Summer 2024

Radiofrequency Interference Detection using Lstmand Statistical Analysis Discriminator

Luke Smith
Missouri University of Science and Technology

Department:

RADIOFREQUENCY INTERFERENCE DETECTION USING LSTM AND STATISTICAL ANALYSIS DISCRIMINATOR

by

LUKE ANDREW SMITH

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

in

COMPUTER SCIENCE

2023

Approved by:

Sanjay Madria, Advisor
Maciej Zawodniok
Siddhardh Nadendla

PUBLICATION THESIS OPTION

This thesis consists of the following articles, formatted in the style used by the Missouri University of Science and Technology:

Paper I, found on pages 6–37, has been accepted by *IEEE MDM 2023*.

ABSTRACT

Wireless devices are becoming increasingly pervasive across all aspects of society. Examples of such devices include radios, routers, mobile phones, tablets, and more. As the number of radio frequency (RF) devices continues to rise, so does the amount of interference and noise increase. This is why an efficient approach to interference detection is explored. Most research within this area has been done strictly within the frequency domain as viewing a signal within this domain provides many insights into what makes the signal. This has, however, led to the time domain being underutilized for this area of research.

To explore the time domain and its uses within radio frequency interference (RFI) detection we propose a lightweight program that requires knowledge of the known set of RF devices. The program utilizes a Long-Short Term Memory model to simulate a known radio set; it does this by training on a set of known signals interfered with each other. A custom statistical discriminator is then used to compare the simulated signal to the received signal. The output bounds of interference are then observed to determine how accurately our model detects and localizes interference.

ACKNOWLEDGMENTS

Firstly, I would like to thank my advisor, Sanjay Madria, for his support, advice, and patience with my work. Also, by guiding me through my journey and teaching me the best practices for clear, expansive, and beneficial research. I would also like to express my sincere gratitude to Vishesh Tanwar and Maciej Zawodniok. Their comments, recommendations, and encouragement have contributed greatly to my success and education, and I could not imagine having made it without them. Throughout my research, I learned numerous things and explored areas I had not even thought of before thanks to their guidance. Finally, I would like to thank the Computer Science department, including all of the faculty and staff, for giving me the resources necessary to continue my work and further my studies.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

PAPER I Page

1. INTRODUCTION

Radio Frequency (RF) devices are increasing in number with an average of 9 devices per household as of 2022. RF devices permeate every aspect of modern existence, be it an Alexa on the counter or a medical telemetry device measuring a patient's vitals. These devices have become a necessity for many and tied in with the strictly increasing number of devices how we handle interference and noise within the RF spectrum becomes ever more important.

Several approaches to RFI detection exist, predominantly within the frequency domain. The frequency domain represents a signal as relative amplitudes across a set of frequencies. This is especially useful for seeing which frequencies at what amplitudes make up a signal and is very effective for determining if interference is present within a signal. This approach, however, relies on the assumption that you have the resources and time to perform a Fast Fourier Transform (FFT) on a signal. The frequency domain also neglects any relations across time a signal may have. When dealing with high-density RF environments, the cost of an FFT becomes apparent, which led us to explore time domain detection.

The time domain poses challenges of its own, namely signal complexity is increased due to a more apparent Signal-to-Noise ratio (SNR), and any signal can, in theory, be broken down into sub-signals thus it is a difficult problem to determine what signals went into creating a received signal. To address these challenges a few approaches and avoidant measures were taken. With these challenges addressed, time domain detection will be shown to be a resource-efficient alternative to frequency domain detection in live-radio scenarios.

In the following paper, we will demonstrate several aspects of time domain Radio Frequency Interference Detection (RaFID). Such aspects include:

- The ability of an LSTM to simulate a known radio/radio set.
- Overcoming time domain learning challenges with statistical detection.
- Comparison of complexity between the FFT and our custom statistical detection.

2. LITERATURE REVIEW

There are a plethora of machine learning approaches designed for anomaly detection and each holds its own benefits and costs [19]. Such requirements may include more strenuous preprocessing, larger data requirements, and so on. Out of all of the machine learning approaches, however, none surpass those within the realm of deep learning [12]. While many deep learning architectures can be used with decent effectiveness, the two most seen structures for anomaly detection are LSTMs [16][17] and CNNs [16], or variants thereof [1][18]. Generally, LSTMs are utilized for time domain datasets while CNNs are used for frequency domain datasets. There are some CNN variants that are built for the juxtaposition of these two domains [18] and they have seen great success within anomaly detection.

