations, there

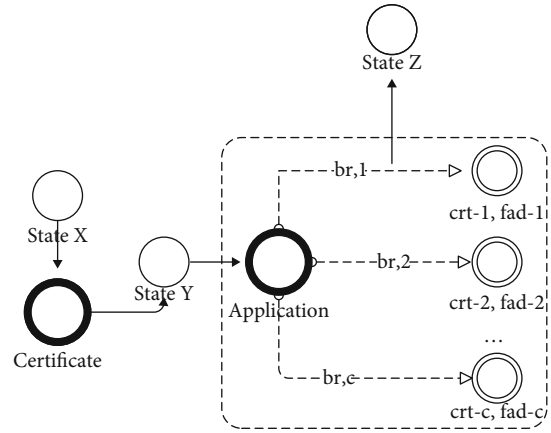

Figure 1: Communication status transition.

are some situations that are not considered in the reference. Figure 2 shows the distribution of the same fingerprint status in network traffic.

Figure 2 shows the conversion between the various states of communication and the distribution of the same fingerprint. state α is the complete process of the above three types of processes, which is juxtaposed with the simplified process of state β, and state γ is the data transmission process. And both state α and state β will convert to state γ. The Certificate message in Figure 2 only appears in state α, and after a large amount of data is transmitted, state α will not occur again for a long time, so Certificate as the core of bigram clustering is difficult to analyze state β and state γ. The other case is to find the fingerprint duplication problem in various states by calculating the average utility of the application fingerprints. As the main object of traffic behavior analysis, state α contains a large number of duplicate fingerprints, which has a greater impact on the classification results.

According to the above problems, this article takes the length of the message as the analysis target, further distinguishes the duplication case between the messages, establishes the N-gram model between the state α, state β, and state γ, and then proposes a message probability distribution model based on Gaussian mixture model constrained clustering to determine the distribution of fingerprints in each state.

3.3. Message N-Gram Model Based on Gaussian Mixture Model Constrained Clustering. The encrypted transmission message has no other valuable features except for the different message length. The length becomes the main feature for analyzing encrypted data. Because of the relative stability of developers and standards, an application generally transmits data in a relatively fixed format. This section analyzes the length-based message N-gram model, which can establish a probability distribution model for messages of different applications and different lengths. Figure 3 is a schematic diagram of the probability distribution model.

In Figure 3, in each state in the communication process, the identical application fingerprint will also have multiple situations, forming the N-gram multivariate model, and each situation will occur with a certain probability. Suppose that in a state, the length of a certain fingerprint contains n

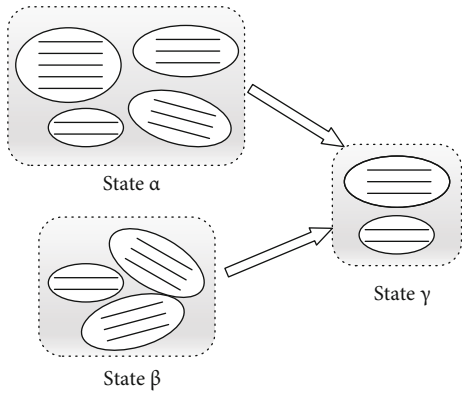

Figure 2: Fingerprint distribution.

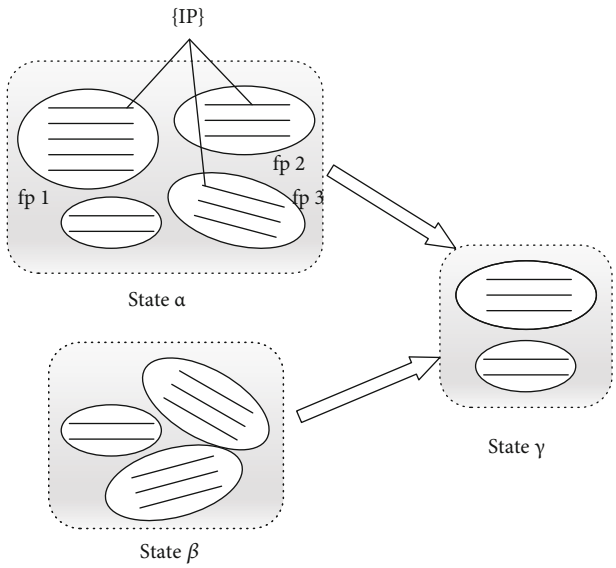

Figure 3: Schematic diagram of probability distribution model.

situations, represented by a vector $\mathrm{fp}^i = (x_1^i, x_2^i, \cdots, x_m^i, \cdots, x_n^i)$, where x_m^i is the length of a message in a fingerprint, so the characteristic fingerprint of an application is an N-gram model and a state has many kinds of N-gram fingerprints $\mathrm{fp}_{\alpha,\beta,\gamma} = (\mathrm{fp}^1, \mathrm{fp}^2, \cdots, \mathrm{fp}^i)$.

