Unknown Protocol Identification Based on Improved K-Means++ Algorithm

Tian Feng¹, Dapeng Man² and Hao Fu²
¹Beijing-Dublin International College at BJUT, Beijing University Of Technology, No.100, Pingleyuan, Chaoyang District, Beijing, China, 100000
²Information Security Research Center, Harbin Engineering University
Email: ijitianzhengchen@163.com, {mandapeng, fuhao}@hrbeu.edu.cn

Abstract. In recent years, the gradual popularization of mobile terminals and the vigorous development of the network have spawned the birth of a new Internet structure and promoted the growth of network traffic. Behind such a large network, effective supervision of network traffic is the cornerstone of network security protection. At present, many studies on the direction of network supervision focus on the analysis of unknown network protocol types. The protocol identification method combined with machine learning is a hot topic in this kind of research. This method extracts data stream features and builds data sets, using machine learning algorithms. The model analyzes unknown network traffic and can obtain better recognition results than traditional network protocol analysis methods. Aiming at the problem of unknown traffic identification, this paper proposes a reasonable unknown traffic identification algorithm. The feature normalization preprocessing, feature selection, LOF outlier analysis, etc. are introduced. The clustering process uses the K-Means++ algorithm, and the maximum local reachable density point in the outlier analysis is used to realize the initial cluster center point. Accurate positioning.

1. Introduction
China's current Internet scale is particularly large and is developing rapidly. The extension of the Internet is essentially an extension of web pages, applications, and the data they produce. With the rise and fierceness of the mobile Internet in the past decade, the number of network applications is about to usher in new growth, and the amount of data transmission, data types, and types of data protocols on the Internet will usher in new expansions[1]. The growth of network traffic will inevitably bring new challenges to network application service quality assurance, malicious traffic detection and monitoring. Users save a lot of private information on web applications, especially in online payment, social network[2], and other applications. If this information is leaked by malicious traffic or applications, it will pose a great threat to the security of users' personal information; To avoid traffic inspection and traffic monitoring, the application also camouflages the structure and ports of its own traffic. Problems such as this have long been commonplace on the Internet.

The main traffic generated by users is mainly from sharing music, video and other files, browsing web pages on the web, sending and receiving emails, chatting in social games, videos, etc., online shopping, etc. Web pages and email content are mainly transmitted via HTTP, HTTPS, SMTP, and POP3 protocols. The HTTPS protocol involves the transmission of encrypted traffic to ensure the secure transmission of user information. In terms of file content sharing, instant messaging, streaming media services, etc., P2P and other protocols are widely used on the network to realize point-to-point traffic transmission between Internet users.
Because the K-Means algorithm has a simple process and the algorithm runs fast, especially when clustering larger data sets, the algorithm is adopted. However, the traditional K-Means algorithm has certain defects in the identification of unknown protocols, the network environment is complex, and the data flow categories are numerous, which makes the algorithm less accurate. This paper adds a departure from the K-Means algorithm. Group point analysis, with the selection point of K-Means++ algorithm and the local reachable density variable in outlier analysis, the initial cluster center point is given, and the initial cluster center point is optimized to improve the accuracy of the algorithm model. Degree, so that the clustering algorithm plays a better role in the identification of unknown protocols.

2. Related Work
In view of the identification problem of unknown network protocols on the Internet, the traditional scheme is usually analyzed from the bottom up according to the structure of a single data packet. There are many solutions for identifying traffic on the Internet. Most of today's identification schemes still use more traditional identification methods. The widely used methods include port-based traffic identification technology; load-based traffic identification technology; machine learning based Traffic identification technology.

2.1. Port-based Traffic Identification Technology
The Internet Digital Distribution Authority IANA is responsible for defining the ports of the transport layer protocol public service. Traditional protocols often register their own port numbers on the IANA official website. IANA specifies that port number 0-1023 identifies "known port", 1024-49151 identifies "registered port", and 49152-65535 identifies "dynamic or dedicated port." In the current common protocol of the Internet, port 80 is used for HTTP protocol, port 21 is used for FTP control packets, and port 22 is used for SSH protocol. The early P2P protocol also used a fixed port[3], such as the BitTorrent protocol using port 6881-6889. For these protocols using fixed ports, the upper layer application protocol can be analyzed by analyzing the packet header layer by layer and passing the port field in the header at the transport layer.