The CNN structures explored in [1] are the YOLOv3 model and a Convolutional Auto-Encoder (CAE) [7]. Briefly, YOLOv3 is a deep 1x1 convolutional neural network that classifies objects in an image or video, and CAEs utilize convolutional neural networks to facilitate the encoding and decoding of the Auto-Encoder; auto-encoders [20] are unsupervised models that attempt to encode input and then decode an output as similar to the input as possible utilizing the generalized pattern extracted from the encoding process. These approaches experience reduced performance when encountering low SNR and SIR, a weakness shared by many CNN derivative structures. Overall, YOLOv3 and CAEs demonstrated 89% and 78% precisions, respectively. The TCN [18] structure demonstrates a great ability to detect anomalies demonstrating a higher recall than that of its CNN or LSTM counterparts. It does come with a high cost of requiring data be in a time-frequency domain format which is a level of processing that can’t be

guaranteed in the majority of environments. This is what brings us to LSTMs as they are more lightweight than their CNN counterparts. LSTMs tout an exceptional ability to detect anomalies in sequential data sets [6] and, standardly, have a faster run-time than CNNs when restricted to CPU processing; it is essential to note that with GPU processing, a CNN runs much faster than an LSTM [21]. While LSTMs are great for time series data estimation and anomaly detection, they can begin to underperform in the presence of high-complexity data sets (i.e., RFI). Our work addresses this issue by batching data into manageable sizes for an LSTM. This adjustment serves to address learning difficulties seen with LSTM detection [18] by offloading detection from the LSTM.

Statistical analysis as a means of RFI detection has seen considerable success, as seen in [9] and [22]. A common trend in these papers is the treatment of a signal as a probability space. The first paper covers using Eigenvalue Analysis to detect RFI, specifically within the space environment. They successfully utilized a maximum-to-minimum eigenvalue (MME) ratio to see if RFI was occurring. They outperformed other techniques that rely on full-band or spectral kurtosis analysis [23]. On the other hand, [22] explores using probability density function (PDF) moment calculations to determine if RFI is present. This paper demonstrates that the formulated approach works well with sinusoidal signals with a duty cycle of less than 50%. They note that there is still work regarding other signal parameters. Another approach utilizes compressive statistical sensing to detect and mitigate RFI [24]. This approach abuses the periodic nature of RF signals to detect where interference may occur with second-order statistical analysis. A method of detection created by Schoenwald et al. [23] investigates using Independent

Component Analysis (ICA) preprocessing with kurtosis as a test statistic to detect interference in an RF signal. The further the kurtosis value from 0, the more likely that interference was in a signal.

While these approaches show promise for RFI detection, they are very strict and limited on what problems and spaces they can apply to and within. A primary limitation of these approaches is the difficulty of detecting interference in a signal where interference is present for a majority of the signal as well as with low SNR and SIR. This is expected as their basis for comparison is a short-term measure of the current signal. With low SNR and SIR, you will see an increase in difficulty with detection as the changes they make to any signal at any given time step are observably small. Thus, a clean basis of comparison is evermore necessary for discrimination. Our approach addresses this weakness of statistical detection by creating a clean signal LSTM generator that simulates a given radio set. This allows us to provide a clean basis of comparison for statistical detection which opens the door for low SNR and SIR detection.

Overall, while many approaches, both machine learning and statistical, exist, very few combine the strengths of both to overcome their limitations. Through this juxtaposition of anomaly detection, the realm of time-domain learning is exhibited as comparable to that of its frequency-domain counterpart.

PAPER

I. RAFID: A LIGHTWEIGHT APPROACH TO RADIO FREQUENCY INTERFERENCE DETECTION IN TIME DOMAIN USING LSTM AND STATISTICAL ANALYSIS