$\mathrm{fp}_{\alpha,\beta,\gamma}$ is a combination of multiple distributions whose specific distribution types and parameters are unknown. How to describe $\mathrm{fp}_{\alpha,\beta,\gamma}$ is the key to establishing a model. The GMM is a linear combination of multiple normal distributions. In theory, any kind of unknown distribution can be represented by a linear combination of multiple normal distributions. This is the GMM. Assuming that a certain application fp^i of state $\mathrm{fp}_{\alpha,\beta,\gamma}$ conforms to a normal distribution, then a Gaussian model can be used to describe fp.

$$p\left(x_{\mathrm{fp}} \mid \theta\right) = \sum_{i=1}^{n} a_i p_i\left(x_{\mathrm{fp}}^i \mid \theta_i\right), \tag{14}$$

where a_i is the mixed parameter $a_1 + a_2 + \cdots + a_n = 1$ of each distribution and θ_i is the parameter (μ_i, σ_i) of the normal distribution. Equation (9) can be transformed into the following:

$$p_i\left(x_{\mathrm{fp}}^i \mid \theta_i\right) = f\left(x_{\mathrm{fp}}^i \mid \mu_i, \sigma_i\right) = \frac{\sqrt{\sigma_i}}{\sqrt{2\pi}} e^{-1/2\left(x_{nf}^i - \mu_i\right)^T \sigma_i^{-1}\left(x_{nf}^i - \mu_i\right)}. \tag{15}$$

All the parameters in Equation (15) are $\theta = (\theta_1, \theta_2, \cdots, \theta_n, a_1, a_2, \cdots, a_n)$. Use maximum likelihood estimation to find the parameter θ. Assuming that the collected sample set is $X = (x_1, x_2, \cdots, x_n)$,

$$\log p(X \mid \theta) = \sum_{i=1}^{n} \log p(X_i \mid \theta). \tag{16}$$

Assuming that the estimated parameter value of θ is $\widehat{\theta} = (\widehat{\theta}_1, \widehat{\theta}_2, \cdots, \widehat{\theta}_n, \widehat{a}_1, \widehat{a}_2, \cdots, \widehat{a}_n)$, in order to find $\widehat{\theta}$, $\log p(X \mid \theta)$ needs to be maximized, which is given by Equation (17):

$$\frac{\partial \log p(X \mid \theta)}{\partial \theta} = 0. \tag{17}$$

Obtaining the estimated value $\widehat{\theta} = (\widehat{\theta}_1, \widehat{\theta}_2, \cdots, \widehat{\theta}_n, \widehat{a}_1, \widehat{a}_2, \cdots, \widehat{a}_n)$ of θ, the probability distribution of the same fingerprint type in the state can be obtained:

$$p_{\mathrm{fp}}(X) = \widehat{a}_1 p_1\left(x_{\mathrm{fp}}^1 \mid \widehat{\theta}_1\right) + \widehat{a}_2 p_2\left(x_{\mathrm{fp}}^2 \mid \widehat{\theta}_2\right) + \cdots + \widehat{a}_n p_n\left(x_{\mathrm{fp}}^n \mid \widehat{\theta}_n\right) + \cdots. \tag{18}$$

There are multiple fingerprints in the same state, and each fingerprint is an N-gram model, and the parameters are obtained through maximum likelihood estimation. The calculation of the parameter estimation method shown in Equation (18) is extremely complicated, and the amount of calculation is very large, which is difficult to obtain in the actual process of finding the parameters. In addition, the sample $\mathrm{fp}^i = (x_1^i, x_2^i, \cdots, x_m^i, \cdots, x_n^i)$ describes the length of the message. The same length of messages in the same fingerprint may belong to different applications. The algorithm proposed in this paper requires the probability distribution of fingerprints with messages of different lengths in a fingerprint. The clustering method divides the fingerprints with small differences into a cluster, which is used to calculate the probability distribution of the N-gram model.

