Android Internet Traffic Classification Based on Bayesian Analysis Techniques

luo dawei[1,a] ,han zijian[1,a],chen shangshu[1,a] ,liu shangdong[1,a]

[1,]Nangjing University of Posts and Telecommunications Jiangsu Nanjing 210046,china

[a]eiluodawei@sina.com

KEYWORD: traffic classification, Naive Bayesian, Android data flow

ABSTRACT. The wireless telecommunication techniques based on android operation system has been widely used both in life and work. It is of huge importance to classify and monitor the accurate android network traffic . However, the appliance of port jump technique and port encryption technique make classifying network traffic based on port type become much less effective. Here we put forward a method based on Naive Bayesian Analysis to classify different Android application, which combines the features of network data flow and Naive Bayesian. The experimental results show that this method has high classification accuracy.

1 Outline

According to the latest third quarter report from Strategy Analytics, Android operating system ranked first in the mobile operating system market with 83.6% market share. So we can see that the mobile network applications based on the platform of Android operating system have already become the mainstream in the market and the applications in Android operating system will increase in large numbers in the future. Thus study and analysis of Android Internet traffic classification has great significance.

However, due to the growing complication of mobile applications and Internet traffic, some emerging mobile terminal business are using technologies such as dynamic port and port encryption. These make the traditional *port-based* and *playload-based* Internet traffic classification methods inefficient. This paper presents a new method that is based on Naive Bayesian method (NB) to classify and analyze Android Internet traffic. The classification works in accordance with property of the Internet traffic data.

2 Traffic classification based on Naive Bayesian

2.1 Naive Bayesian

Naive Bayesian is a kind of distribution function that combines the categories and properties. It estimates the category of experiment samples by calculating the joint probability of them.

Naive Bayesian under normal circumstance: Assume that the sample space of random experiment E is S , $B_1, B_2, ..., B_n$ is a divide of sample space S , A is an event and $P(A) > 0$,here is the formula:

$$P(B_i \mid A) = \frac{P(B_i)P(A \mid B_i)}{\sum_{j=1}^{n} P(B_j)P(A \mid B_j)} i = 1, 2, ..., n \qquad (2\text{-}1)$$

$P(B_i)$ is called prior probability . We can achieve that by experience. The $P(B_i \mid A)$ got from the formula above is called posterior probability[2].

2.2 Naive Bayesian applied in data flow

When using Naive Bayesian to identify and classify internet traffic, we assume the classification of traffic flow F is $\{C_1, C_2, ..., C_k\}$, and k represents the number of categories of traffic. If randomly given a internet flow F ,the formula of conditional probability of that flow F belongs to category C_k is:

$$P(C_k \mid F) = \frac{P(F \mid C_k)P(C_k)}{P(F)} \qquad (2\text{-}2)$$

$P(C_k)$ is the prior probability of category C_k, it can be acquired by the proportion of flow C_k in the whole data flow. $P(F)$ is a normal constant, which signifies the marginal probability of flow F. $P(F \mid C_k)$ is the conditional probability of that flow F belongs to category C_k, we can get it from training data. Furthermore, the properties of traffic flow F can be simplified into the property eigenvector $(F_1, F_2, ..., F_n)^T$. So the probability $P(F \mid C_k)$ can be explained by formula (2-3)

$$P(F \mid C_k) = P(F_1, F_2, ..., F_n \mid C_k) \qquad (2\text{-}3)$$

According to the assumption of prior condition of Naive Bayesian, the properties of internet flow are independent to each other and they follow Gaussian distribution[3]. As a result of that, formula(2-3)can be simplified into(2-4)

$$P(F_1, F_2, ..., F_n \mid C_k) = \prod_{j=1}^{n} P(F_j \mid C_k) \qquad (2\text{-}4)$$

Then we apply (2-4) into (2-3) we can get (2-5)

$$P(C_k \mid F_1, F_2, ..., F_n) = \frac{\prod_{j=1}^{n} P(F_j \mid C_k)P(C_k)}{\sum_{k=1}^{m}\prod_{j=1}^{n} P(F_j \mid C_k)P(C_k)} \qquad (2\text{-}5)$$

3 Collecting sample of data flow

3.1 Choosing properties of data flow

Because the number of data we collected is so huge, we need to select appropriate and typical data flow so that we can apply them into Naive Bayesian, in which way can we avoid the cumbersome calculating process and ensure the accuracy of the Naive Bayesian as well. We choose five properties[4] as shown in Table 1.