ABSTRACT

Recently, the utilization of Radio Frequency (RF) devices has increased exponentially over numerous vertical platforms such as medical instrumentation, airplane control system, computing hardware, smart homes, etc. This rise has led to an abundance of Radio Frequency Interference (RFI) that continues to plague RF systems today and can significantly disrupt the normal functioning of RF-incorporated devices. The continued crowding of the RF spectrum makes RFI's efficient and lightweight mitigation more critical. Detecting and localizing the interfering signals is the foremost step for mitigating RFI concerns. Addressing these challenges, we propose a novel and lightweight approach, namely RaFID, for detecting and localizing the RFI by incorporating deep neural networks and statistical analysis via batch-wise mean aggregation and standard deviation calculations. The proposed RaFID approach investigates the generation of an expected signal using deep neural networks, specifically convolutional neural networks (CNN) and long short- term memory (LSTM), within the time domain only. We generated the RF data using the Phase Shift Keying modulation scheme to evaluate our scheme. In addition, we performed the statistical analysis to compare our generated expected signal with the received signal to detect the existence of interference and determine interference frequency. Experimental results show that signal estimation is

accurate, with a mean squared error of 0.012 and an average run-time of 0.5 seconds. Further, RaFID locates the interference within a tenth of a second from the occurrence location and detects interference in environments with a Signal-Noise-Ratio of more than 1.75.

1. INTRODUCTION

The fast growth of society and the progression toward a more wireless world have led to a steadily growing number of Radio Frequency (RF) devices corresponding to an increased crowding of the RF spectrum. The overpopulation of machines within the finite RF spectrum has led to the growth of Radio Frequency Interference (RFI)[1]. Caused by a variety of issues such as unknown signals and noise, RFI adversely affects wireless performance on all fronts. It is a critical weakness that RF devices must be able to efficiently detect and ignore or remove, which poses three significant challenges: (i) it is functionally impossible to determine every source of interference within a signal, much less determine the potential behaviors of each point of interference [1], (ii) the problem is only exacerbated by the dynamic and sporadic nature of the interference [2], and (iii) the amount of RFI present in most systems causes the relative Signal-Noise-Ratio (SNR) and Signal-Interference-Ratio (SIR) to become very small [3]. Researchers have developed software and hardware for detecting RFI to address these challenges. Hardware approaches include such things as band-pass filtering and the use of spectrum analyzers

[1] https://cpb-eu-w2.wpmucdn.com/blogs.bristol.ac.uk/dist/6/635/files/2021/05/SWANWH2.pdf

[4]. Hardware approaches demonstrate superior performance in detection, whereas software-based techniques provide unparalleled flexibility over their hardware counterparts [5].

Our work focuses primarily on the software aspect as we propose a deep learning approach that functions in a lightweight manner as long as minimum processing requirements are met. The RFI detection software methods can be categorized into (i) Machine Learning and (ii) Statistical Analysis. Machine learning approaches involve well-established model structures with common approaches involving the deep neural networks, most standardly the LSTM [6], and CNN [1], [7]. In brief, these approaches utilize classification for anomaly detection. They train a model on clean data without interference and then run the models on a received signal where the signal is classified as having interference. Some advantages of deep learning as an approach to RaFID include model flexibility, applicability to highly-complex problems, and recognition of underlying patterns. While effective at problems of this nature, disadvantages to RFI detection with deep learning exist. For instance, deep learning models typically have a much longer run-time when compared to their non-deep learning counterparts and require adequate time to train models. This is due to the high dimensionality needed for creating a deep learning model. In contrast, within the area of Statistical Analysis, there exist several methods of anomaly [8]. One such approach to RFI detection utilizes eigenvalues to find points of interference in space [9]. This
approach calculates maximum and minimum eigenvalues to determine if interference exists. Given the success of both deep learning and statistical methods in RFI detection, we aim to combine the two fields in such a manner as to accentuate the benefits of each

while minimizing the detriments. By utilizing deep learning for signal estimation and statistical analysis via batch-wise mean aggregation, we can create a dynamic and flexible basis of comparison for use with our statistical detection method.

1.1. MOTIVATIONS AND PROBLEM STATEMENT

In 2022, there were 15.96 billion mobile devices in the world, projected to be 18.22 billion in 2025 [10]. The steady increment of RF devices poses various threats to RFI towards the existing wireless communication systems. Increased amounts of RFI can make it hard to discern between trusted and untrusted devices. For instance, the average American has 13 wireless devices (i.e., cell phones, laptops, tablets, smart tv, etc.) in their household [11]. In households where at least this amount of devices are present, it can be hard to detect when an intrusive or unknown device is within the same environment. In the case of aircraft or unmanned aerial vehicle communications, if multiple aircraft are present in an environment, it may become difficult to determine when an unknown airliner enters the same airspace, and the ability to efficiently and actively detect interference becomes even more paramount. While considerable research has been performed in the realm of RFI detection [12][13], a large portion of that research has been conducted within the frequency domain [14] and left the time domain neglected. The time domain poses several challenges with RFI detection, including a weakness to highly-complex data and noise. Still, these challenges can be circumvented in such a manner as to provide a level of improvement in efficiency by avoiding the execution of the Fast Fourier Transform (FFT), which runs in logarithmic time [15]. This paper explores the efficacy of utilizing deep neural networks and statistical analysis with raw

time-series signals to detect and locate RFI within an environment efficiently and actively.