The main role of clustering is to separate unlabelled data into discrete sets [16]. The traditional clustering method [17] is to randomly divide several samples into several groups, calculate the distance between each sample and the center of the class by iterative method, and redivide each class. Taking into account the characteristics of network data itself, an IP address will have multiple different fingerprint types, so the fingerprint itself can be divided into different clusters according to the IP address when collecting data. By adding restriction conditions in the parameter calculation, the fingerprints of the same application can be divided into one cluster as much as possible to speed up the

convergence speed and obtain a more accurate state transition matrix. In addition, the number of IP addresses is much larger than the number of classifications, and the obtained data packets are divided into $Y = (y_1, y_2, y_3, \cdots, y_n)$ equivalence sets through clustering.

Assuming that constrained condition (Φ_c) is given to the sample $X = (x_1, x_2, x_3, \cdots, x_n)$ in the process of finding the parameters, there is a class of division $Y = (y_1, y_2, y_3, \cdots, y_n)$, $(x_i \in y_i)$. The sample X can be divided into $X = (x_1, x_2, x_3, \cdots, x_n)$, where $X_i = (x_1^i, x_2^i, x_3^i, \cdots, x_n^i)$ is a subset of X, and the constrained condition is $\Phi_c = \{Y \mid (y_1^i = y_2^i = y_3^i = \cdots = y_n^i)\}$. Therefore, the expected function (θ, θ_y) of the parameters (θ, θ_y) is as follows:

$$F(\theta, \theta_y) = \sum_y \log p(X, Y \mid y \in \Phi_c, \theta) P(y \mid X, y \in \Phi_c, \theta_y), \quad (19)$$

where θ_y is the parameter which is added Φ_c.

Equation (19) is the expected function of the parameter (θ, θ_y), and then, the maximum likelihood is used to estimate the parameter (θ, θ_y). After expanding Equation (19), Equation (20) can be obtained:

$$F(\theta, \theta_y) = \sum_{s=1}^{M} \sum_{l=1}^{K} P(l \mid X_s, y \in \Phi_c, \theta_y) \cdot \left(N_s \log a_l + \sum_{n=1}^{N_s} \log p_l(x_n^s \mid \theta_y) \right) - \sum_{s=1}^{M} \log \sum_{l=1}^{K} (a_l)^{N_s}, \quad (20)$$

where $P(l \mid X_s, y \in \Phi_c, \theta_y)$ is the posterior probability of X_s.

Then, it can calculate sequentially the estimated value of the parameter (μ_i, σ_i) of the Gaussian model:

$$\widehat{\mu}_i = \frac{\sum_{s=1}^{M} P(l \mid X_l, y \in \Phi_c, \theta_y) \left(x_1^s + \cdots + x_1^{N_s} \right)}{P(l \mid) X_l, y \in \Phi_c, \theta_y)(N_1 + \cdots + N_M)}, \quad (21)$$

$$\widehat{\sigma}_i = \frac{\sum_{s=1}^{M} P(l \mid X_l, y \in \Phi_c, \theta_y) \left[(x_1^s - \widehat{\mu}_l)(x_1^s - \mu \wedge_l)^T + \cdots + \left(x_{N_S}^s - \widehat{\mu}_l \right) \left(x_{N_S}^s - \mu \wedge_l \right)^T \right]}{P(l \mid X_l, y \in \Phi_c, \theta_y)(N_1 + \cdots + N_M)}. \quad (22)$$

According to $(\widehat{\mu}_l, \widehat{\sigma}_l)$, the probability can be calculated:

$$P(l \mid X_l, y \in \varphi_c, \theta_y) = \frac{\left| \sigma_l^y \right|^{N_S/2} e^F}{\sum_{j=1}^{K} \left| \sigma_l^y \right|^{N_S/2} e^F}, \quad (23)$$

$$F = \sum_{n=1}^{N_S} \left[-\frac{1}{2} \left(x_n^s - \mu_l^y \right) \left(\sigma_l^y \right)^{-1} \left(x_n^s - \mu_l^y \right) \right] \left(a_l^y \right)^{N_l}. \quad (24)$$

