Classification of Data of Mobile Chat Applications

Sachin P C and Ashutosh Bhatia

Computer Science & Information Systems, Birla Institute of Technology & Science, Pilani, IND.
h20180140p@alumni.bits-pilani.ac.in,ashutosh.bhatia@pilani.bits-pilani.ac.in

Abstract. Social media applications are used extensively in mobile and the data generated is enormous. This project captures the encrypted data transferred over the network, extracts the essential data, performs PCA and applies machine learning models to determine the mobile application associated with the data.

Keywords: Classification, Data, Mobile Application, Network, Wireshark

1 Introduction

With increasing mobile usage, there is a vast amount of data generated through social media chat applications every day. At the network level, this data is transferred as encrypted data and hence it is very difficult or impossible to analyze whether the data captured are of a voice call, a text message or a video call, or a Gmail message, etc. The process of capturing network data and applying a classification model to classify those data is defined as Traffic Classification. With mobile data usage is increasing day by day, the application-specific data is also increasing, and it is becoming challenging to classify the network data. With the increasing adoption of security protocols and more applications, it is becoming challenging to classify the network data, as it is becoming impossible to determine the kind of application the data belongs to. This paper presents a method to capture the network data, analyze it, and extract the essential data from the packets. The extracted data are considered features and a machine learning classification algorithm is applied to analyze and determine to which class the input data belongs. With this, if a sufficiently large amount of data is there as training data for each class, we can classify the unknown data into its corresponding class with very high accuracy. The prominent advantage of this approach is that we don't have to decrypt to figure out the type of data it is. This method is used at the application level, where we capture a particular application data using network analysis applications such as Tpacketpro.
This method can help the government understand the data transfer over the network better and analyze the network traffic over the internet. Not always,

analyzing the packets sent through the network is wrong; sometimes it helps understand the usage of data, it gives providers a fair idea of the application data that has been used the most, without actually knowing what the data is. Hence, this approach helps in analyzing the network data and classifying it into different classes.

2 Background

As the years pass by, mobile data traffic is increasing exponentially. Now, with decreased data rates, people have been using the internet through mobile very rapidly. If we compare the statistics, there is a massive rise in mobile data traffic when the data rates were considerably reduced and in India, there is a four-fold increase in IP traffic from 2016 to 2021. This only suggests how rapidly the network traffic is increasing. All this is because the network data is made available to most parts of the places and with reduced costs, most of the people are using various applications, and hence the application data are increasing rapidly.
An increase in network data implies that there is a massive amount of traffic data[1] that can be utilized to analyze what kind of traffic is produced and what applications form most part of the network data. We can classify the data using a specific machine learning algorithm depending on the requirement. With increasing mobile traffic, more security protocols are implemented, so it isn't easy to decrypt the network data as they come in encrypted form. We cannot analyze the network data just by seeing the encrypted data, as it will be completely different and hence, we cannot figure out what application data or what data it is just by extracting it and hence analyzing the network data is complex. However, with a large amount of data, there will be a pattern for the same kind of application data and hence, using a classification algorithm on these data, we can determine the class of data to which it belongs.

3 Proposed Approach

This section will illustrate the step-by-step procedure to obtain the classification model to classify various mobile chat application data.

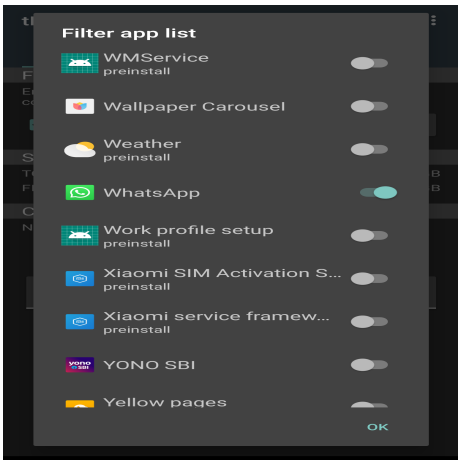
3.1 Data Capture

The initial measure is to capture the data. One of the primary and essential steps in this approach is to capture data, as it is used to classify. We capture the data at the application level. We have used TpacketCapture Pro[3] to capture the data packets, which captures the inbound and outbound packets of particular chat applications which we specify. While capturing the data, specify a particular application and then start capturing the packets. The tpcaket capture application captures the packets and stores them in pcap files which are later

used to extract the features needed. We can use wireshark to visualize the packets captured and stored pcap files. Wireshark helps in visualizing the packets and its structure and fields.
Steps to capture the data using tpacketCapture pro:

- Install the application on your mobile device.
- Specify the application you want to capture the data using the "Filter App List" option.
- Start Capturing the data.
- Once you have sufficient data, stop the capture.