1.2. CONTRIBUTIONS

The core contributions of our proposed work are:

- We proposed a detection and localization system for Radio Frequency Interference Detection, RaFID for the time domain aspects rather than the frequency domain by leveraging the deep neural network specifically LSTM.
- We investigated using the LSTM to generate the expected signal for a set of known radios. This was done by interfering with known signals onto each other and batching the composite signal for use with training. The model would then predict the next batch provided the current batch of signals.
- We investigate utilizing batch-wise mean aggregation and standard deviation calculations to detect where interference occurs by comparing the received signal to our expected signal at each batch. This allows for an active approach to detection and localization.
- The performance of the proposed approach is analyzed using simulated RFI data. This data is simulated using Monte Carlo generation and M-Phase Shift Key (M-PSK) modulation. The interfering signal additively interferes with the known signal set in a randomized time window, and the model then estimates where the interference occurs.

Paper organization: Section II explains the related work on RFI, and the preliminaries of LSTM and signal estimation is provided in Section III. Section IV is devoted to our

proposed methodology, RaFID, followed by the experiments and discussion in Section V. The paper is concluded with future work in Section VI.

2. RELATED WORK

In this section, we discuss various approaches for RFI detection based on machine learning and statistical paradigms.

2.1. MACHINE LEARNING-BASED APPROACHES

Two primary deep learning frameworks are utilized for RFI detection (or, more generally, anomaly detection [12]). Namely, recurrent neural networks (RNNs) and convolutional neural networks (CNNs) are explored in [16]. In brief, the LSTM [17], and TCN [18] perform comparably with equivalent layer counts. The LSTM takes longer to train but holds a higher precision score than the TCN. Approaches such as Naive-Bayes, decision trees, and K-NN clustering [19] have promising results but require precise and extensive preprocessing. While helpful in advancing RFI detection, these machine learning (ML) approaches are generally outclassed by deep learning methodologies [12] when comprehensive preprocessing is not possible or available; specifically CNN and RNN structures.

The CNN structures explored in [1] are the YOLOv3 model and a Convolutional Auto-Encoder (CAE) [7]. Briefly, YOLOv3 is a deep 1x1 convolutional neural network that classifies objects in an image or video, and CAEs utilize convolutional neural networks to facilitate the encoding and decoding of the Auto-Encoder; auto-encoders [20] are unsupervised models that attempt to encode input and then decode an output as

similar to the input as possible utilizing the generalized pattern extracted from the encoding process. These approaches experience reduced performance when encountering low SNR and SIR. Overall, YOLOv3 and CAEs demonstrated precisions of 89% and 78%, respectively.

LSTMs have been widely used in anomaly detection and for a good reason. They tout an exceptional ability to detect anomalies in sequential data sets [6] and, standardly, have a faster run-time than CNNs when restricted to CPU processing; it is essential to note that with GPU processing, a CNN runs much faster than an LSTM [21]. While LSTMs are great for time series data estimation and anomaly detection, they can begin to underperform in the presence of high-complexity data sets (i.e., RFI). Our work addresses this issue by batching data into manageable sizes for an LSTM.

2.2. STATISTICAL ANALYSIS-BASED APPROACHES

Statistical analysis as a means of RFI detection has seen considerable success, as seen in [9] and [22]. A common trend in these papers is the treatment of a signal as a probability space. The first paper covers using Eigenvalue Analysis to detect RFI, specifically within the space environment. They successfully utilized a maximum-to-minimum eigenvalue (MME) ratio to see if RFI was occurring. They outperformed other techniques that rely on full-band or spectral kurtosis analysis [23]. On the other hand, [22] explores using probability density function (PDF) moment calculations to determine if RFI is present. This paper demonstrates that the formulated approach works well with sinusoidal signals with a duty cycle of less than 50%. They note that there is still work regarding other signal parameters. Another approach utilizes compressive statistical