Equation (19) requires a very large amount of calculation, where $(x_n^s - \mu_l^y)(\sigma_l^y)^{-1}(x_n^s - \mu_l^y)$ is the Mahalanobis distance [18], the more commonly used Euclidean distance [19] is selected instead, and the cluster with the smallest distance must be selected during the clustering process. Therefore, Equation (23) is simplified to the following:

$$P(l \mid X_s, y \in \Phi_c, \theta_y) = \begin{cases} 1, & \text{if } l = \arg\min \sum_{\substack{n=1 \\ l}}^{N_s} \left\| x_n^s - \mu_l^y \right\|^2, \\ 0, & \text{otherwise.} \end{cases} \quad (25)$$

According to Equation (24), the probability distribution of the identical fingerprint can be easily calculated:

$$P_{\text{fp}} = \begin{bmatrix} p_{1,1} & \cdots & p_{1,J} \\ \vdots & \ddots & \vdots \\ p_{K,1} & \cdots & p_{K,J} \end{bmatrix}, \quad (26)$$

where $P_{K,J} = \|X_J\| / \|X_1\| + \|X_2\| + \cdots + \|X_K\|$ is the probability of each application in the identical fingerprint in a cluster.

After the above detailed derivation, we propose the GCC-ETC algorithm. The specific steps are as follows:

According to the algorithm shown, Equation (8) is improved to Equation (27):

$$P(<\text{msg}_1, \text{msg}_2, \cdots, \text{msg}_M>) = \text{INIP}_{\text{msg}} \times \prod_{i=2}^{l} p_{(i-1) \sim i \sim (i+1)} \times \text{EXTP}_{\text{msg}} \times P_{\text{fp}}. \quad (27)$$

4. Experiment and Result Analysis

The dataset required for the experimental test is the network data captured by Wireshark [20] in the actual environment. The device for capturing the experimental data is five

Input: Apply fingerprint.
Step1. Set $\{fp^1, fp^2, \cdots, fp^n\}\{fp^1, fp^2, \cdots, fp^n\}; \Phi_c; K$ to the equivalent set $X_1, X_2, \cdots, X_K$ according to the restriction Φ_c set by the IP address of the data packet;
Step2. Calculate the average value $\mu_1, \mu_2, \cdots, \mu_K$ of $X_1, X_2, \cdots, X_K$;
Step3. Calculate each sample in $X_1, X_2, \cdots, X_K$ according to $l = \arg\min \sum_{n=1}^{N_s} \|x_n^s - \mu_l^y\|^2$, and generate a new partition $X_1, X_2, \cdots, X_K$;
Step4. Recalculate the mean $\mu_1, \mu_2, \cdots, \mu_K$;
Step5. Repeat step 3 until the sample mean change does not exceed the threshold;
Step6. Recalculate P_{fp}.
Output: Division of application fingerprints $X_1, X_2, \cdots, X_K$.

Algorithm 1

smartphones equipped with Android systems in the lab, and common software is installed on them. According to the classification of each application market, this article installs the 4 most commonly used types of software on smartphones, including video, news, communication, and life, such as QQ, WeChat, email clients, and news clients. After 100 data capture, the average value obtained is used as the experimental dataset which can effectively verify the clustering effect and encrypted traffic classification effect of the algorithm proposed in this paper. The specific dataset is shown in Table 2.

Dataset1 is the situation where the smartphone obtains data packets when the user does not run any installed programs. The smartphone sends very few data packets when the smartphone standstill, mainly the push of some messages and the data sent by the operating system itself. Such data is for observing the data used by nonusers and measuring the impact on the data used by users. From the overall view of the above dataset, there are only 911 data packets in 15 minutes, which is very small compared with the mixed traffic collected in 15 minutes. This part of the data situation can be ignored; Dataset2, Dataset3, Dataset4, and Dataset5 are the traffic collection situations that only run related categories of software, and Dataset6, Dataset7, and Dataset8 are mixed traffic, which is the traffic collected by running all applications. The time is 5 min, 10 min, and 15 min.

