Wi-Fi Encrypted Traffic Classification

Alessandro Zito, Carlos Santillán

September 6th 2022

Supervisor: Alessandro Enrico Cesare Redondi

Course: Wireless Internet

Contents

1 Introduction

This project aims to implement a classification algorithm in the context of Wi-Fi traffic. The classification problem in this context may aid in making more informed decisions in network resource allocation or when performing diagnostics.

The presented pipeline starts with sniffing encrypted Wi-Fi traffic in monitor mode, the subsequent construction of a traffic dataset, and the training of a classifier to distinguish different types of traffic. Finally, we will focus on a single known device, but the procedure can be generalized to many devices contemporaneously. Data analysis was performed with R.

2 Sniffing and Dataset Building

2.1 Sniffing

The first step in building the dataset is to capture Wi-Fi traffic. We decided to distinguish between five types of traffic:

- Idle device
- Web browsing
- VoIP calls
- Video calls
- Youtube streaming

We captured (in *monitor mode*) approximately 20 minutes of traffic per type (via Wireshark) and stored them in .pcap files. We filtered traffic sent from and received by the target device[1]. Since the traffic is encrypted, we cannot rely on port numbers, IP addresses, etc., for the purposes of classification. Instead, we had to work exclusively with statistical features.

2.2 Dataset Building

Each entry in our dataset consists of a traffic flow lasting W seconds. This required some preprocessing on the original data. For each traffic flow, which includes all packets sent and received by the target device in the interval $[t, t + W)$, we extracted the following statistical features:

- Mean and variance of packet length

[1] MAC address A4:42:3B:D2:F7:08

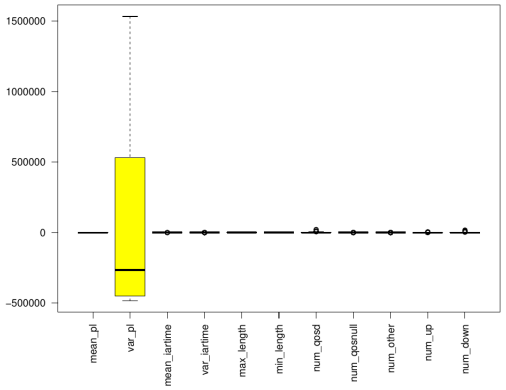

Figure 1: Boxplot of original features, variance of packet length creates an unbalance

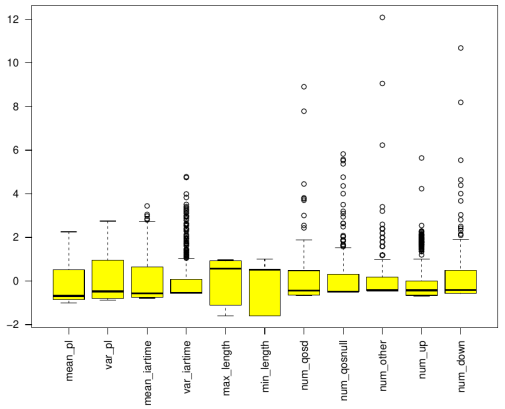

Figure 2: Boxplot of normalized features, our classifier will no longer be biased towards one feature

- Mean and variance of inter-arrival time
- Maximum and minimum packet length
- Number of packets of type: QoS Data, QoS Null function, Other
- Number of sent and received packets

The choice of the parameter W was based on the ability to distinguish the traffic types in that time frame significantly. Starting from 5 seconds and increasing, we finally chose 15 seconds since it met our criteria. So we processed the data to obtain the features and produced a dataset containing 477 rows.

3 Data Exploration

We looked at the variance of the features and noticed a significant unbalance (see Figure 1) that we addressed by normalizing the data (see Figure 2).

Because of the high number of features, we performed principal component analysis for visualization purposes. The first three principal components explain over 75% of the variability. We proceed to assess the separability of the data qualitatively. Figure 3 shows the PC1 v. PC2 plot. Figure 4 shows PC2 v. PC3, which yields similar results.

The plots confirm the presence of well-defined clusters. With this information, we proceed to train the classifier using the original features.