sensing to detect and mitigate RFI [24]. This approach abuses the periodic nature of RF signals to detect where interference may occur with second-order statistical analysis. A method of detection created by Schoenwald et al. [23] investigates using Independent Component Analysis (ICA) preprocessing with kurtosis as a test statistic to detect interference in an RF signal. The further the kurtosis value from 0, the more likely that interference was in a signal. While these approaches show promise for RFI detection, they are very strict on what problems and spaces they can apply to and within. A primary limitation of these approaches is the difficulty of detecting interference in a signal where interference is present for a majority of the signal as well as with low SNR and SIR.

3. PRELIMINARIES

This section consists of preliminary information that may be useful in understanding some of the aspects of this paper.

3.1. LONG SHORT-TERM MEMORY (LSTM)

LSTMs are a type of RNN that utilize the RNN structure with an added Short-Term element. LSTMs use a cell state to maintain long-term learning; this cell state adjusts without weight or bias. Short-Term Memories lie within the hidden state of the cell; these are changed with weights and biases. In short, long-term memory allows older inputs to affect current estimations but also takes a more remarkable account of local trends in the inputs with short-term memory. This combination of long and short-term memory allows the LSTM to mitigate the vanishing/exploding gradient within basic

RNN structures. For more information on LSTM, please refer to [17]. LSTMs use gates of 3 types: input, output, and forget. The generic equations for each are as follows:

$$\begin{aligned} i_t &= \sigma\left(w_i\left[h_{t\_1}, x_t\right] + b_i\right) \\ o_t &= \sigma\left(w_o\left[h_{t\_1}, x_t\right] + b_o\right) \\ f_t &= \sigma\left(w_f\left[h_{t\_1}, x_t\right] + b_f\right) \end{aligned} \tag{1}$$

For these equations, w_i denotes the weight for the respective gate, h_{t-1} denotes the output of the previous LSTM block at time t−1, x_t denotes the current timestamp's input, bi denotes the bias for the respective gate, and σ denotes the sigmoid activation function. The role of these gates is to determine if a feature is kept for further use. The input gate determines which current features to keep and which to ignore, the forget gate determines which previous state features to keep and discard, and the output gate controls the output of the current cell state. The current cell state, candidate cell state, and final output are calculated as follows:

$$\begin{aligned} \tilde{c}_t &= tanh\left(w_c\left[h_{t\_1}, x_t\right] + b_c\right) \\ c_t &= f_t * c_{t\_1} + i_t * \tilde{c}_t \\ h_t &= o_t * tanh(c_t) \end{aligned} \tag{2}$$

For these equations, $\tilde{c}_t$ denotes the current cell state, c_t denotes the candidate cell state, and h_t denotes the final output. With this information, a given cell state determines which information is necessary and which to forget, hence the Long Short-Term name.

3.2. SIMULATED SIGNAL ESTIMATION

Our methodology uses an LSTM architecture to learn a known radio set and estimate an expected signal based on the received signal. This estimation provides a basis

for comparison between received and expected signals. LSTMs allow us to make a simple model that applies to most if not all, signals modulated with M-Phase Shift Key (M-PSK). It is important to note that further experimentation needs to be performed on other modulation schemes to confirm if this model structure also works for them (QAM – Quadrature Amplitude Modulation, AM - Amplitude Modulation, FM - Frequency Modulation, etc.).

$$S_n(t) = \sqrt{2E_s/T_s}\cos(2\pi f_c t + (2n-1)\pi/M) \quad (3)$$

Eq. 3 represents modulating a signal with an M-PSK modulation scheme mathematically. Es denotes the energy of the waveform, Ts denotes the duration of the signal, fc denotes the carrier frequency expressed as an angular frequency, t denotes the current time index, n denotes the symbol duration, and M denotes the number of phases.

4. PROPOSED METHODOLOGY

This section introduces our proposed radio frequency interference detection scheme, RaFID, followed by the method of simulated data generation.