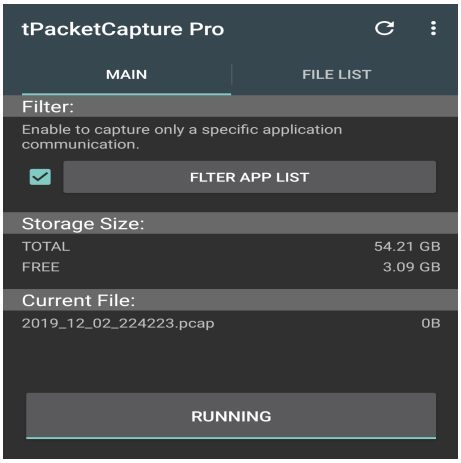
The captured data is stored in a pcap file, and we can visualize the captured data using Wireshark.



Filter Application whose data has to be captured.

3.2 Extracting the features

The next step is to extract the required features from the pcap file[4]. All the pcap files captured will have a vast amount of data. We need to extract specific data from those pcap files and perform some analysis on those data to obtain the right features. Every pcap file consists of packets of data. Every packet will have header data, payload, etc. We need to extract essential features like the length of the payload, protocol of the data packet, which type it is, etc and we need to group all the data which are of a specific session or connection. I.e. we need to group packets from the beginning of a three-way connection in start to finish in the end, if it is a TCP connection. If it is a UDP connection, we need to group packets concerning the same source IP, port number, and destination



Start Capturing of Data.

IP and destination port number.
The grouping of packets as one is helpful in getting certain features such as the average size of the packets, maximum size, minimum size and many more other features. These features are extracted for every group of packets and is taken as a data set for the machine learning model. So, whenever a new data set has to be classified, these following steps have to be performed, and then we have to apply a classification model to classify the extracted tuple into a specific class.

	A	B	C	D	E	F	G	H	I	J	K
1	MEAN	MEDIAN	STANDARD DEVIATION	MINIMUM	MAXIMUM	INBOUND_AVG_IAT	INBOUND_MEAN	INBOUND_MEDIAN	INBOUND_SD	INBOUND_MIN	INBOUND_MAX
2	158.0590406	43	496.2503216	40	5580	0.001365459748	246	40	682.7232236	40	5580
3	852.7435897	40	1432.795147	40	5472	0.003654572699	1655	1398	1728.21729	40	5472
4	1827.486486	40	3001.385945	40	9546	0.0005558799295	3659	3552.5	3484.167189	40	9546
5	1210.344086	40	1630.657259	40	5627	0.0005244149102	2384	2756	1624.813528	40	5627
6	15129.07143	43	25827.97681	40	64280	0.0005938640008	30163	22447	29977.35587	40	64280
7	63.94957983	40	60.78662999	40	549	0.0005035117521	66	40	72.89718787	40	549
8	71.76363636	40	85.74989526	40	668	0.001368743723	82	40	113.1149857	40	668
9	348.7647059	40	763.7743064	40	3019	0.003479003906	620	55.5	1078.009276	40	3019
10	80.67948718	43	78.81910585	40	475	0.0005924490434	95	40	100.2945662	40	475
11	80.93103448	40	143.4702873	40	2113	0.0009214888919	100	40	196.0408121	40	2113
12	2440.487179	40	4136.668555	40	13775	0.0005417797301	4915	6830	4847.068289	40	13775
13	750.4451411	40	805.7512897	40	2756	0.0007048362418	1454	1398	556.3658868	40	2756
14	78.88111888	40	94.7698836	40	738	0.001577483283	89	40	120.6772555	40	738
15	80.25592417	43	108.8973929	40	1593	0.000503105341	94	40	143.4851909	40	1593
16	86.95505618	43	104.3227727	40	707	0.0004835854406	107	40	137.1787156	40	707
17	81.94444444	43	95.39419772	40	668	0.0004925021419	95	40	122.9512098	40	668
18	63.91666667	40	71.14831926	40	289	0.0005340099335	40	40	0	40	40
19	99.35064935	41	194.8197267	40	2197	0.0007809601821	126	40	266.3400083	40	2197
20	888.5977011	40	1114.768924	40	4114	0.0005313101269	1733	1398	1044.088598	40	4114
21	101	41.5	207.5801589	40	1398	0.0004943609238	136	40	283.0017668	40	1398
22	832.8205128	40	1313.306351	40	4114	0.0005066527261	1615	1398	1536.350546	40	4114
23	91.29166667	40	122.8411768	40	585	0.0008965015411	109	40	164.972725	40	585

Features extracted from pcap file.