2.2. Load-based Traffic Identification Technology
In order to improve the recognition accuracy, many studies focus on the structural characteristics of application data, and identify unknown traffic through regular expressions, automata and other methods. According to this idea, the high-level data traffic of the data must be structured data, and the data has a fixed flag. Even if the data traffic uses a random port, the data structure of the application high-level is unified. In this way, the fixed flag portion of each data packet can be compared and judged by the empirically defined flag bit. If the structural feature is successfully matched, the corresponding unknown traffic protocol type can be determined.

2.3. Machine Learning Based Traffic Identification Technology
In recent years, machine learning technology has gradually emerged and plays a key role in all aspects. Many researchers have found that because machine learning works well in text classification, the concepts and ideas of machine learning can also be applied to the detection and classification of data traffic. By identifying the unknown traffic with the help of machine learning, the workload of traffic identification and analysis can be effectively reduced.

Many studies have shown that combining machine learning algorithms with unknown protocol identification yields better recognition[4]. Clustering algorithms based on unsupervised[5], supervised[6], semi-supervised[7] are also widely proposed and practiced.

3. Algorithm
Based on the analysis of the characteristics of network traffic, the original data is processed to establish the data set, and the features are optimized by feature engineering. Based on the K-Means algorithm, an unknown network protocol recognition algorithm model is established. According to the algorithm model. Processing of unknown network protocol data sets.
The analysis of feature construction, normalization preprocessing, feature selection and redundant feature dimension in order to avoid the error of the clustering index biased to the branch due to the different dimension of the feature; the basis of feature engineering In the above, the LOF algorithm is used to remove the outliers, and the initial clustering center point is selected according to the local reachability density parameter in the LOF algorithm; the algorithm uses the K-Means++ algorithm, which reduces the initial clustering point selection. In the randomness, the selection of the initial cluster center point is more scientific and effective than the K-Means algorithm. Finally, the proposed unknown protocol identification model is tested and experimented, and the protocol identification algorithm model proposed in this paper is K-Means/ The K-Means++ algorithm compares the clustering results of unknown network protocols.

3.1. Improved Algorithm Introduction
In the process of feature engineering and improved algorithm, it is necessary to use the outlier algorithm to detect and remove data for the characteristics of the network data set due to various network and human factors, such as the flow duration and the large difference in flow state. Concentrated outliers to reduce the impact of this factor on the clustering effect. Outliers can cause deviations from the cluster center, and even the outliers selected as cluster centers can lead to erroneous clustering results. Removing the outliers can improve the accuracy of the clustering algorithm model. At the same time, the local reachable density of the intermediate parameters in the outlier detection algorithm can play a guiding role in the initial cluster center point selection of the clustering algorithm.

Feature construction refers to the observation and analysis of network datasets to obtain a set of physical meanings. For the network data stream, the feature construction needs to be able to describe the characteristics of a stream from various aspects, in order to improve the accuracy of the clustering algorithm for identifying unknown protocols as much as possible. Different data streams have certain differences in terms of stream duration and data interaction due to their different execution functions. These differences are mainly divided into three categories: data size, number of data packets, and interval time.

The feature needs to describe the complete characteristics of a flow from setup to completion. These characteristics are expressed in the size of the uplink and downlink packets, the number of transmissions; the difference in the size and number of packets in the opposite direction; the upstream and downstream packets are in the Characteristics such as the interval between adjacent packets during transmission. In order to fully demonstrate the difference of data, this paper will find the maximum, minimum, mean, standard deviation and so on for all eigenvalues.

