Wi-Fi encrypted traffic machine learning classifier

Claudia Baz Alvarez, David Quintero Olaya
054326 - WIRELESS INTERNET
REDONDI ALESSANDRO ENRICO CESARE

Abstract
A supervised machine learning model is propose to resolve the classification of packets between 5 different applications: web browsing, Youtube, Instagram, video conference and an idle device. The traffic curves generated by the applications are analysed, filtered and some statistical metrics are calculated from them as the input of the machine learning model. For the simple model used, the results are satisfactory, however, leaving a path for improvement

Contents

1 Introduction

Network traffic classifiers are essential tools for network administrators to properly manage network resources, optimize network performance and monitor it. They make it possible for administrators to control and prioritize traffic according to its relevance identifying different forms of QoS priorities based on the application sending the traffic.

However, typical network classifiers have substantial difficulties because to the complexity and volume of network traffic that is only growing. When machine learning techniques are integrated with statistical methods, network traffic identification and classification can be done accurately.

2 Methodology

A supervised model requires a labeled dataset. To fulfill it, a controlled environment was set, where packets could be captured and labeled. After, the raw packets with the associated ground truth were sampled within a constant time window of 4 seconds. Finally, the samples were processed to collect statistical metrics. This final data was the input of the model.

2.1 Data Collected

The packets were collected from the direct connection between an Access Point and a mobile device. Knowing the MAC address of the device, a computer with Wireshark enabled on monitor mode was connected to the same network. Then, the packets coming or arriving to the mobile device could be filter based on the mobile address. There were considered only those with the destination and source address equal to the MAC device:

wlan.sa == aa:3a:0e:b6:ed:12 or wlan.da == aa:3a:0e:b6:ed:12

The packets were collected as the mobile device was consuming a certain application or performing a certain task. In that sense, there were collected 5 different datasets each one with its label:

- Simple **browsing** activity: searching an reading google indexed pages
- The device as **idle**, just connected to the network
- Scrolling in the **Instagram** app
- The device being use for a **video call**
- Playing some **Youtube** videos

All the tasks were "recorded" for a period of at least 20 minutes.

sample_no	avrg_IAT	var_IAT	avrg_len	var_len	count	upload	download	PSM	qos_type0	qos_type6	qos_type1	gt	qos_type4
0	0.004879	0.002596	245.433796	105150.644104	793	177457.0	17172.0	990.0	774.0	1.0	3.0	browsing	0.0
1	0.020837	0.007192	336.933333	179371.477607	195	65630.0	72.0	858.0	178.0	1.0	1.0	browsing	0.0
2	0.174556	0.042765	107.928571	2477.637755	14	1511.0	0.0	528.0	6.0	0.0	0.0	browsing	0.0
3	0.176373	0.322270	361.866667	210630.715556	30	10856.0	0.0	462.0	23.0	0.0	0.0	browsing	0.0
4	0.055887	0.073080	423.642857	369054.479592	56	23724.0	0.0	2244.0	21.0	1.0	0.0	browsing	0.0
5	0.120829	0.113386	212.571429	120615.387755	14	2976.0	0.0	660.0	4.0	0.0	0.0	browsing	0.0
6	0.510028	1.708154	235.357143	108788.229592	14	3295.0	0.0	330.0	9.0	0.0	0.0	browsing	0.0
7	0.701263	1.508475	104.800000	2264.560000	5	524.0	0.0	198.0	2.0	0.0	0.0	browsing	0.0

Figure 1: Packets samples processed

2.2 Data process

After sampling the data, statistical metrics such as average packet length or IAT (Inter Arrival Time) were calculated. In addition, there were counted the upload/download packages and the different types of QoS priority packets per sample.

2.3 Model Used

In the area of machine learning, K-Nearest Neighbors (KNN) is a popular supervised algorithm. Is is popular due to its simplicity and that it can be utilized for both classification and regression problems. Determining the k-nearest data points to a given data point where a prediction needs to be made is how the KNN algorithm operates. Then, it assigns a class label to the nearby data points, which is subsequently used to forecast the value of the input data point.

Due to its simplicity and that it fit the requirement of a supervised algorithm, it was chosen as the classifier model.

3 Results

3.1 Analysis of the data

The data collected is represented by Figures 4 and 3.