The experiment verifies the effect of the MET-GCC algorithm proposed above, which is divided into two parts. The first part analyzes the clustering effect of constrained clustering on the length of the data packet. This article adopts the traffic classification based on the Markov process. When calculating the category to which a segment of the data stream belongs, if the calculated probability of each category is similar, the classification effect will be poor, and the opposite is better. Now analyzing how to measure the clustering effect of constrained clustering according to length, the experiment considers the two extreme situations. The first extreme situation is the most ideal situation, and all packets of an application are similar in length so that all messages will be clustered in the same cluster during clustering. In this way, getting a message again of the same length can quickly determine its category. The second extreme situation is the worst situation. The message length of an application is relatively scattered. Messages of various lengths will be grouped into different clusters during clustering so that the probabilities of various categories are similar in the calculation. It is difficult to make a judgment. Therefore, from the analysis of these two extreme situations, the evaluation standard is that the same application message should be in one cluster as much as possible, and only one application should be included in one cluster. Of course, this is the most ideal situation, and it cannot exist in practice, so the experiment defines this evaluation criterion as clustering coefficient (CL-CO) [21], as shown in Equation (28).

Table 2: Dataset.

Name	Nature category	Time (min)	Number of data packets
Dataset1	Background traffic	15	911
Dataset2	Video	15	335725
Dataset3	News	15	405436
Dataset4	Communication	15	231516
Dataset5	Life	15	241128
Dataset6	Mixed traffic	5	66429
Dataset7	Mixed traffic	10	120447
Dataset8	Mixed traffic	15	181975

$$\text{CL-CO} = \frac{1}{k}\left(\sum_i \prod_k \frac{N^i}{N_k^i} + \sum_k \prod_i \frac{N^i}{N_k^i}\right), \tag{28}$$

where $\sum_i \prod_k N^i/N_k^i$ represents the distribution of the same application in a cluster and $\sum_k \prod_i N^i/N_k^i$ represents the distribution of the message length of each application in the same cluster.

The message length is fixed, and other settings are the same; the CL-CO is related to the distribution of the message length, and the CL-CO is also related to the clustering parameter selection K. The number of clusters K is their input parameter, and then, generate the vector of all cluster centers as the output. In fact, however, we are unable to determine an appropriate value of K in advance, because different applications have a high degree of overlap in packet length [9]. Therefore, enumerate K from 1 to a relatively large number (i.e., the largest K) and use the clustering method to calculate the cluster center vector for each specific

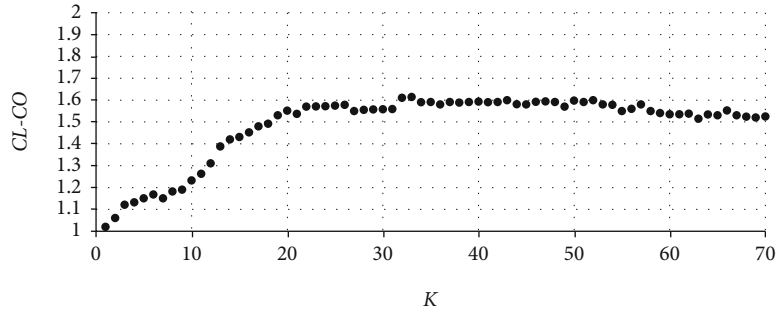

Figure 4: Clustering coefficient (CL-CO).

K. In order to compare all candidate values of K, the aggregation accuracy metric criterion CL-CO is used, and the optimal K value is determined by the value of CL-CO. The mixed traffic Dataset8 is used as the dataset for this analysis because the dataset has a large number of data packets and a long collection time. The distribution of network data packets is relatively stable, and the specific results are shown in Figure 4.

In the most ideal and worst conditions, the CL-CO is infinitely close to 0. Different collected network traffic determines the optimal K value. Therefore, the choice of K value is related to the classification effect. When the value of K is about 33, the CL-CO reaches the best. If the experiment continues to increase the K value, the CL-CO will slowly decrease.

The next experiment analyzes the effect of encrypted traffic classification. It is compared with the MCF algorithm of literature [8] and the SOM algorithm of literature [9]. These two methods are based on the traffic classification method proposed by the Markov model. The difference is that the MCF algorithm only considers the communication establishment phase. SOM is an improvement on the MCF algorithm. On this basis, the impact of different lengths of the Certificate message on the classification effect is considered. Two commonly used values are used as the evaluation criteria for the classification effect, as shown in Equations (29) and (30). TPR represents the current flow is classified into the positive sample category, and the practical positive sample accounts for the proportion of all positive samples; FPR represents the current traffic is incorrectly classified into the positive sample category and the proportion of practical negative samples to the total number of all negative samples.

$$\mathrm{TPR} = \frac{\mathrm{TP}}{(\mathrm{TP} + \mathrm{FN})}, \tag{29}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{(\mathrm{FP} + \mathrm{TN})}, \tag{30}$$

where TP is the true positive, which means that traffic belonging to the positive sample category is classified as a positive sample category; FN is the false negative, which means that traffic belonging to the positive sample category is classified as a negative sample category; FP is the false positive, which means negative. The traffic of the sample category is classified as a positive sample category; TN is the true negative, which means that the traffic of the negative sample category is classified into a negative sample category.

