# Poster: Data Collection for ML Classification of Encrypted Messaging Applications

Jason Hussey
*Computer Science*
*Colorado School of Mines*
Golden, Colorado, USA
jasonhussey@mymail.mines.edu

Ethan Taylor
*Computer Science*
*Colorado School of Mines*
Golden, Colorado, USA
ewtaylor@mymail.mines.edu

Kerri Stone
*iCR, inc.*
Lousiville, Colorado, USA
kstone@icr-team.com

Tracy Camp
*Computer Science*
*Colorado School of Mines*
Golden, Colorado, USA
tcamp@mines.edu

***Abstract*—Network traffic classification is used to identify the nature of traffic on a network. Entities capable of monitoring network traffic use classification for all manner of reasons, including identification of mobile applications being used on the network. It is possible that the usage of encrypted messaging applications by users on these networks can be detected, betraying elements of their privacy.**

**In this paper, we describe a system that leverages campus network resources to generate real-world data alongside a more curated dataset captured from Android application traffic. We also explore the ability of machine learning (ML) models to accurately classify traffic from these encrypted messaging applications. Understanding what is revealed from network data is important given that the use of these applications is meant to maximize privacy in the first place.**

***Index Terms*—network traffic classification, Android application, encrypted messaging, machine learning**

## I. Introduction

### A. Encrypted Messaging Applications

Millions of people around the world use encrypted messaging applications to preserve the privacy of their messages [2]. These applications securely encrypt the contents being sent back and forth between users, ostensibly providing privacy from any entity observing the network traffic. Motivation for using these types of applications could range from everyday concerns about privacy by ordinary citizens to more operational purposes such as coordinating protests, sharing digital materials, organizing illegal activities, and more.

Even though message contents are encrypted with these applications, much can be revealed by observing the communications at a high level. Similar to call records from a telephone system, or TCP/IP headers from web traffic [3], interested parties who have access to metadata about conversations can begin to reveal aspects of the conversation that are generally assumed to be private.

### B. MIRAGE-19 dataset

In [1], a team of researchers provide an open source dataset containing bi-directional flows (bi-flows) from applications on Android mobile phones. The data is available in JSON format and is comprised of traffic from 40 different applications on three phones generated by 280 volunteers. While the authors describe the broad techniques used in creating the dataset, the source code is not available nor is there sufficient detail about libraries and tools used to capture and process the datasets such that other researchers can replicate the process. The inability to create data for new applications is an issue in our case as we seek to explore the application of ML to a very specific set of applications not represented in this dataset. Without the source code, we create and implement our own pipeline to approximate the approach in [1].

## II. Methodology

### A. Data Collection

The data we have collected for this work falls into two broad categories. The first is a fine-grained packet capture of network traffic generated by a *single application* on an Android phone similar to that described in [1]. For the second, we augment the Android application data with a three-week capture of packet headers from Colorado School of Mines' Wi-Fi network.

*1) Android Application Traffic Capture:* Capturing network traffic generated by a specific application on a mobile device requires careful system design. The main challenge lies in the inability to differentiate between packets generated by *different applications on the same mobile device*. To address this, we leverage the techniques broadly described in [1]. We simultaneously run `tcpdump` on an Ubuntu station serving as a wireless access point and log network system calls from a specified Android application using `strace`[1]. These `strace` log file reveals network socket connections used by the application, allowing us to filter the packet capture down to just those packets.

*2) Wi-Fi Network Traffic Capture:* With our institution's (Colorado School of Mines) Information Technology Service, we establish a port mirror that captures the bi-directional Wi-Fi traffic on campus[2]. We receive the mirror on a server within our research team's control where we further process the data.

The sustained data rate of this traffic is approximately 1.5 Gbps during core hours. To capture this large volume of data without any packet loss, we utilize the ntop organization's n2disk[3] utility which allows for multi-gigabit captures.

[1]https://strace.io/
[2]This partnership was approved by our institution's IRB
[3]https://www.ntop.org/products/traffic-recording-replay/n2disk/

## B. *Experimental Design*

For Android application traffic captures, we generate the data from two models of phones (Samsung A51 and Xiaomi Poco X3) running their stock Android OS. While the team in [1] use a custom OS across all of their phones (e.g., CyanogenMod), we believe the software variance in our dataset will ultimately provide models a greater ability to generalize to real-world data. Both phones are rooted to obtain super-user privileges enabling us to load the *strace* utility onto the phone, generate strace logs for an application during its use, and retrieve those log files from the phone after a capture session is finished.

