Demo: PhyForm - A Cloud SDR Framework for Security Research Supporting Machine Learning of Wireless IoT Signal Data Sets

Antony Chung
School of Computing and Communications / Security Lancaster
Lancaster University, UK
a.chung@lancaster.ac.uk

Abstract

Software defined radio (SDR) enables the use of digital signal processing (DSP) to identify IoT security issues based on waveform analysis. Such research requires the handling, processing and interaction with large data sets of digitised RF. Those supporting activities are a high overhead.

An extensible framework is introduced for the curation, filtering, pre-processing, and analysis tasks associated with RF data sets in machine learning and IoT research. It provides a web interface, API, SigMF data sharing and integration with GNU Radio. The aim is improved data set and algorithm collaboration. A LoRa example provides context.

1 Introduction

Many Internet-of-Things (IoT) and Machine-to-Machine (M2M) systems use wireless communication. This is often more exposed to attack than wired solutions. For system operators to benefit from this technology they need to control threats like malicious data injection. It may not be possible to depend solely on cryptographic keys or device integrity.

Other sources of data can provide situational awareness that might help to identify security breaches and give the confidence network operators need in secure environments. Waveform analysis is one approach to provide detection of physical anomalies potentially linked to malicious activity on a wireless network, thus augmenting cryptography[1].

A received signal is a function of the overall system including transmitter, channel and receiver. The waveform can be influenced by manufacturing differences, device health and the propagation environment. Development of algorithms that isolate those properties and provide security indications is complex. It requires data analysis taking into account changes over time and noise (artificial and natural).

Machine Learning (ML) on features derived from waveforms can distinguish conditions directly[2] or the models can be evaluated to design new algorithms. The large data sets for training must be collected and *pre-processed* (Fig. 1).

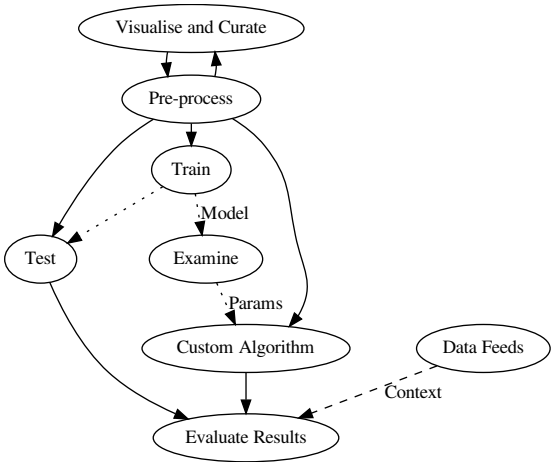

Figure 1. Typical steps in RF machine learning.

Good application-centric pre-processing is crucial given that overall performance depends on quality inputs at every stage. The data needs visual curation to group data and remove bad data (i.e. due to interference). Filtering must refine quality and reduce volume. Features need locating, extracting and processing. High data volumes make this hard.

Algorithms tend to be less optimal in the early stages of development, which compounds the problem by requiring overnight or distributed processing. RAM limitations, reboot risk, and iterative code debugging motivate persistent caching of outputs to avoid redundant reprocessing.

A review of this workflow indicated opportunities for a generic framework providing visual RF examination, curation, output caching, data feed (ground truth) rendering and distributed processing. This would free researchers to focus on DSP or data science, plus help data and algorithm sharing.

2 Architecture

PhyForm can run on a single machine, remote system, or a cloud. It comprises of a controller, file server and workers (Figure 2). Most user interaction is via a web browser.

International Conference on Embedded Wireless Systems and Networks (EWSN) 2020
17–19 February, Lyon, France
ISBN: 978-0-9949886-4-5

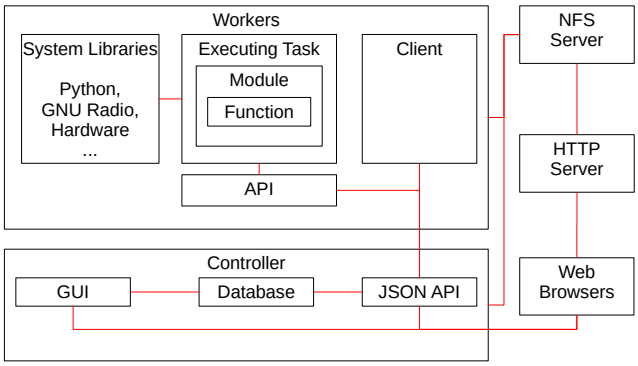

Figure 2. Structure of the system

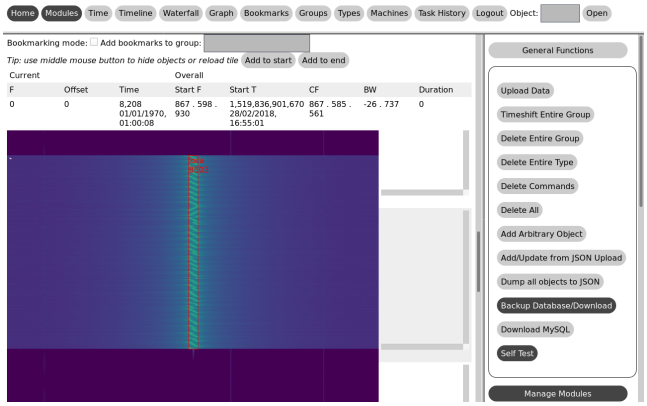

Figure 3. Interactive ISM band interference assessment

Waveform IQ data is captured from hardware or loaded from files. Signal processing and data analysis is bundled into modules, which are zip files containing code plus XML to extend the GUI. From a data science perspective, PhyForm is object-oriented. Algorithms create and manipulate typed objects, which are linked to files like IQ or graphic output.