4.1. RAFID SCHEME

We propose a supervised learning approach for quick and efficient RFI detection. While convolutional neural networks (CNNs) have been successfully employed in RFI detection, they need raw time-domain signals to be converted into frequency-domain using Fast Fourier Transformation (FFT). Using the FFT, a time domain signal can be quickly converted to the frequency domain with an x-axis of Frequency and a y-axis of decibels. The frequency domain provides insights into constructing a signal outside of

time, making detecting and localizing interference quite simple. While effective, the FFT algorithm runs in logarithmic time, disregarding other preprocessing that may be necessary. This cost of CNN leads us to leverage the LSTM to detect RFI in the time domain only. It is important to note that while the frequency domain does demonstrate essential aspects of the signal mathematical (i.e., energy distribution over a range of frequencies, phase-shift information), these mathematical factors remain accounted for in our overall method rendering the use of a CNN unnecessary. The pictorial presentation of the RaFID approach is presented in Figure 1.

To perform the comparative statistical analysis, we must first provide a statistical distribution for our received signal to be compared against. We train an LSTM on a given known radio set to define this distribution. Utilizing Eqs. 1-2 defined in Section III-A, we can further understand the LSTM portrayed in Figure 1. From left to right, each part is as follows: forget gate, input gate, current cell state, output gate, candidate cell state, and final output. This LSTM model is then used to estimate a signal given a small sample of the received signal, roughly 10 data points. Our approach assumes that the first batch of data is without interference which can be guaranteed and continues until a signal is not needed to compare. Under these assumptions, the first batch is considered to have no interfering signals and is used to initialize our saved mean squared error (MSE) set, say μ. This also allows us to run this model actively alongside a signal being received. Once the expected signal is generated, it is compared to the received signal in batches of 10 input sets. The pseudo-code for our proposed scheme is provided in Algorithm 1.

Algorithm 1 Statistical Detection
Input: 1 batch from received and expected signal
Output: Indices for interference start and stop
1: **while** *Signal being received* OR *Radio on* **do**
2: Calculate MSE of *Received* and *Estimated* signal
3: **if** *isInitial(curr_signal_batch)* **then**
4: *MSE_List[0] = MSE*
5: CONTINUE
6: **end if**
7: Calculate Mean of *MSE_List* as μ
8: Calculate Standard Deviation of *Received*
9: **if** $\sigma < 0.2$ and *history_avg* < 0.3 **then**
10: *MSE_List*.append(*current_batch*)
11: **else**
12: Save current batch number/index
13: **end if**
14: **end while**

Figure 1: Psuedo-code for our statistical discriminator.

The distribution obtained by LSTM training is utilized to calculate the current batch's standard deviation, say σ, from our pre-saved list of batches without interference. With a sufficiently strict standard deviation, we can accurately detect and capture when the interference starts and ends within 10-time steps (which can be fractions of a second dependent on measurement rate) and runs in a very lightweight manner. Once the detection has occurred, the data can be passed to desired ML models to process aspects of the interfering signals for further classification.

This method for the statistical detection of interference was created by analyzing how signals can be broken down into sums of cosines utilizing Discrete Cosine Transforms (DCTs). This allows us to treat each signal as a sum of cosines and, thus,

each batch of a signal as a potential cosine wave. Another critical part of our statistical approach is treating each signal as a distribution and each signal batch as a sub-distribution. Given our treatment of each signal batch as a cosine wave, we consider each signal batch as a cosine distribution; it should be noted cosine distributions are effective approximations for normal distribution, allowing us to treat each batch as its Gaussian distribution to improve computational complexity. After evaluating the most similar distributions of the received signal to our saved batches, we can detect batches having distributions outside what we have defined as known. More explicitly, we are calculating a batch-wise mean aggregate for each batch. This results in a single mean value that functionally encodes the statistical aspects of each batch. At each step, the standard deviation of the current batch is calculated concerning the previous batches, and the three most recent standard deviations are stored as a short-term history which is averaged to create $history_{avg}$ (as seen in Algorithm 1). This history of standard deviations assists in avoiding misrepresenting a spurious drop in difference as the end of interference. The current standard deviation is then compared to the ceiling accepted value of 0.2, and the current $history_{avg}$ is compared to 0.3. If the conditions are met, then the current batch is added to the list of saved batches as it closely resembles those batches, whereas if the conditions are not met, then the current index is stored as the start of an interfering signal. When the conditions are met again, the index is stored as the end of an interfering signal, and the algorithm continues to execute in the same overall manner. This approach to RaFID detection introduces localization as a byproduct. The use of active detection and batching allows this approach to locate where a signal interference occurs reliably.