Table 3: MCF, SOM, and MET-GCC algorithm comparison.

Application	MCF		SOM		MET-GCC	
	TPR	FPR	TPR	FPR	TPR	FPR
Video	0.59	0.36	0.79	0.29	0.85	0.21
News	0.44	0.49	0.71	0.21	0.80	0.22
Communication	0.71	0.30	0.73	0.28	0.79	0.17
Life	0.55	0.32	0.76	0.33	0.90	0.29

Table 4: MGHMM and MET-GCC algorithm comparison.

Application	MGHMM		MET-GCC	
	Precision	Recall	Precision	Recall
Video	0.79	0.82	0.96	0.85
News	0.99	0.98	0.89	0.80
Communication	0.87	0.75	0.91	0.79
Life	0.85	0.62	0.98	0.90

The experiment here also uses Dataset8 as the tested dataset and selects the relatively better coefficient $K = 33$ for the CL-CO. First, calculate the classification situation of each category application, and the results are shown in the following Table 3.

From the analysis of the classification results in Table 3, it can be seen that the classification effect of the MCF algorithm is not very good, because the MCF algorithm does not consider the communication behavior in the communication maintenance phase. The classification performance of the SOM algorithm has been improved to a certain extent because the SOM algorithm considers the Certificate type of message in the communication maintenance phase. The MET-GCC algorithm is obviously better than these two algorithms because it also analyzes the length of other messages as an important feature in addition to Certificate messages. From the perspective of various classification situations, the more single a category of application traffic behavior is, the better the classification effect is, and the effect of video traffic classification is generally better. The reason is that it mainly sends videos, and the size and format of the video are relatively stable, while the news category contains data of multiple categories such as text and video, which has a certain impact on the classification.

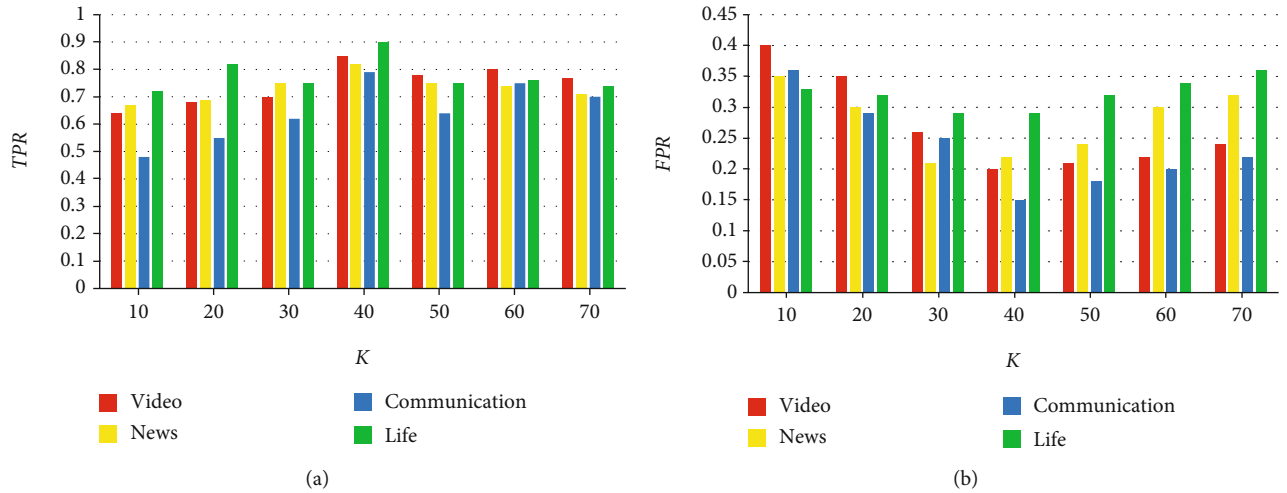

Figure 5: Comparison of TPR and FPR of different K.