The data sets constructed according to network traffic differ in the dimensions of the different dimensions of the above features, and the values vary greatly. Since the K-Means++ algorithm is used in this paper, the Euclidean distance between samples needs to be calculated in the algorithm process, and the distance calculation is affected by the eigenvalue range. Therefore, after the feature is constructed, the data needs to be normalized in the preprocessing stage. If the feature has a single value, does not diverge, and has a low correlation with the target, this feature should be removed. Therefore, after the feature is established, the feature selection operation should be performed according to the corresponding algorithm[8].

The first step of the algorithm model is to process the data packets in the network data, extract the complete data stream, and construct a data set according to the data stream extracted by the feature processing, and perform a chi-square test for different protocol possibilities, and calculate each The chi-square test value of a feature is used to query the threshold value table of the chi-square distribution according to the chi-square test value, and remove the features of the chi-square test result as untrustworthy, and the optimized features obtained as shown in Table 1 are shown in Table 1.
Table 1. Selected feature.

| Serial number | Feature name   | unit  | description                                      |
|---------------|----------------|-------|-------------------------------------------------|
| 1             | maxUpLength    | Byte  | Maximum length of upstream packets              |
| 2             | meanUpLength   |       | Average length of upstream packets              |
| 3             | stdUpLength    |       | Upstream packet length standard deviation       |
| 4             | totalUpLength  |       | Total upstream packet size                       |
| 5             | maxDownLength  |       | Maximum length of downstream packets            |
| 6             | meanDownLength |       | Average length of downlink packets              |
| 7             | stdDownLength  | Byte  | Downstream packet length standard deviation      |
| 8             | totalDownByte  |       | Total downlink packet size                       |
| 9             | totalByte      |       | Total uplink and downlink packet size            |
| 10            | upNumber       | Number| Number of upstream packets                       |
| 11            | upBumberRatio  |       | Upstream packet ratio                           |
| 12            | downNumber     | Number| Number of downstream packets                     |
| 13            | downNumberRatio|       | Downlink packet ratio                           |
| 14            | totalNumber    |       | Total number of packets                         |
| 15            | maxUpInterval  | Microsecond | Maximum upstream packet interval    |
| 16            | meanUpInterval |       | Upstream packet average interval                 |
| 17            | stdUpInterval  | Microsecond | Upstream packet interval standard deviation |
| 18            | maxDownInterval|       | Downstream packet maximum interval              |
| 19            | meanDownInterval|     | Downstream packet average interval              |
| 20            | stdDownInterval|       | Downstream packet spacing standard deviation    |

After the normalization process and feature selection, the outliers can be analyzed. In a real network environment, the same protocol flow is affected by many network environment factors such as time of occurrence, network bandwidth, etc., even if the same function is performed, for example, opening the same web page operation may also have completely different features. When the network environment is good, the data packets arrive in order, the webpage is opened, and the data uplink and downlink can be successfully completed. When the network environment is poor, the packet loss situation may occur frequently, so the same is true.

In terms of traffic, the amount of data transmitted, the number of packets will increase, and even the long-term client, the server does not respond, the client suddenly shuts down and other factors cause the stream to terminate the data without ending. Send. The LOF outlier algorithm will exclude some sample points representing this type of stream, which may affect the accuracy of the clustering results. The flow of the LOF algorithm is shown in Figure 1.

In the LOF outlier analysis, the local reachable density needs to be calculated for each point. If the local reachable density is larger, the more sample points near the position of the sample point, the higher the density of the points. It is more likely that it is the center point of a certain cluster. In view of this conclusion, this paper makes some improvements to the K-Means++ clustering algorithm. When using the LOF outlier analysis, the improved version records the sample point with the highest average reachable density during the outlier analysis as the K to be used. The initial clustering center point of the Means++ algorithm, and this method is used instead of the K-Means++ algorithm to select random points. If there are sample points of the same density in the data set, the centroids of all sample points of the same density are obtained, and the sample points closest to the centroid are selected as the initial cluster center point of the K-Means++ algorithm. The specific process of the K-Means++ algorithm is shown in Figure 2.
**Figure 1.** LOF outlier detection process.