Users can use Python or HTTP JSON APIs to develop modules. The APIs assist object and IQ handling, and provide features like distributed processing. The architecture allows for local development using remote data before modules are uploaded. GNU Radio blocks and interfaces allow for the use or adaptation of conventional flow graphs.

3 Test Case and Example Results

The initial objective was to differentiate - without demodulation - LoRa devices deployed across 1 km^2 of our campus.

RF recorders remotely triggered node transmission and captured complex IQ waveforms from USRP X310 hardware. Hundreds of recordings per experiment, node and configuration (power, spreading, etc.) led to thousands of "rough cut" objects that encapsulated transmissions.

Algorithms extracted preamble features from the rough cuts. These were curated visually in the GUI (Figures 3 and 4) to form data sets. Further processing produced measurements, which were aggregated into ML inputs for processing on platform (using TensorFlow) or export to MATLAB.

Figure 5 shows a typical confusion matrix using the exported data with MATLAB to produce a KNN trained using

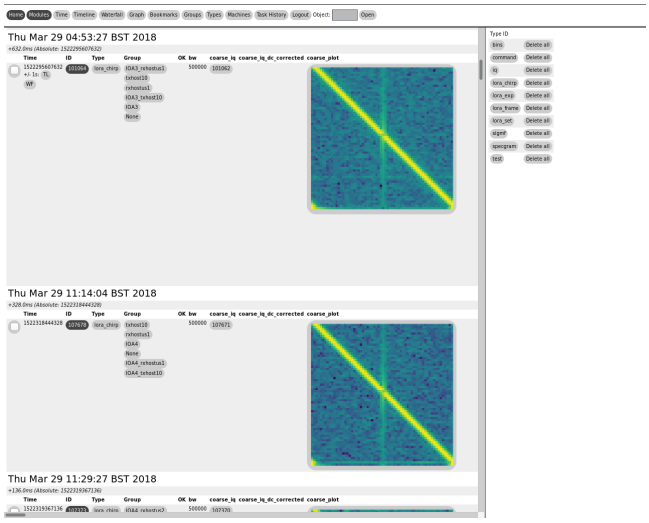

Figure 4. Curation of LoRa preamble features (chirps)

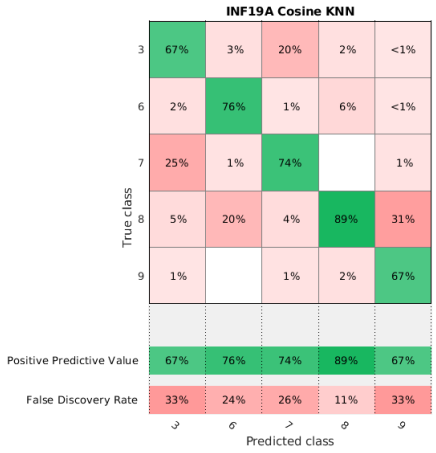

Figure 5. Example k-nearest neighbour classification

one dataset and tested with another. It compares the actual and predicted classification for a 5 node experiment and is useful to determine the impact of changes to pre-processing algorithms. As the intention of this demo is to demonstrate PhyForm, further evaluation of the data is omitted.

4 Conclusions and Future Work

PhyForm simplified the handling and processing of bulky IQ data sets for waveforms research. It helped with repetitive activities such as pre-processing of rough cuts to extract features. Data retention and movement requirements were reduced, and iterative workflows optimised, due to the cloud architecture and object persistence. This generic framework can enable collaboration with data sets and algorithms.

5 References

[1] C. K. Dubendorfer, B. W. Ramsey, and M. A. Temple. An rf-dna verification process for zigbee networks. In *MILCOM 2012 - 2012 IEEE Military Communications Conference*, pages 1–6, Oct 2012.

[2] S. Gopalakrishnan, M. Cekic, and U. Madhow. Robust wireless fingerprinting via complex-valued neural networks. In *IEEE Global Communications Conference (Globecom) 2019*, 2019.

Demo Information

The presenter will demonstrate how to capture a LoRa transmission, upload it into the system, check the quality of the recording, filter it, and then pre-process individually or as part of a group to obtain features. This will demonstrate a variety of capabilities, principally using a remote system.

For more interested attendees, the presenter can discuss the API and walk through a number of Python examples.

The demonstration requires a table, four power sockets and Internet access. The presenter will bring up to two laptops and other equipment. It would be preferable for there to be a TV screen and/or an elevated table to ease interaction with larger audiences over extended periods, but these need to be provided by the conference venue as it will be impractical to bring these.