Then, compared with the latest MGHMM algorithm based on Markov and GMM of literature [4], the HGHMM algorithm and MET-GCC algorithm both use Markov and GMM traffic classification methods. The difference is that the MGHMM algorithm is based on the analysis of the two traffic characteristics of IPT and PS, and the MET-GCC algorithm is based on the analysis of the message length. At the same time, add restrictions when clustering. It also uses two common values as the evaluation criteria for the classification effect, as shown in Equations (31) and (32). Precision and Recall, respectively, represent among all traffic classified as positive samples, the proportion of real positive sample category and the current traffic is classified into the positive sample category, and the practical positive sample accounts for the proportion of all positive samples.

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})}, \tag{31}$$

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})}, \tag{32}$$

where the meaning of TP, FP, and FN is the same as Equations (29) and (30).

Select the same dataset and aggregation coefficient to compare the classification conditions, as shown in Table 4.

Through comparison, it is found that the proposed MET-GCC algorithm has obvious advantages in traffic classification. The average Precision and Recall of classification have increased by 6.86% and 5.36%, respectively. It can be seen from Table 4 that the MGHMM algorithm has outstanding effects in classifying communication and news traffic, especially news. This is because the MGHMM algorithm is based on the analysis of the two traffic characteristics of IPT and PS. Communication and news traffic adopt methods like transmission interval interference and message filling, which have a greater impact on IPT and PS, and it is easy to extract features and facilitate classification. However, when a regularization method that has a small impact on IPT and PS is used to shape the traffic content, the classification effect of the algorithm is significantly reduced. The MET-GCC algorithm uses the analysis of the length of the message, and the classification will not be affected by the difference in the method. As a result, it can be seen that the MGHMM algorithm has its advantages, but the disadvantages are also obvious. The classification effect fluctuates greatly and is restricted by the method. The MET-GCC algorithm has a stable classification effect for various types of traffic, and the average classification effect is also better than the MGHMM algorithm. MET-GCC algorithm is significantly better.

In addition to the above analysis of the classification of each algorithm, the following experiment analyzes the influence of the coefficient K on the classification. From the above experiment, it can be seen that different K values have different CL-CO, and the clustering effect is also different. The experiment still uses Dataset8 to analyze the TPR and FPR for different K values. The results of the analysis are shown in Figure 5.

From Figure 5, the choice of K value in the clustering algorithm has a greater impact on the results of the experiment. Choosing an appropriate K value can increase TPR and reduce FPR. Choosing a K value that is too small or too large will affect the results of the classification experiment. Therefore, it is necessary to select a K value that guarantees a large TPR and a small FPR. It can be seen from the results that when the CL-CO is relatively high, the relative classification effect will also be improved. When $K = 40$, TPR and FPR have the best effects.

5. Conclusions

The paper proposes a new type of encrypted traffic classification algorithm—Classification of Markov Encrypted Traffic on Gaussian Mixture Model Constrained Clustering (MET-GCC). Based on the Markov model, it is through constraining clustering for the GMM of the message length to realize the traffic classification of different states. MET-GCC solves the problem that the traditional algorithm does not consider the state transition of network communication, and the fingerprint is a duplicate. The experiment proves the

effectiveness of the MET-GCC algorithm and reveals that the MET-GCC algorithm performs better than the latest encryption traffic classification algorithm based on Markov and GMM. In the future work, we plan to solve the impact of traffic complexity on the classification effect, so as to improve the classification accuracy of the MET-GCC algorithm for various types of traffic.

Data Availability

The data used to support the finding of this study are included in the article.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (U1636208).

References

[1] F. Xiao, L. Chen, H. Zhu, R. Hong, and R. Wang, "Anomaly-tolerant network traffic estimation via noise-immune temporal matrix completion model," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1192–1204, 2019.

[2] A. D'Alconzo, I. Drago, A. Morichetta, M. Mellia, and P. Casas, "A survey on big data for network traffic monitoring and analysis," *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 800–813, 2019.

[3] Z. Cao, G. Xiong, Y. Zhao, Z. Li, and L. Guo, "A Survey on Encrypted Traffic Classification," in *2014 International Conference on Applications and Techniques in Information Security*, pp. 73–81, Springer, Berlin, Heidelberg, 2014.