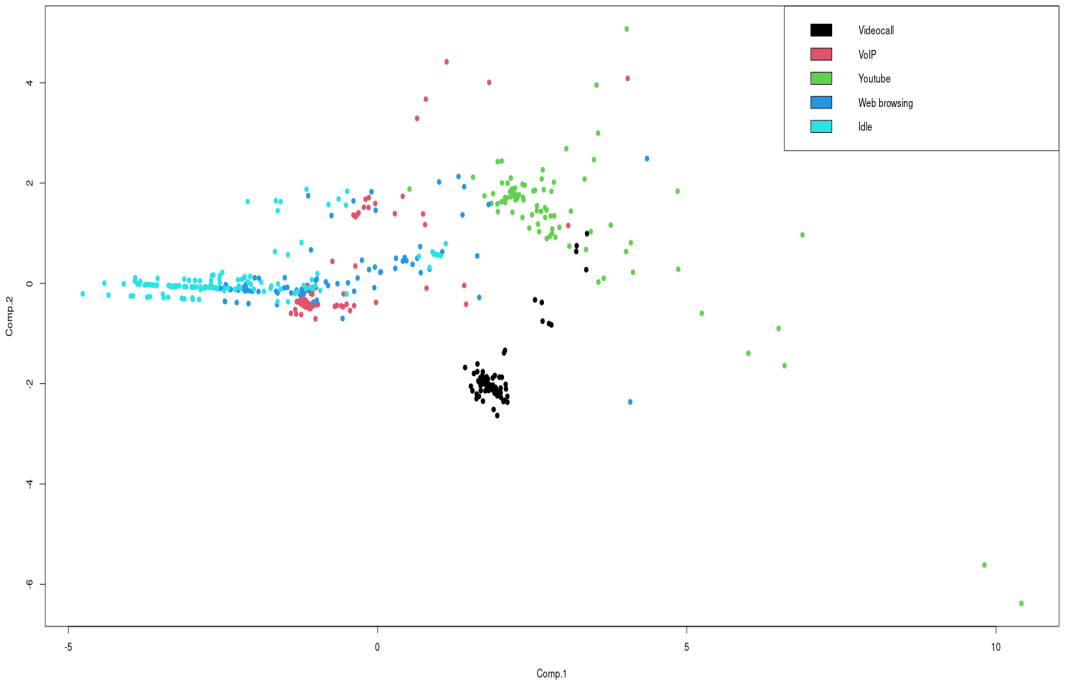

Figure 3: The PC1 vs PC2 plot shows a significant degree of separability

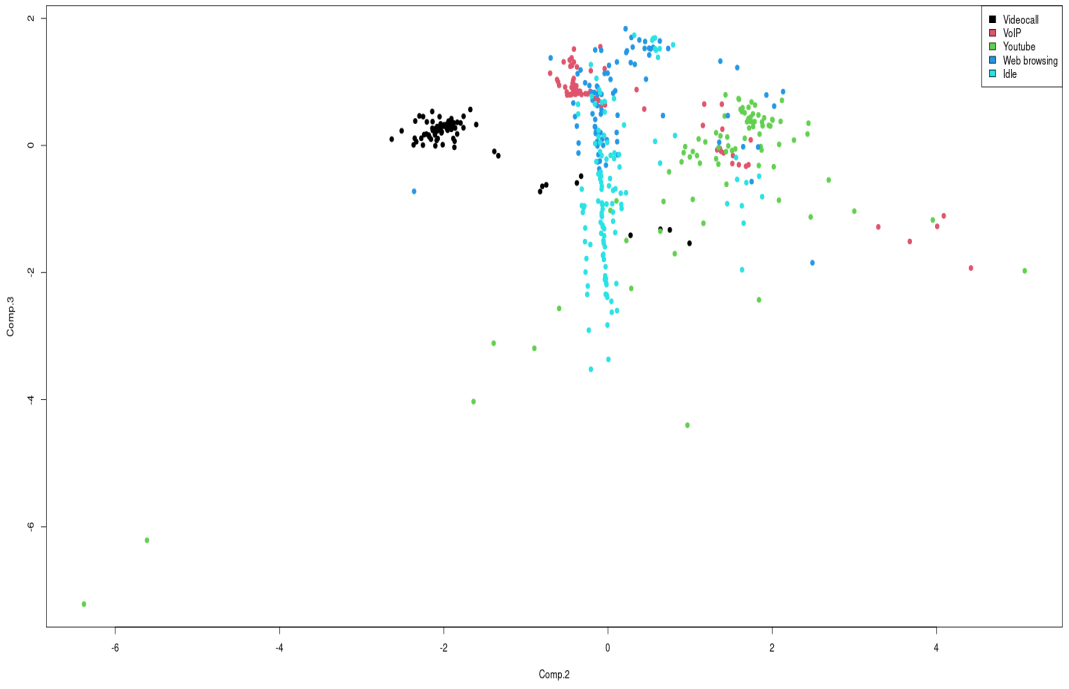

Figure 4: Confirms a significant degree of separation in the PC2 v. PC3 plane

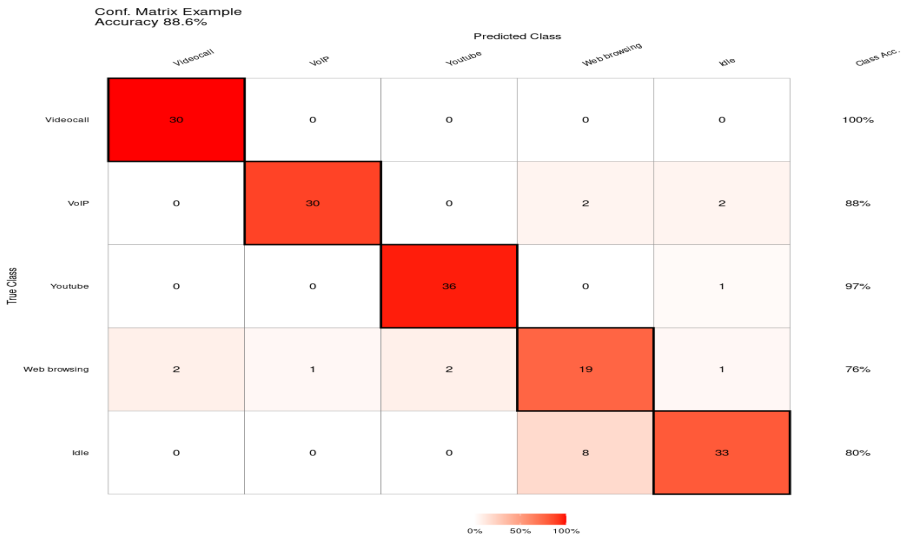

Figure 5: Confusion matrix of the test set

4 Classification and results

Many machine learning techniques can tackle this problem: logistic regression, support vector machines, Bayes classifiers, etc. However, we chose k-Nearest-Neighbours since the data was well-clustered for it to perform well, and it did not require any strict assumptions on normality and equal variance between clusters. When KNN classifies a new sample, it finds its k nearest neighbors (in the training dataset) and classifies it in the most common class among those neighbors. We used the euclidean distance. We shuffled the data and divided it into a training set (65%) and a test set (35%) and tuned the hyper-parameter k via leave-one-out cross-validation.

After finding 6 as our optimal k, we proceed to compute the test error[2]. We obtain an accuracy of 88.6%. Figure 5 shows the confusion matrix, and Figure 6 approximates the classification region. We see that web browsing and idle traffic are difficult to distinguish, whereas video calls are much different from the others; therefore, it is easy to classify them correctly.