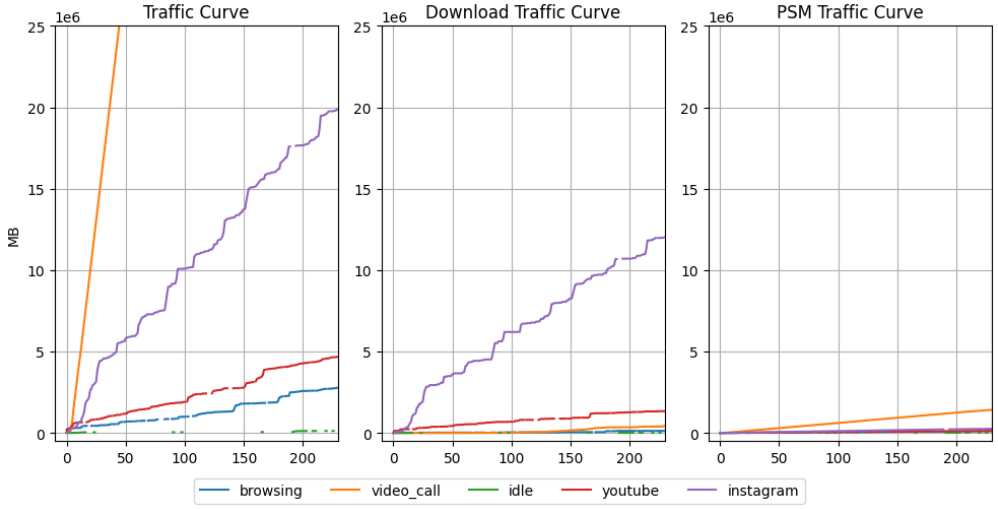

Figure 2: KNN accuracy based on the number of neighbors

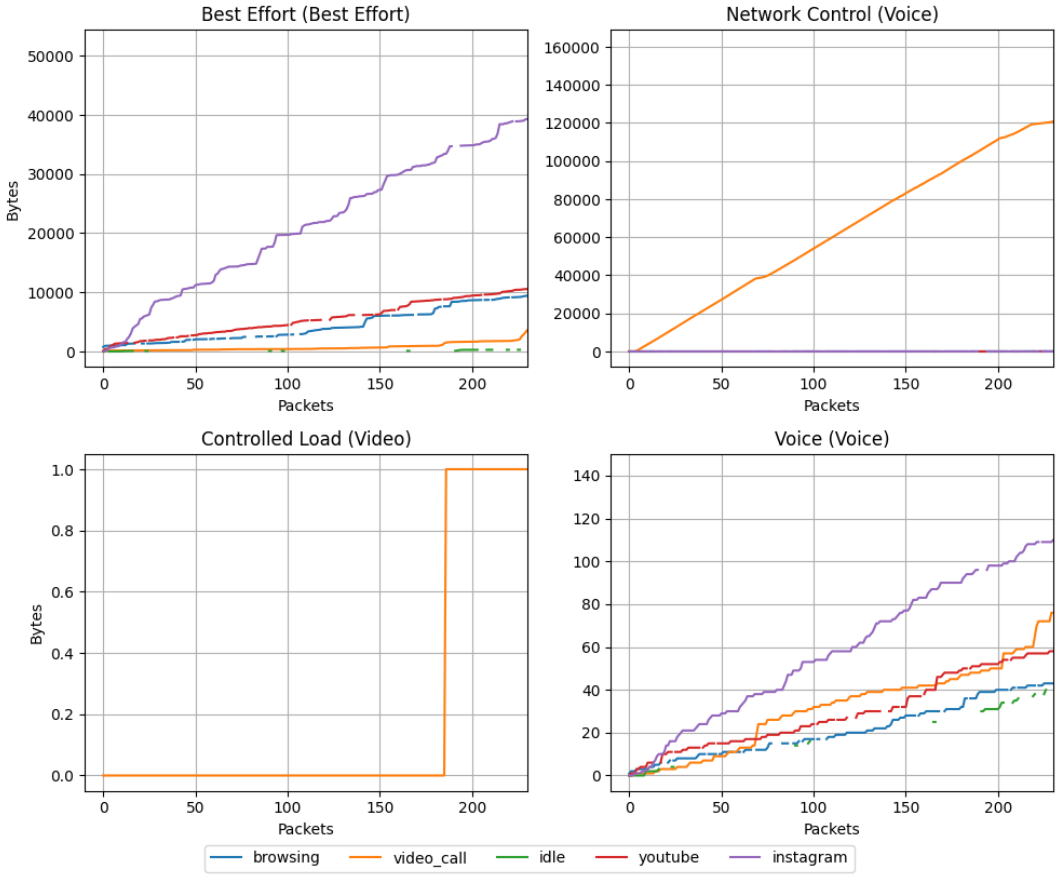

Figure 3: Traffic curves of QoS packets

As expected, the most incomplete curves are those one corresponding to the idle activity, followed by the browsing one. The application that exchanged more information (bytes) is the video conference, followed by Instagram, Youtube browsing and the idle, in that sequence. The video call was the one with the highest number of priority packets in order of kilobytes, compared to the bytes and lower priority QoS of the rest of applications. Some voice packets can be associated with the browsing activity due to propaganda appearing on searched websites. Besides that, Instagram was the second application in quantity of higher priority packets received/sent and the one with more packets of the lowest priority queue.

3.2 The model results

To obtain the best result from the KNN its required to iterate over the possible values of K (number of neighbors) and choose the one with the best accuracy. In that sense, K=7 was the one with better performance, as showned in Figure 4.

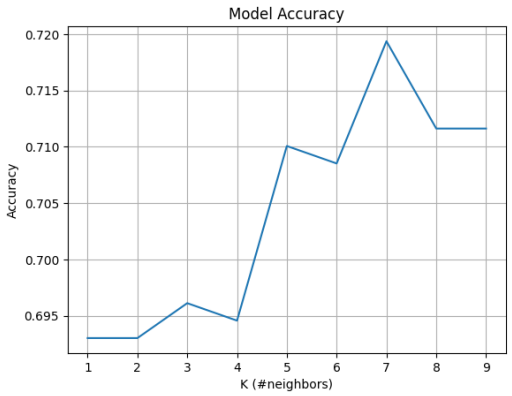

Figure 4: KNN accuracy based on the number of neighbors

The final result can be seen in Figure 5 where the confusion matrix is plotted:

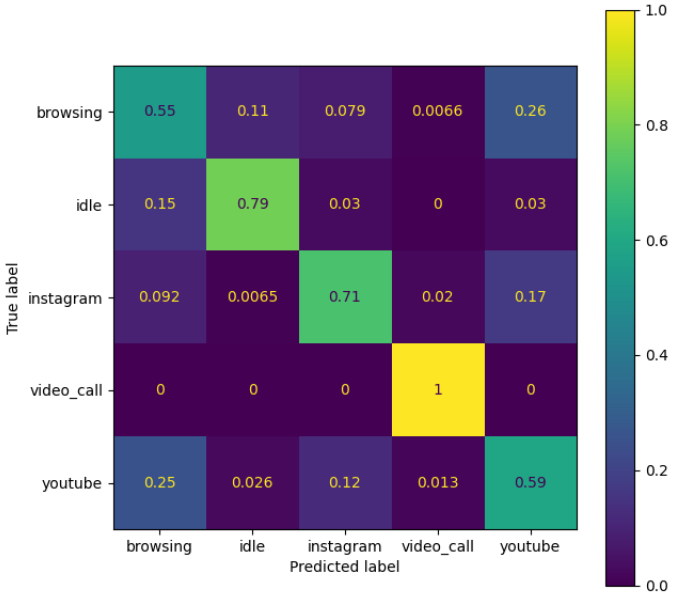

Figure 5: Confusion Matrix

4 Conclusions

The video conference is the application detected with best accuracy. As the one with the highest number of packets and priority packets is the easiest to be classified. Behind it, the idle class presents the opposite behavior. It's low demanding with few packets to be send or receive. This contrasts in relation to the other classes help to define a clearer profile for them.

In second hand, the ones with lowest accuracy were the browsing and the Youtube activities. Both, between them are the second most probable label. Their traffic curves of QoS are similar although the Youtube activity has a higher consume of bytes, especially on higher priority queues. This result could be improve using a model that weighted higher the average length packets.

The Instagram class its founded with in the middle, not having an obvious different behaviour than the Youtube type, however with more traffic differentiating it from the browsing activity more than the Youtube one.

In conclusion, the statistical parameters obtains from sniffing helped to classified the different applications at the MAC layer. A more sophisticated model could be use to improve the classifier performance with the same metric.