**Figure 2.** K-Means++ algorithm flow

**Figure 3.** Improved K-Means++ Co-identification algorithm complete process.
3.2. Improved Algorithm Introduction

In this paper, the above algorithm idea is used to identify the unknown network protocol. The overall algorithm model will be composed of feature engineering, outlier detection and K-Means+++ clustering algorithm. Based on this, the algorithm will be improved to improve the recognition accuracy of the algorithm. The specific process of the overall algorithm framework is as follows.

The flow of the algorithm framework is as follows:

1. Feature construction, constructing a data set, where n is the number of samples and m is the feature dimension;
2. Normalize the dimensions of each feature and then select features. Obtaining a data set optimized by feature engineering;
3. Perform LOF outlier analysis and analyze the outliers of each point. Sample points with outliers greater than 1 may affect the experimental results, and remove these sample points from the data set;
4. Record the sample point with the largest local reachable density parameter obtained by the LOF outlier algorithm in the previous step, which is the initial cluster center point of the K-Means+++ algorithm, and start the clustering process by inputting the data set into the algorithm.;
5. Get the final result of the clustering algorithm, that is, K protocol stream categories.

The detailed flow chart of the algorithm is shown in Figure 3:

4. Experiment

4.1. Experiment Environment

The computer used in the experiment is a virtual machine established on the real server, and the test program runs on the virtual machine. The specific configuration of the virtual machine is shown in Table 2.

This experimental program is written in Python. Based on Python 3.5, the program runs on the server. This experiment uses Tcpdump to capture some data packets in the campus network. This part of the data is used for data stream feature analysis and ciphertext feature analysis, packet capture. The network topology is shown in Figure 4.

| Device name       | virtual machine                                      |
|-------------------|------------------------------------------------------|
| operating system  | Intel(R) Xeon(R) Platinum 8163 CPU @ 2.50GHz         |
| CPU               | 2GB                                                  |
| RAM               | CentOS7.1                                            |
| Network card      | Intel Gigabit LAN                                    |

Figure 4. Experimental network topology.

4.2. Experiment Environment

This section will further verify the effectiveness of the improved algorithm model for the identification of unknown protocols through experiments.

In this paper, the experimental network dataset comes from many sources, including the WIDE public network dataset provided by MAWI Lab[9] and the data packet collection captured by Tcpdump on the campus network server. In these datasets, the dataset is more popular. The eight
application protocols evaluate the algorithm framework proposed in this chapter. These eight protocols are HTTPS, SSH, SMTP, DNS, FTP, BT, ARP, and ICMP. The feature set is first constructed for the training set to construct feature F.

For the protocol of single packet transmission, the analysis is constructed in units of data packets; since the UDP stream is byte oriented and there is no complete flow state, a flow is defined according to the timeout mechanism. The timeout period of this experiment is defined as 60 seconds, when UDP After the last packet is sent for 60 seconds, there is still no next packet, indicating that the current UDP stream ends, and a sample data is constructed through a UDP stream. The completion of the TCP flow is completed as the completion of the three-way handshake. The end of the flow is marked by the completion of the RST packet or TCP four times. This entire TCP stream is a sample of data. In order to reduce the clustering accuracy problem caused by the uneven distribution of samples, the number and proportion of each stream are not too different.

In this experiment, a total of 5 data sets were selected, and 5 experiments of unknown protocol identification were performed by cross-validation. The training set of each experiment consisted of 4 data sets, and the remaining 1 data set was used as a test set to 5 The test results are used as indicators for evaluating the algorithm model method. The specific idea is shown in Figure 5.

![Figure 5. Cross validation.](image)

The data in the five data sets is 6000, of which HTTPS, SSH, and BT accounts for 16.7%, and the remaining agreements each account for 10%. The number and ratio of experimental data sets are shown in Table 3.