5 Conclusion

We have reached the goal of classifying traffic with an acceptable level of accuracy. It may be possible to achieve even better results with a further transformation of the original features (use of basis functions) or via kernel functions and support vector machines. In addition, the pipeline

[2]The test set is normalized using the same normalization obtained from the training set.

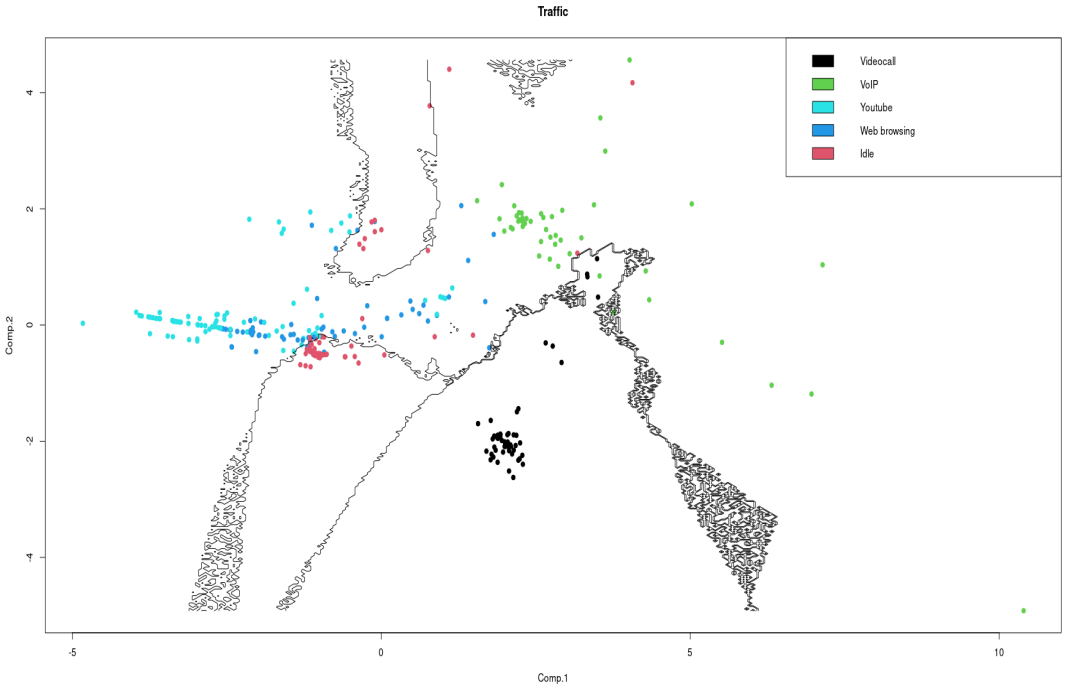

Figure 6: An approximation of the classification region using the first two principal components

can be smoothly adapted into a real-time environment so that we may monitor the state of the network (at least in terms of traffic) and allocate resources accordingly.

Bibliography

[1] Pacheco, Fannia, Ernesto Exposito, Mathieu Gineste, Cedric Baudoin, and Jose Aguilar. "Towards the deployment of machine learning solutions in network traffic classification: A systematic survey." IEEE Communications Surveys & Tutorials 21, no. 2 (2018): 1988-2014.

[2] Li, Wei, and Andrew W. Moore. "A machine learning approach for efficient traffic classification." In 2007 15th International symposium on modeling, analysis, and simulation of computer and telecommunication systems, pp. 310-317. IEEE, 2007.