The above figures show that the final features taken are majorly grouped by inbound packets, outbound packets, and the mixture of both.
There are 20 features extracted from the pcap files and a set of tuple are created,

	L	M	N	O	P	Q	R	S	T
1	OUTBOUND_AVG_IAT	OUTBOUND_MEAN	OUTBOUNT_MEDIAN	OUTBOUND_SD	OUTBOUND_MIN	OUTBOUND_MAX	TOTAL_INBOUND_PACKETS	AL_OUTBOUND_PACK	PROTOCC
2	0.00142340046	66	43	56.82429058	40	360	138	133	6
3	0.003678043683	89	40	135.6981945	40	523	19	20	6
4	0.0005252080805	92	40	138.8956443	40	523	18	19	6
5	0.0005255699158	61	40	91.1811384	40	505	46	47	6
6	0.0006013313929	95	40	147.1971467	40	565	14	14	6
7	0.0005090487631	61	43	45.80392996	40	289	60	59	6
8	0.00142451433	60	43	39.73663297	40	289	56	54	6
9	0.00348510061	107	40	103.7737925	40	317	8	9	6
10	0.0006331946399	64	43	41.74925149	40	289	80	76	6
11	0.0009505962426	60	43	37.62977544	40	289	221	214	6
12	0.0005476739671	88	40	132.6989073	40	505	19	20	6
13	0.0007052074505	51	40	62.30569797	40	557	159	160	6
14	0.001671121401	68	43	55.26300752	40	338	73	70	6
15	0.0005305128939	64	43	47.89572006	40	375	216	206	6
16	0.0005215564406	64	43	37.08099244	40	289	93	85	6
17	0.0005224171807	67	43	51.29327441	40	289	55	53	6
18	0.0004316568375	87	47.5	98.80789442	40	289	6	6	6
19	0.0008133488732	70	42	54.66260147	40	348	78	76	6
20	0.0005324454535	62	40	94.23905772	40	506	43	44	6
21	0.0005032631659	63	43	47.2546294	40	289	35	33	6
22	0.0005099905862	88	40	132.8645927	40	506	19	20	6
23	0.0008197697726	76	43	75.57115852	40	303	11	13	6

Features extracted from pcap file.

which are used as data for the classification algorithm that we will use next to classify the data into its specific class.

3.3 Classification

For classification[5], we use concepts of Machine Learning. Machine learning algorithm written is such that it learns from the previously input data and hence tries to produce better results with the new data. The classification can be done using many machine learning algorithms such as K-means, K-nearest neighbor, Support Vector Machine, Naive Bayes, etc. We can use a specific algorithm depending on the data. As of now, we have used the k-means algorithm to classify the given data. K- means is an unsupervised algorithm, which uses the concepts of cluster to classify the given data. A set of clusters are defined and for every cluster and for every iteration a new centroid is calculated for the clusters until no more new changes for the centroids are there.The algorithm aims at minimizing the squared error given by:

$$j(v) = \sum_{i=1}^{c} \sum_{j=1}^{c_i} (|x_i - v_j|)^2$$

where,

- $|x_i - v_j|$ is the euclidean distance between $x_i and v_j$.
- c_i is the number of data points in i^th cluster.
- c is the number of cluster centers.

We have applied Principal Component analysis to remove the highly co-related components in the data, so that it will not affect the classification with redundant data. Once PCA is applied, we get a set of features that are used as final features for the classification of data.

4 Results

We have used a K-means algorithm to classify the voice call, text message, video call, email messages data. Each of these is taken as separate classes and is trained. The implementation of this algorithm gave an accuracy of around 65 percent. This accuracy can be increased as the model is trained with more data of different classes.

5 Future Scope

The result obtained by performing the classification algorithm is moderate as the classification performed was for a minimal amount of data. In the future, we can perform the same classification algorithm with more data used for training the model, which might help in giving better results. We can use different algorithms like SVM, Decision Tree, etc., compare the results with the obtained result, and analyze which classification algorithm performs better.

6 Conclusion

With Mobile network data increasing rapidly, there is a huge scope to explore the network data generated. As the data generated is enormous, we can explore the pattern among similar kind of applications and analyze them. With many machine learning models, along with network analysis, we can identify the class of data without decrypting the data sent over the network. This can be beneficial for many agencies, governments, etc, to analyze network data better.

References

1. Peter Velan , Milan Cermak, Pavel Celeda, Martin Drasar. "A Survey of Methods for Encrypted Traffic Classification and Analysis".
2. Peek-a-boo, i still see you: "Why efficient traffic analysis countermeasures fail" by KP Dyer, SE Coull, T Ristenpart, T Shrimpton - 2012 IEEE symposium on security and privacy, 2012
3. Android software - tpacketcapture. URL http://www.taosoftware.co.jp/en/android/packetcapture
4. Mauro Conti, Luigi V. Mancini, Riccardo Spolaor, and Nino Vincenzo Verde. "Can't you hear me knocking: Identification of user actions on android apps via traffic analysis", pages 297 304, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3191-3. doi:10.1145/2699026.2699119. URL-http://doi.acm.org/10.1145/2699026.2699119
5. T. T. T. Nguyen and G. Armitage. "A survey of techniques for internet traffic classification using machine learning".IEEE Communications Surveys Tutorials, 10(4):56 76, Fourth 2008.ISSN 1553-877X. doi: 10.1109/SURV.2008.080406.