**Table 3. Experiment environment.**

| Protocol type | Data Set 1~5 Number | Proportion(%) |
|---------------|---------------------|---------------|
| HTTPS         | 1000                | 16.7          |
| SSH           | 1000                | 16.7          |
| SMTP          | 600                 | 10.0          |
| DNS           | 600                 | 10.0          |
| FTP           | 600                 | 10.0          |
| BT            | 1000                | 16.7          |
| ARP           | 600                 | 10.0          |
| ICMP          | 600                 | 10.0          |
| Sum           | 6000                | 100           |
According to the above analysis, the type of integrated protocol is set to 8 for the improved algorithm. For the comparison experiment, the input of the K-Means and K-Means++ clustering algorithm is also set to 8, which can enhance contrast and better. Verify the effectiveness of the improved algorithm on unknown protocol identification.

The specific experimental scheme is as follows:

(1) Feature engineering for feature F uses scikit-learn processing. scikit-learn contains rich feature processing methods, including data preprocessing, feature selection, and dimensionality reduction. Data preprocessing is performed using the preprocessing library in scikit-learn, and the MinMaxScaler class inside the library normalizes the data. Then, according to the SelectKBest class in the feature_selection library, the chi-square test class chi2 is used for feature selection. The specific idea of the chi-square test is that for each feature, according to the different value range of the number of packets, length and time, the chi-square test is performed for different protocol possibilities, and the chi-square test value of each feature is calculated separately, according to the card. The square test value queries the threshold value table of the chi-square distribution, and removes the features of the chi-square test result that are unreliable.

(2) The LOF outlier analysis is performed on the training set, and the outlier database is analyzed using the estimator library. The outliers are calculated for each sample point, and the outliers of all sample points are recorded in an array, and the sample points represented by the part of the array whose outliers are far more than 1 are removed;

(3) and then carry out a 5-fold cross-validation, this step is processed using the model_selection library, and each verification process is a complete clustering process;

(4) The processed training set is used as the input of the K-Means++ algorithm, and both the algorithm and the control experiment use the cluster class. In the improved algorithm, the sample with the highest local reachable density obtained in the second step is defined as the initial cluster center point of the K-Means++ algorithm, and is input into the algorithm together with the selected value 8 for training;

(5) After the training is completed, the data of the untrained set can be processed by the same feature engineering to obtain the test set, and the test set is input into the algorithm model for clustering;

(6) Record the recall rate and accuracy rate of HTTPS, SSH, SMTP, DNS, FTP, BT, ARP, and ICMP during each test, and calculate the F1 value.

4.3. Experimental Results and Analysis

In the five rounds of experiments, the accuracy and recall rate of each experiment were recorded, and the average of 5 experiments was taken as the final individual accuracy rate recall rate and individual recall rate. The detailed comparison data of the improved algorithm and the comparison algorithm were as follows. Table 4 shows.

| Protocol | STD | K-Means(%) | K-Means++(%) | Improved Algorithm(%) |
|----------|-----|------------|--------------|-----------------------|
|          | precision | recall | F1 | precision | recall | F1 | precision | recall | F1 |
| BT       | 83.01 | 82.83 | 82.92 | 85.93 | 85.20 | 85.56 | 89.16 | 88.87 | 89.01 |
| HTTPS    | 84.86 | 84.50 | 84.68 | 87.28 | 87.10 | 87.19 | 89.94 | 89.98 | 89.95 |
| SSH      | 86.06 | 85.20 | 85.62 | 88.07 | 87.40 | 87.73 | 90.37 | 89.40 | 89.88 |
| FTP      | 84.69 | 83.80 | 84.24 | 87.04 | 86.30 | 86.67 | 89.47 | 88.72 | 89.09 |
| SMTP     | 83.47 | 83.67 | 83.56 | 86.85 | 86.67 | 86.76 | 89.85 | 89.32 | 89.58 |
| ICMP     | 84.71 | 84.50 | 84.60 | 86.89 | 87.00 | 86.94 | 90.06 | 88.85 | 89.45 |
| DNS      | 84.81 | 84.16 | 84.48 | 86.75 | 87.50 | 87.12 | 90.10 | 88.83 | 89.46 |
| ARP      | 85.73 | 85.16 | 85.44 | 87.03 | 86.50 | 86.76 | 89.51 | 90.07 | 89.78 |
According to the above table data, according to the number of each protocol in the data set to obtain the overall accuracy, recall rate and F1 value, the overall accuracy, recall rate and F1 value comparison chart of different algorithms are shown in Figure 6.