Table 1 properties selected of data flow

Property 1	Duration of flow
Property 2	Total number of packets
Property 3	Total number of flows
Property 4	Average number of bytes in each packet
Property 5	Time intervals between each packet

3.2 Collecting data

We use *wireshark* network packet analysis software[5] to grab the network data packets on computers in the lab. Then we connect several mobile equipments with Android operating system to the wireless network generated by the computers and at the same time, run different network applications such as *wechat* , *weibo* and so on. W e can finally get network packet in the form of *pcap*[6]

3.3 Experiment tools and data handle

After collecting enough data packets, we use *JNetPcap*[8] to analyze the data flow based on Naive Bayesian. At first we install the *API* library in *eclipse* and set up *JNetPcap*. When we finish setting up the professional environment, we are able to write *Java* code to read the packets whose form is *pcap* [9] coming from *wireshark* .Finally, we are able to print the content of data packets and collect the statistics of data properties.

The data we analyze in the essay are all real. We continuously collected data for six hours from 10am to 4 pm on July 22th, 2015. 10928.54 M data was got in total as the sample data. It contains 68940 network flows generating by sex different types of application. All the data we got is shown in Table 2.

Table 2 data collected in lab

application	Number of flows	Number of packets	Average bytes
youku video	2844	70524	129821.6
wechat	5688	83088	733.6
weibo	35928	583416	319.7
QQ	10440	1112184	776.4
UC browser	10440	367704	517.3
baidu music	3600	992448	783.7

4 Result analysis

The indicators used to estimate the efficiency of method are *accuracy* and *overall accuracy* . The *accuracy* is used to identify single application and *overall accuracy* is used to identify the whole sample.

Assume that the number of classifying correctly is N_r and that of classifying wrong is N_w ,we have:

$$accuracy = \frac{N_r}{N_r + N_w} \qquad (4\text{-}1)$$

$$overall \;\; accuracy = \frac{\sum N_r}{\sum N_r + \sum N_w} \qquad (4\text{-}2)$$

4.1 Classification of the whole sample

When we analyze the experiment data, it is of importance to choose appropriate training data and testing data. It will help us to know about the stability and tendency of this method. We decided to choose separately 80%, 50% and 20% as the training data, the corresponding data left as the testing one. The results are shown in Table 3

Table 3 overall accuracy for NB

Training data	minimum	maximum	Average
20%	38.58%	81.13%	59.86%
50%	53.14%	87.67%	70.41%
80%	62.17%	88.95%	75.56%

We can clearly see that In the table the prior probability changes when the number of training data grow from 20% to 80%. In this process the *overall accuracy* for NB rises. However, the rising is relatively drastic, which means that NB is lack of stability and the accuracy itself is relatively low.

4.2 classification of single application

In order to study the accuracy of data analysis of *NB*,we also need to work on the *accuracy* for single application. We set up the mathematical model using 80% of sample data as the training data. The result is shown in table 4

Table 4 the accuracy for classifying different application

application	*youku video*	*wechat*	*weibo*	*QQ*	*UC browser*	*baidu music*
accuracy	80.70%	47.90%	77.00%	54.60%	92.10%	86.20%

The conclusion we can get from table 4 is that the difference *NB* make about the accuracy of different application is big, especially when working on the complicated application such as *wechat* who has many functions.

5. Conclusion

As Android operating system becoming the mainstream in the market, developing Android Internet traffic monitoring and analysis technology is of great significance in Internet management and supervision. This paper presents a new Android Internet traffic classification method based on Naive Bayesian method. According to the data we can see that NB method can be used in Android Internet traffic classification. However, this method has its disadvantages, e.g. Lack of stability, relatively low accuracy etc. One of its reason is that in NB method, the properties of Internet traffic only obey Gauss distribution while there are more complex relationships among them. Thus we can improve the NB method in this direction to make the classification more accurate in the future.

Reference

[1] Moore A W, Papagiannaki K Toward the accurate identification of network applications[C]//Proceeding of Sixth Passive and Active Measurement Workshop(PAM 2005). Springer-VerlagLNCS,2005:41-45

[2] Kong Gaohua, He Ming etc. Probability and Stochastic Processes.People's Posts and Telecommunications Press, 2012:18-19

[3] Chang Jianping, Li Hailin etc. Random signal analysis. Science Press，2006:30-32

[4] A.W.Moore and D.Zuev. Discriminators for use in flow-based classification Technical report, Intel Research,Cambridge,2005

[5] Wang Xiaohui etc. Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems. Tsinghua University Press, 2015:122-134

[6] Xu Yingkang， Zhang Ali. Design of network data packet analysis software based on PCAP.Modern Electronic Technology， 2013（10）：49-51

[7] Li Han， Liang Wei. Research on network traffic identification method.Communications Technology，2008,41（11）：88-90

[8] Mark Bednarczyk. JNetPacp[EB/OL]. http://www.jnetpcap.com/

[9] Mark Bednarczyk. Packet flows[EB/OL].http://jnetpcap.com/node/144

2015-07-21

[10] A.McCallum and K.Nigam. A comparison of event models for naive bayes text classification. In *In AAAI-98 Workshop on Learning for Text Categorization*,1998