Development and Potential Designs for an Opto-Electronic Fingerprinting System

Vikash Modi and Jack Schenkman

September 2019

Introduction:

Radio frequency (RF) signals are all around us, in WiFi, our phones, etc, however as we become more reliant on wireless devices, we need to make sure that these signals, essential for communication, are secure. Unfortunately, due to their complex nature, the process of identifying the device from which an RF signal was emitted can be computationally expensive and time-consuming. This process of RF fingerprinting is critical for ruling out falsified, malicious signals. Many neural network fingerprinting approaches have been developed, but an accurate, efficient solution has yet to be found.

A primary constraint of these previous techniques is that they require a significant amount of processing of the raw RF signal, including analog to digital conversion. Our lab proposes that by using the field of neuromorphic photonics (physical photonic neural networks) in conjunction with recent developments of field programmable array (FPGA) boards, we can greatly reduce the signal processing times compared to the time that traditional CPU or GPU fingerprinting techniques require. Over the summer we focused on developing a neural network algorithm, for an eventual photonic chip, that would read and fingerprint RF data in its raw analog form. We developed a python framework, as well as several methods for further exploration during this year. If successful, this project could lead to near real-time RF fingerprinting, a significant step in cybersecurity in our modern world.

Summer Overview:

Our goal was to build an optimal algorithm that would eventually be implemented in an optoelectronic (photonic and electronic) system for near real-time radio frequency (RF) signal fingerprinting. Our summer work consisted of two phases. During Phase 1 we focused on developing a Python simulation of a Time-Delayed Neural Network (TDNN) that could be used for chaotic time series prediction. In the second phase our focus was developing an optimal (simulated) algorithm using our Python TDNN, along with RF signal data, and a modified Convolutional Neural Network (CNN) signal detector. Our work is part of a project being pursued in the Lightwave Communications Laboratory.

Phase 1:

After conducting an extensive literature review we implemented a time delay reservoir using a Python script. The reservoir is modeled by the Mackey-Glass equation which is a delayed differential equation. This type of differential equation is useful for two primary reasons. Firstly, the delay component ensures that previous inputs have an impact on the current state, thereby giving the system a form of memory. Secondly, the seemingly chaotic behavior of the system projects the input data onto a much higher dimensional space, essentially functioning as a kernel. The vector representing the state of the Mackey-Glass system within a given time delay is multiplied by a weight vector to yield output. Only this weight vector connecting the reservoir states to the output is modified during training. To assess the performance of our simulation we calculated how accurately the reservoir could predict the values of the NARMA10, a chaotic time series. We achieved a normalized root mean square error of less than 0.2.

Phase 2:

After reproducing results on the feasibility of time delay reservoirs for time series prediction tasks we proceeded to apply the reservoir algorithm to radio frequency (RF) fingerprinting. A former Princeton student gathered RF data and built a convolutional neural network (CNN) to classify the device from which a signal was emitted for his senior thesis. We assessed 4 primary applications of the reservoir system in relation to RF fingerprinting: direct classification, target-basis creation, noise reduction, and frequency extraction. The first two methods were not very successful, so we will focus on the latter two.

1. Noise reduction: After training the reservoir to increase the signal to noise ratio (SNR) in simple sinusoidal waves, we found that the reservoir was able to increase the SNR in RF signals. In our preliminary trials, utilizing the reservoir in conjunction with the CNN did not improve fingerprinting accuracy, however, we faced other bottlenecks that could have hindered this process. We are actively working on interfacing these systems.
2. Frequency extraction: We assessed the ability of the reservoir to filter a select frequency bandwidth from a type of sinusoidal wave. We found that the reservoir output more closely matched the desired waveform in the time domain than the input waveform matched the desired waveform. Filters that can be trained to a select set of frequencies are desired for a wide variety of applications.

Roadblocks:

Memory management posed a major challenge in applying the reservoir simulation to the RF data. Each RF packet had 30,000 samples and the data set consisted of over 5,000 packets. To process such large volumes of data, we used a graphics processing unit (GPU). Utilizing the GPU allowed us to parallelize some mathematical operations such as matrix multiplication. In addition, because the system was prone to frequent crashes in which all results from a given session were lost, we built and included software to automatically save outputs on a regular basis. With these measures, we were able to significantly increase the scalability of our simulations. Since processing such large volumes of data has posed a challenge, we utilized an echo state machine to develop a preliminary adaptive sampler. We are currently testing this apparatus on basic sinusoidal waves with the hopes of moving towards more complex signals.

Future Steps:

We will be part of a team simulating the reservoir using a field-programmable gate array (FPGA) and the hardware description language Verilog. After these simulations, we will be interfacing the FPGA with a photonic neural network for real-time classification of RF signals.