Mobile Application Identification based on Hidden Markov Model

Xinyan Yang[1,*], *Yunhui* Yi[1], *Xinguang* Xiao[1], and *Yanhong* Meng[1]

[1]State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, China

Abstract. With the increasing number of mobile applications, there has more challenging network management tasks to resolve. Users also face security issues of the mobile Internet application when enjoying the mobile network resources. Identifying applications that correspond to network traffic can help network operators effectively perform network management. The existing mobile application recognition technology presents new challenges in extensibility and applications with encryption protocols. For the existing mobile application recognition technology, there are two problems, they can not recognize the application which using the encryption protocol and their scalability is poor. In this paper, a mobile application identification method based on Hidden Markov Model(HMM) is proposed to extract the defined statistical characteristics from different network flows generated when each application starting. According to the time information of different network flows to get the corresponding time series, and then for each application to be identified separately to establish the corresponding HMM model. Then, we use 10 common applications to test the method proposed in this paper. The test results show that the mobile application recognition method proposed in this paper has a high accuracy and good generalization ability.

1 Introduction

Smart phones and other mobile devices has become an indispensable part of people's lives. Researches from the firm IDC statistics [1] and StatCounter [2] has proved that internet traffic of the mobile device has exceeded the traditional desktop device. The explosive growth of the number of mobile application has brought great challenge to network management, such as, security issues.

Mobile application identification has vital significance to network security and network management. Traditional applications (traffic) identification technology can be divided into three categories: port based methods, payloads based methods and statistical features based methods. Mobile application identification needs to consider applications number, layer protocol, Content Delivery Network, cloud service and so on. But for now the method have poor scalability and the speed of application identification can be improved.

Traditional traffic identification can be divided into two categories: payload features based and statistical features based. As most of the mobile applications using HTTP

* Corresponding author: 2523242271@qq.com

protocol for network communication, application-layer header fields are commonly used as payload features. Wei et al. [3] and Xu et al. [4] proposed method to analyze and identify Android application. However, this method is valid only for a limited application. [5~7] proposed methods extracts different traffic feature to analyze and identify the Android applications. Although these recognition methods for mobile applications can obtain better accuracy, encryption protocol communication mode and violation of user privacy are problems faced by these methods. So Wang et al. [8] studied more detailed identification of the different mobile applications and use Random Forests to train classifier. Similarly, Conti et al. [9, 11] proposed a system framework to infer which particular actions are executed in application on user's device, and gain 95% accuracy. Park et al. [10, 12] used the same method with Conti to identify online chat mobile application. They extract behavioral characteristic based on data packets and combined three kind of classification algorithms: *Naive Bayes*, *AdaBoost* and *Decision Tree* to build classifier.

Saltaformaggio et al. [13] proposed NetScope, identifying fine-grained user activity of different mobile applications, but the accuracy is only 78.04%. Fu et al. [14] focused on identifying various use types of instant messaging applications. They use *Hierarchical Clustering* and threshold heuristic methods to divide network traffic into session and dialogs type what they define. Then *Random Forests* is used to train classifier. As a result, the accuracy of classifier is 95%. Alan et al. [15] tries to use TCP header information, data packet length feature and three existing methods and gain 88% accuracy. Taylor et al. [16] focused on the identification of encrypted traffic, they use *SVM* and *Random Forests* to build classification model respectively with 18 statistical features and gain 98% accuracy.

Based on the existing research technology, this paper proposes a mobile application traffic identification based on hidden Markov model for the shortcomings of existing research. Our approach can handle encrypted network traffic by extracting the statistical characteristics of multiple network traffic. We combine the statistical and temporal characteristics of multiple network flows and establish the corresponding models for different mobile applications. When identifying new applications, we only need to write a new model for the new application.

The remainder of this paper is organized as follows. In the second part we discuss the definition of HMM. In the third part, we elaborate on the application identification process presented in this paper. In the fourth part, we present the performance evaluation of our method. Finally, we give brief conclusions.