It can be seen from the above table and the above figure that the K-Means++ algorithm has already optimized the K-Means algorithm to a certain extent, so it will be improved compared with the K-Means algorithm in the evaluation criteria of this experiment. The algorithm is superior to the direct use of K-Means and K-Means++ algorithms in various evaluation criteria, which proves that the proposed algorithm model has better effect on unknown protocol identification.

The average accuracy of the five rounds of experiments with different protocols is shown in Figure 7.

The average recall rate of 5 rounds of experiments with different agreements is shown in Figure 8.

The average F1 values for the five rounds of experiments with different protocols are shown in Figure 9.

The improved algorithm for clustering results of each location network protocol is superior to K-Means and K-Means++ algorithms in all evaluations. Among the various types of applications, the three algorithms are the most accurate for the identification of the SSH protocol. This is because the SSH protocol is different in terms of the functions of the protocol and the HTTPS and FTP protocols. The interaction between the client and the server is stronger. The difference in the dimensions of the features is more obvious; while the recognition rate of the three algorithms is lower than that of the BT, the main reason is that since BT is in a stream, there are more hosts involved, and there are more possibilities for the dimensions of the features. Therefore, there are some similarities between the protocol and the HTTPS, FTP, and SMTP protocols in various dimensions. The ARP, DNS, and ICMP protocols have small gaps in stream duration, packet length, and packet similarity time, and there will be some interference in the clustering process.

The above results show that the improved algorithm has improved in various evaluation criteria after extracting the effective information of the data set using feature engineering, removing the noise and optimizing the selection of the initial clustering center point. Among different types of protocols,
HTTPS, SSH, BT, SMTP, and FTP protocols respectively represent a type of application protocol, and ARP, ICMP, and DNS are added to the data set to simulate the real network environment. In a real network environment, in addition to application protocols, there are some protocols that maintain links and other functions. These protocols do not include application data, and their tasks are simple, but they are indispensable in a real network environment. It can be known from the experimental results that the algorithm is feasible in the actual network environment. After receiving a series of unknown class data packets, the algorithm can reliably cluster unknown network protocols.

5. References
[1] Feng C, Pan D, Wan J, et al. Data mining for the Internet of Things: literature review and challenges[J]. International Journal of Distributed Sensor Networks, 2015, (2015-8-30), 2015, 2015(9):12.
[2] Fu Y, Wang W, Fu H, et al. Privacy Preserving Social Network Against Dopv Attacks[C]//International Conference on Web Information Systems Engineering. Springer, Cham, 2018: 178-188.
[3] Alderson, Chang, Roughan, et al. The many facets of internet topology and traffic[J]. Networks & Heterogeneous Media, 2017, 1(4):569-600.
[4] Zheng J. An association analysis and identification for unknown protocol of bitstream oriented[J]. Concurrency & Computation Practice & Experience, 2016, 28(15):4067-4081.
[5] Jie Z, Li J. Feature identification of unknown protocol[C]// International Computer Conference on Wavelet Acti ve Media Technology & Information Processing. 2017.
[6] Alan H F, Kaur J. Can Android Applications Be Identified Using Only TCP/IP Headers of Their Launch Time Traffic?[C]// Acm Conference on Security & Privacy in Wireless & Mobile Networks. 2016.
[7] Hua L, Zheng T. Research on classification arithmetic of unknown protocol space data[C]// International Conference on Computer Science & Service System. 2011.
[8] Cao T T, Zhang M, Andreae P, et al. Bagging and Feature Selection for Classification with Incomplete Data[J]. 2017.
[9] WIDE MAWI WorkingGroup. Traffic Trace Info(2018)[EB/OL]. http://mawi.wide.ad.jp/mawi/samplepoint -F/2018/