[4] Z. J. Yao, J. G. Ge, Y. L. Wu, X. Lin, R. He, and Y. Ma, "Encrypted traffic classification based on Gaussian mixture models and hidden Markov models," *Journal of Network and Computer Applications*, vol. 166, article 102711, 2020.

[5] C. Liu, L. He, G. Xiong, Z. Cao, and Z. Li, "Fs-net: a flow sequence network for encrypted traffic classification," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pp. 1171–1179, Paris, France, April 2019.

[6] X. M. Ren, H. Gu, and W. Wei, "Tree-RNN: tree structural recurrent neural network for network traffic classification," *Expert Systems with Applications*, vol. 167, article 114363, 2021.

[7] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "DISTILLER: encrypted traffic classification via multimodal multitask deep learning," *Journal of Network and Computer Applications*, vol. 183-184, article 102985, 2021.

[8] M. Korczynski and A. Duda, "Markov chain fingerprinting to classify encrypted traffic," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pp. 781–789, Toronto, Canada, April 2014.

[9] M. Shen, M. Wei, L. Zhu, and M. Wang, "Classification of encrypted traffic with second-order Markov chains and application attribute bigrams," *IEEE transactions on Information Forensics and Security*, vol. 12, no. 8, pp. 1830–1843, 2017.

[10] Y. Chen, T. Zang, Y. Zhang, Y. Zhou, and Y. Wang, "Rethinking encrypted traffic classification: a multi-attribute associated fingerprint approach," in *2019 IEEE 27th International Conference on Network Protocols (ICNP)*, pp. 1–11, Chicago, IL, USA, October 2019.

[11] C. Liu, Z. Cao, G. Xiong, G. Gou, S.-M. Yiu, and L. He, "Mampf: encrypted traffic classification based on multi-attribute Markov probability fingerprints," in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, pp. 1–10, Banff, AB, Canada, June 2018.

[12] Z. Chen, G. Cheng, B. Jiang, S. Tang, S. Guo, and Y. Zhou, "Length matters: fast internet encrypted traffic service classification based on multi-PDU lengths," in *2020 16th International Conference on Mobility, Sensing and Networking (MSN)*, pp. 531–538, Tokyo, Japan, December 2020.

[13] I. Lopez-Gazpio, M. Maritxalar, M. Lapata, and E. Agirre, "Word n-gram attention models for sentence similarity and inference," *Expert Systems with Applications*, vol. 132, pp. 1–11, 2019.

[14] X. Li, R. Zhang, Q. Wang, and H. Zhang, "Autoencoder constrained clustering with adaptive neighbors," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 443–449, 2021.

[15] I. Yahav, O. Shehory, and D. Schwartz, "Comments mining with TF-IDF: the inherent bias and its removal," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 3, pp. 437–450, 2019.

[16] C. L. Chowdhary and D. P. Acharjya, "Segmentation of mammograms using a novel intuitionistic possibilistic fuzzy c-mean clustering algorithm," in *Nature Inspired Computing*, pp. 75–82, Springer, Singapore, 2018.

[17] M. J. Gómez-Silva, A. de la Escalera, and J. M. Armingol, "Back-propagation of the Mahalanobis istance through a deep triplet learning model for person re-identification," *Integrated Computer-Aided Engineering*, vol. 28, no. 3, pp. 277–294, 2021.

[18] R. M. Alguliyev, R. M. Aliguliyev, and L. V. Sukhostat, "Efficient algorithm for big data clustering on single machine," *CAAI Transactions on Intelligence Technology*, vol. 5, no. 1, pp. 9–14, 2020.

[19] L. Morin, P. Gilormini, and K. Derrien, "Generalized Euclidean distances for elasticity tensors," *Journal of Elasticity*, vol. 138, no. 2, pp. 221–232, 2020.

[20] P. Narwal, D. Kumar, and S. N. Singh, "A hidden Markov model combined with Markov games for intrusion detection in cloud," *Journal of Cases on Information Technology*, vol. 21, no. 4, pp. 14–26, 2019.

[21] P. Gracar, A. Grauer, L. Lüchtrath, and P. Mörters, "The age-dependent random connection model," *Queueing Systems*, vol. 93, no. 3-4, pp. 309–331, 2020.