2 Hidden markov model

Hidden Markov Model (HMM) [17, 18] is one of the basic models in natural language processing, which has wide range usage. HMM is a probabilistic graphical model for time series data modeling, which is consist of a sequence of hidden states has the Markov property and cannot be observed, an observing sequence corresponding to hidden sequence.

A HMM model can be described by a five-tuples (Q,V,π,A,B) corresponds to the Markov model of network traffic. Collection of hidden states Q represents all the hidden state. Collection of observation states V corresponding to Q, every observation state has a certain probability with different hidden states. State transition probability matrix A represents transference between hidden states. The observation probability matrix B represents probabilities of every observed value corresponding to every hidden state. The initial state probability vector π represents initial probability distribution of the hidden state.

3 Application identification

3.1 Model training

Mobile application identification method presented in this paper is based on the modeling of network traffic that is generated when the application starts. A combination of statistical and temporal features of multi-network flows is used in this paper. We divide network traffic within time generated by application starts into one group. After statistical analysis of different applications' network traffic within 60s after applications starting, we choose to extract first 20 network flows according to the start time of network flow and then extract flow characteristic information from each of the network flow. In order to reduce the influence of network noise data of connection failures and other problems, we ignore the network flow that packet number less than 7, because theoretically speaking, the smallest complete TCP network stream includes TCP three-way handshake packets (3 packets), request packets, response packets and ACK packets. Then, three characteristics are calculated for each network flow, the average bytes of packet (BYTES_PER_PKG), the average bytes of the upstream packet (BYTES_PER_UP_PKG), and the average bytes of the down packet (BYTES_PER_DOWN_PKG).

Network traffic characteristics Difference for different applications leads to the different number of hidden states for the HMM model. In this paper, the number of hidden states is determined by clustering analysis of the network flow of the application. Hierarchical clustering is designed to build hierarchical clusters. This clustering method has an obviously advantage, effective distance metric. We use a condensed hierarchical clustering [17], at the beginning, each sample is an initial cluster, and then find the nearest two clusters to merge, the hierarchy moves upwards. In order to determine which clusters should be merged, you need to specify the distance between the cluster and the link criteria. The key of hierarchical clustering is how to calculate the distance between clusters, that is, to specify the distance and the link criteria. In order to measure the time series distance, we use the dynamic time warping method [19] (DTW).

The HMM model corresponding to each mobile application can be uniquely determined by $\lambda \sim (A, B, \pi)$. We use Baum-Welch (BW) algorithm to learn these three parameters π, A and B. We obtain that:

$$Q(\lambda,\bar{\lambda}) = \sum_I \left(\ln P(O,I \mid \lambda)\right) P(O,I \mid \bar{\lambda}) = \sum_I \ln \pi_{i_1} P(O,I \mid \bar{\lambda}) + \sum_I \left(\sum_{t=1}^{T-1} \ln a_{i_t i_{t+1}} \right) P(O,I \mid \bar{\lambda}) + \sum_I \left(\sum_{t=1}^{T} \ln b_{i_t o_t} \right) P(O,I \mid \bar{\lambda}) \quad (1)$$

By maximizing Q can obtain the parameters A, B and π. We get:

$$\pi_i = \gamma_1(i) \Big/ \sum_{i=1}^{N} \gamma_1(i), \quad a_{ij} = \sum_{t=1}^{T-1} \xi_t(i,j) \Big/ \sum_{t=1}^{T-1} \gamma_t(i), \quad b_{ik} = \sum_{t=1, o_t = v_k}^{T} \gamma_t(i) \Big/ \sum_{t=1}^{T} \gamma_t(i).$$

For the observation probability matrix B, the observed state is a continuous numerical variable that follows the three-dimensional normal distribution, so B is no longer represented by b_{ik} ,but by the mean variable $\mu_i = \sum_{t=1}^{T} \gamma_t(i) o_t \Big/ \sum_{t=1}^{T} \gamma_t(i)$ and the variance matrix $\delta_i = \sum_{t=1}^{T} \gamma_t(i)(o_t - \mu_i)(o_t - \mu_i)^T \Big/ \sum_{t=1}^{T} \gamma_t(i)$.