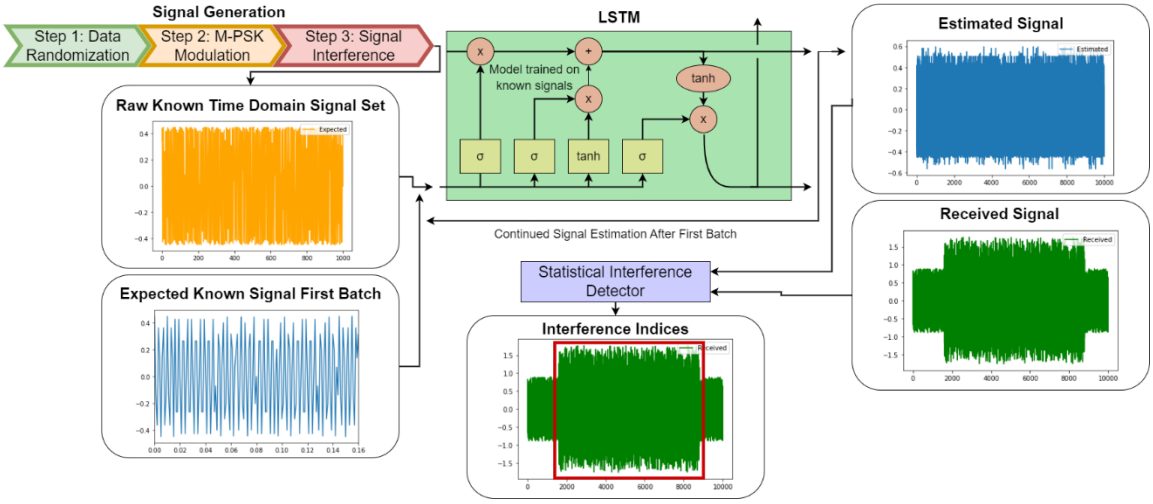

Figure 2: Our proposed architecture utilizes an LSTM to generate an expected signal using the previously generated batch to predict the currently expected signal batch. Each batch is compared to the current received signal batch. Interference has been detected when that comparison results in a large enough standard deviation.

4.2. SIMULATION DATA

Deep neural networks (DNNs) used to solve RF-based problems rely on a signal's indicative features like time-frequency domain data, angle of arrival, etc. Diversity in the dataset contributes to the robust learning of the model with the assumption of the availability of multiple feature types in all spaces. Our scheme looks at circumventing this assumption by only using the raw time domain signals that would be received directly by radio. However, time domain data inherits only a single type of feature, making a significant concern for training DNNs, due to the heavy time-based relationships within the signal. Therefore, to enhance the robustness of model training, the data is batched into batches of 10-time steps with a window shift of 10-time steps; these time steps are roughly a hundredth of each second. Moreover, we expand our single

feature input into a ten-feature input set where each feature is the next time-step, and each batch is the next ten time steps. Time domain data also suffers from various noises in real-world scenarios. To this end, our data set must replicate the noise experienced in the real world. Thus, we have included white Gaussian noise, with each signal acting as a representative noise experienced as a result of hardware abnormalities and faults.

To simulate real-world interference, we considered multiple signals overlapping when they interfere, ensuring our approach can detect interference even when multiple RF signals exist in the same time frame. Treating them as a single interfering signal in situations where multiple signals overlap in the same time frame is mathematically acceptable. Thus, the case of multiple signals interfering at multiple times with no overlap is adequately covered. The equation for signal interference and the equations for SNR, SIR, and SINR are as follows:

$$S_c = S_o + S_i \tag{4}$$

$$SNR = P_s/P_n \tag{5}$$

$$SIR = P_s/P_i \tag{6}$$

$$SINR = P_s/(P_i + P_n) \tag{7}$$

As is seen here, interfering with two signals is simply an additive process. For the equations above, Sc denotes the composite or interfered signal, So denotes the signal being interfered upon, Si denotes the interfering signal, Ps denotes the power of the signal being interrupted upon, Pn denotes the power of the noise, and Pi denotes the power of the interfering signal. It can then be inferred that as more interfering signals and noise occupy an environment, the lower the SNR, SIR, and SINR get relative to our chosen signal.

4.3. LIGHTWEIGHT ARCHITECTURE

To determine the lightweight characteristics of our proposed approach, we analyze various metrics such as memory usage, storage, model robustness, and run-time.