When capturing packets from the Wi-Fi network, we only keep layer 3 and 4 headers from each packet. Layer 2 headers are relevant only to the local network and payloads are largely encrypted today, thus they are both discarded. Excising these headers has the added bonus of reducing total disk space required to store the dataset during processing.

We have chosen to initially focus on creating a dataset of labeled traffic from encrypted messaging applications versus applications spanning multiple genres. We believe, based on preliminary results in Section III, applications belonging to different genres are fairly discernible from one another given their distinct usage patterns. On the other hand, applications all belonging to the encrypted messaging genre, like WhatsApp, Signal, and Telegram have similar usage patterns.

## C. *Data Analysis*

We choose to process all of the packet captures into bi-flows: a conversation between two hosts where each packet shares the same 5-tuple identifier (destination and source IP addresses and port numbers plus the layer 4 protocol). In the case of a bi-flow, ordering of the source and destination IP addresses does not matter, though importantly we record the direction of each packet so re-creation of uni-directional flows is possible with our dataset.

For each flow, timing and size values of the associated packets are stored in arrays. Aside from being a component of the key-field, no IP addressing is carried forward into the analysis, as in many cases this would heavily bias results by tying applications to IP addresses. These arrays become the basis for all future ML applications with this data. From this per-packet data associated with each 5-tuple key, we can generate myriad statistics about the flow, including features such as total bytes sent, total length of the flow, data rates, etc.

## III. Preliminary Results

In the course of exploring the dataset in [1], we applied Random Forest models to the bi-flow metadata for a simple classification task. In these experiments, we evaluated the ability for a model trained on application data from one phone to classify traffic generated by that same application on a different phone. For this model, we discovered that the layer 4 payload size in the outbound direction is the most important feature for classification, as depicted in Figure 1. This makes sense to us as the applications belong to different genres and likely have very different usage patterns, thus that usage pattern (bytes sent up and down) is a strong discriminator between the various applications.

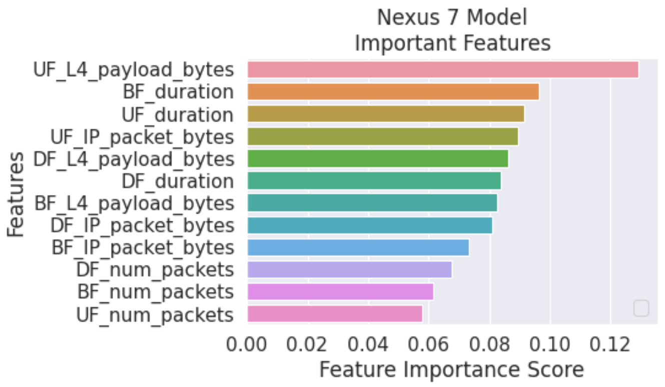

Fig. 1. The features important in a Random Forest model trained on the Samsung Nexus 7 data provided in the MIRAGE-19 dataset for classifying bi-flows from a standard train-test split as well as those from a Xiaomi phone also included in the dataset.

## IV. Future Work

We envision two distinct paths forward with this work. The first is the training of supervised and semi-unsupervised ML models on this dataset. We will have a strong emphasis on encrypted messaging applications which we expect to all appear similarly in the network data. The goal of these models will be classification and clustering of the encrypted messaging applications in various environments.

The second endeavor we will pursue is systems focused. We intend to thoroughly describe the methodologies, technologies, and considerations used in creating bi-flow data like this. We hope this empowers other networked systems researchers to create datasets for specific applications of interest to them in addition to leveraging the real-world data at their local institutions.

## Acknowledgment

We would like to thank Colin Randall and Dr. Phil Romig of Information Technology Services at Colorado School of Mines. They were instrumental in the data collection as well as the navigation of and compliance with institutional policies.

## References

[1] G. Aceto, D. Ciuonzo, A. Montieri, V. Persico, and A. Pescape, “MIRAGE: Mobile-app Traffic Capture and Ground-truth Creation,” in *2019 4th International Conference on Computing, Communications and Security (ICCCS)*. IEEE, Oct. 2019, pp. 1–8.

[2] R. Hodge, “After Musk tweet, Signal and Telegram see millions of new downloads.” [Online]. Available: https://www.cnet.com/tech/services-and-software/after-musk-tweet-signal-and-telegram-see-millions-of-new-downloads/

[3] J. Yan and J. Kaur, “Feature Selection for Website Fingerprinting,” *Proceedings on Privacy Enhancing Technologies*, vol. 2018, no. 4, pp. 200–219, Oct. 2018. [Online]. Available: http://content.sciendo.com/view/journals/popets/2018/4/article-p200.xml