3.2 Traffic classification

When the mobile application is identified, the sample data (observation state sequence) to be recognized is input to the HMM model corresponding to each mobile application, and the posterior probability of the sample data in each HMM model corresponding to the mobile application is respectively measured. Then, the decision principle is chosen to select the recognition result of the mobile application corresponding to the HMM model with the highest probability of posterior probability, then obtain the corresponding mobile application recognition result. The forward algorithm or backward algorithm is used for posterior probability.

According to the Bayesian decision principle, the HMM model corresponding to the maximum probability of posterior probability is selected. As the application category of the sample to be identified, and the posterior probability of the HMM model corresponding to the mobile application is obtained by the forward algorithm or the backward algorithm. Assume that there are n HMM model corresponds to the mobile application, noted λ_i, $i = 1, 2, \cdots, n$. The sample to be identified is an observation sequence O^m with the length m, which is composed of m network streams.

$$\lambda_{\text{best}} = \underset{\lambda_i \in \{\lambda_1, \lambda_2, \ldots, \lambda_n\}}{\arg\max} P(O^m \mid \lambda_i) \tag{2}$$

4 Performance evaluation

We use three different Andrew smartphone devices as the mobile application platform and connect to the PC side to receive the ADB instructions for PC-side transmission.

Table 1. Sample Information

Number	Name	Number of eigenvector		
		$Device_1$	$Device_2$	$Device_3$
App1	com.baidu.BaiduMap	166	65	35
App2	com.eg.android.AlipayGphone	144	55	30
App3	com.netease.cloudmusic	141	51	33
App4	com.netease.newsreader.activity	150	50	30
App5	com.ss.android.article.news	150	57	30
App6	com.taobao.taobao	153	61	31
App7	com.tencent.mm	180	50	35
App8	com.tencent.mobileqq	148	50	33
App9	com.UCMobile	153	54	35
App10	com.youku.phone	150	58	34
Sum		1535	551	326

The PC devices install USB wireless network cards to provide network access services for mobile device. $Device_1$ is Nexus 4 with Android 4.4.4, $Device_2$ is Nexus 5 with Android 5.1.1, $Device_3$ is One Plus with Android 6.0.1. The PC runs Wireshark software to capture the real-time network traffic data from USB wireless network card. The resulting samples are shown in Table 1.

We used the package manager to import the HMMWeka provided in [20] into Weka. The HMM classifier can be used to classify the sequence data represented as a relational attribute in Weka. The data instance must have a nominal type attribute, and a relational sequence attribute. Since we extract the eigenvectors are multivariate numerical properties, we use the Gaussian HMM model.

Using the corresponding eigenvector of $Device_1$ as the training data set, using Ten-fold cross-validation, the final result is shown in Fig. 1.

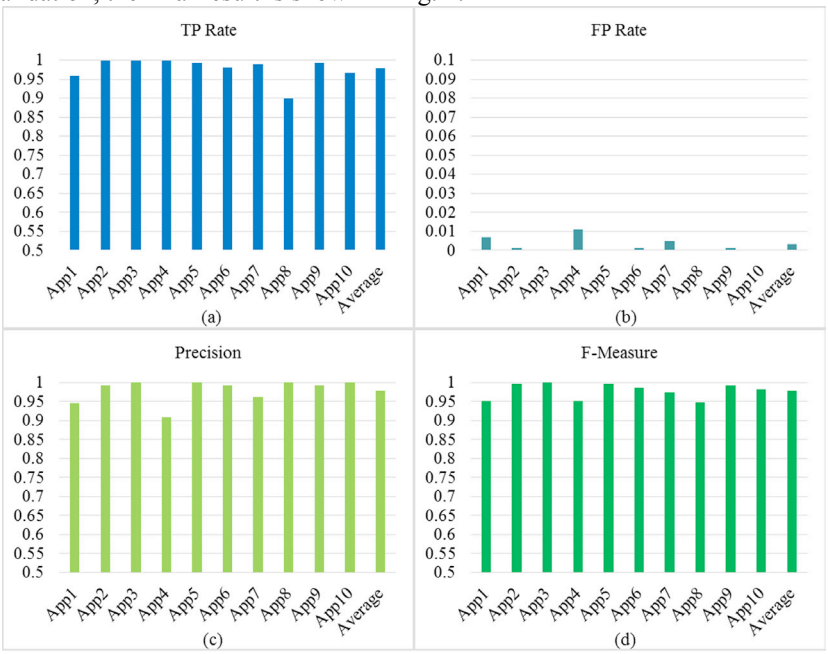

Fig. 1. Identify the accuracy rate results

In Fig. 1, the average accuracy of the classification recognition is 97.9 %, the average recall rate is 97.8%, F measure is 0.978, and all achieve good results. From the various categories of applications, the experiment of the application App4 (NetEase news) with the minimum accuracy that 90.9%, but the recall rate is 1.00. By observing the recognition results confused matrix display, NetEase news's 150 samples all correctly identified, however, 7 Baidu map samples, 4 excellent cool samples, 3 mobile QQ samples, one WeChat sample were wrongly classified as NetEase news.

In order to verify the generalization ability of the method, we use the eigenvector corresponding to $Device_1$ as the training data set, and the eigenvector corresponding to $Device_2$ and $Device_3$ as the test data set. The final result is shown in Fig. 2.

We test the network traffic of application on different devices, the average recall rate is 96.8%, the average recall rate is 96.6% and the F measure is 0.967. It can be seen that the method proposed in this paper has good generalization ability.

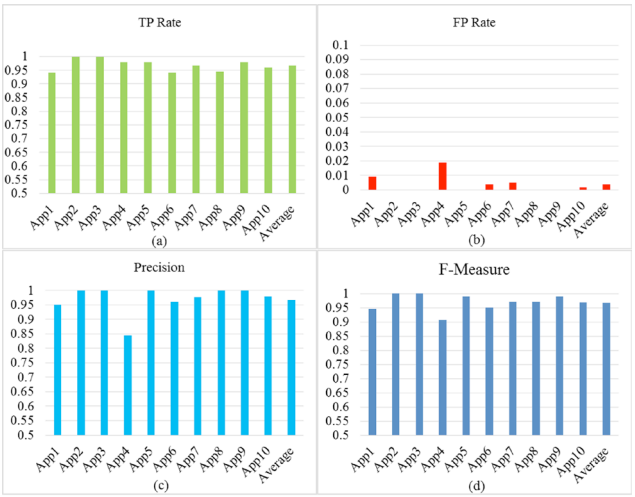

Fig. 2. Generalization ability test results

5 Summary

This paper introduces the process of mobile application recognition based on HMM model, and contains the data acquisition, feature extraction, model training, classification and identification. Finally, the proposed algorithm for mobile application recognition is 97.9%, and the generalization ability of the method is tested. The recognition accuracy is 96.8%. The experimental data used in this paper from a self-built wireless network with few users. The influence of different network on the accuracy of the method needs further verification.

Acknowledgment: This work was supported by the National Natural Science Foundation of China under Grant Nos 61372076 and 61301171, and the 111 Project under Grant No B08038.

References

1. Michael Shirer. Apple Tops Samsung in the Fourth Quarter to Close Out a Roller Coaster Year for the Smartphone Market[EB/OL]. http://www.idc.com/getdoc.jsp?containerId=prUS42268917. [2017-02-01].
2. Stat Counter Global Stats. Mobile and tablet internet usage exceeds desktop for first time worldwide[EB/OL]. http://gs.statcounter.com/press/mobile-and-tablet-internet-usage-exceeds-desktop-for-first-time-worldwide. [2016-11-01].
3. Wei X, Gomez L, Neamtiu I, et al. ProfileDroid: multi-layer profiling of android applications[C]//Proceedings of the 18th annual international conference on Mobile computing and networking. ACM, 2012: 137-148.
4. Xu Q, Erman J, Gerber A, et al. Identifying diverse usage behaviors of smartphone apps[C]//Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference. ACM, 2011: 329-344.
5. Dai S, Tongaonkar A, Wang X, et al. Networkprofiler: Towards automatic fingerprinting of android apps[C]//INFOCOM, 2013 Proceedings IEEE. IEEE, 2013: 809-817.
6. Su X, Zhang D, Dai S, et al. Mobile traffic identification based on application's network signature[J]. International Journal of Embedded Systems, 2016, 8(2/3):217.

7. Chen Z, Yu B, Zhang Y, et al. Automatic Mobile Application Traffic Identification by Convolutional Neural Networks[C]//Trustcom/BigDataSE/I SPA, 2016 IEEE. IEEE, 2016: 301-307.
8. Wang Q, Yahyavi A, Kemme B, et al. I know what you did on your smartphone: Inferring app usage over encrypted data traffic[C]//Communications and Network Security (CNS), 2015 IEEE Conference on. IEEE, 2015: 433-441.
9. Conti M, Mancini L V, Spolaor R, et al. Can't you hear me knocking: Identification of user actions on android apps via traffic analysis[C]//Proceedings of the 5th ACM Conference on Data and Application Security and Privacy. ACM, 2015: 297-304.
10. Park K, Kim H. Encryption Is Not Enough: Inferring user activities on KakaoTalk with traffic analysis[C]//International Workshop on Information Security Applications. Springer International Publishing, 2015: 254-265.
11. Conti M, Mancini L V, Spolaor R, et al. Analyzing android encrypted network traffic to identify user actions[J]. IEEE Transactions on Information Forensics and Security, 2016, 11(1): 114-125.
12. Datta J, Kataria N, Hubballi N. Network traffic classification in encrypted environment: a case study of google hangout[C]//Communications (NCC), 2015 Twenty First National Conference on. IEEE, 2015: 1-6.
13. Saltaformaggio B, Choi H, Johnson K, et al. Eavesdropping on fine-grained user activities within smartphone apps over encrypted network traffic[C]//Proc. USENIX Workshop on Offensive Technologies (WOOT'16, in conjunction with Security'16). 2016.
14. Fu Y, Xiong H, Lu X, et al. Service Usage Classification with Encrypted Internet Traffic in Mobile Messaging Apps[J]. IEEE Transactions on Mobile Computing, 2016, 15(11): 2851-2864.
15. Alan H F, Kaur J. Can Android Applications Be Identified Using Only TCP/IP Headers of Their Launch Time Traffic?[C]//Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks. ACM, 2016: 61-66.
16. Taylor V F, Spolaor R, Conti M, et al. Appscanner: Automatic fingerprinting of smartphone apps from encrypted network traffic[C]//Security and Privacy (EuroS&P), 2016 IEEE European Symposium on. IEEE, 2016: 439-454.
17. Zhihua Zhou.Learning: Machine learning [M]. Tsinghua University Press, 2016.
18. Li Deng.Analytical Depth Learning --- Speech Recognition Practice [M]. Electronic Industry Press, 2016.
19. Müller M. Information retrieval for music and motion[M]. Heidelberg: Springer, 2007.
20. Marco Gillies. HMMWeka[CP]. http://www.doc.gold.ac.uk/~mas02mg/software/hmmweka/index.html. [Last accessed 2017-04-20].