4.3.1. Memory Usage and Storage. We shall define good memory usage as small enough in demand so that our model can run on a UAV without hindering other operations to a detrimental degree. To test memory usage, the tracemalloc library in Python 3.0 is used.

After 20 executions of our program, we determined that our program used a ceiling value of 10.7 MB and an average of 9.9 MB of memory throughout signal estimation and interference detection, as reported in Figure 2. The reported memory could be further reduced by refactoring the code into a more function-based design. The current program only requires a ceiling of 3 Mb of storage, a significantly small program with relatively low memory requirements compared to other deep-learning models. This is achieved by reducing the hidden layer dimensionality of the LSTM through batching and avoiding costly preprocessing steps like converting to the frequency domain.

4.3.2. Robustness. We shall define robustness as the ability of our model to perform accurately and efficiently regardless of the relevant spectrum of the interfering signals to our known signal(s). To test the overall robustness of RaFID, we utilized Monte Carlo generation to randomly generate various signal and modulation parameters with a uniform distribution. Our modulation scheme is M-PSK, and our randomized parameters were M (the number of constellations, Eq. 3) and the carrier frequency, which is used for signal creation as stated in Eq. 3. We generated a few hundred signals, all of

which performed similarly to the signal used for the results displayed in Section V. This continued positive outcome demonstrates the robustness of our model in the time domain.

4.3.3. Run-time. We define an excellent run-time of an algorithm as the run-time taken in pseudo-real time with CPU processing. We evaluate our scheme's run-time with an input signal of 10,000-time steps (or 10 seconds of signal). Our model can process a signal of this size within a ceiling value of 9.5 seconds, indicating that the detection takes 0.95 seconds to process 1 second of data. Contrasting, the training time is a metric considered in DNNs-based approaches. Thus, it is essential to note that our approach works on the assumption that the model is trained beforehand on the central server, not actively, making the evaluation of training time irrelevant.

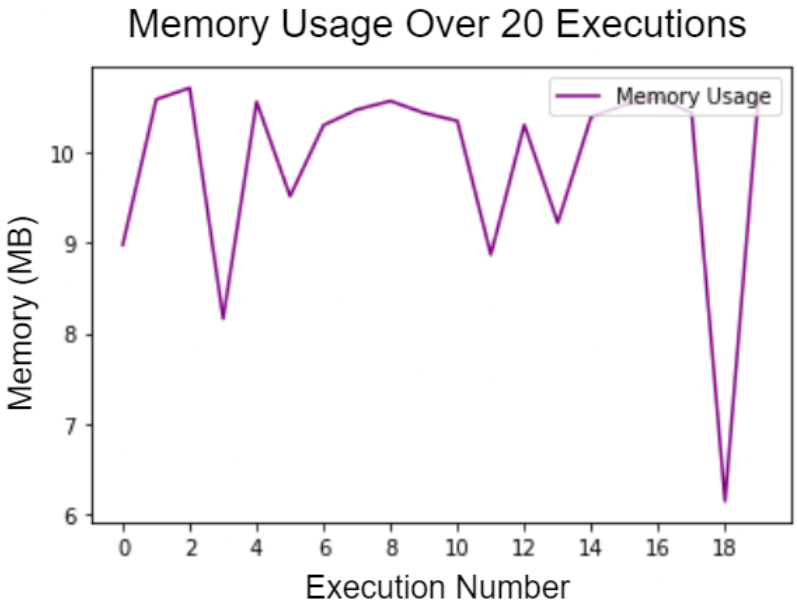

Figure 3: Memory usage over 20 executions of RaFID.

5. EXPERIMENTS AND RESULTS

In this section, we conducted various experiments to evaluate our proposed RaFID and perform a detailed discussion of the reported results.

5.1. SYSTEM REQUIREMENTS

We performed all our experiments on 64-bit Windows 10 OS, version 10.0.19044, build 19044, with AMD Ryzen 7 5800X 8-Core Processor and 32 GB RAM. The IDE is Visual Studio Code version 1.74 with a Python 3.10.4 Jupyter Notebook.

5.2. EVALUATION DATASET

Our data set comprises ten signals with 10,000 data points each that are generated randomly and modulated with an M-PSK modulation scheme and represent a total time frame of 10 seconds. The static hyperparameter values are an energy value of 1, sample rate of 1000, sample count of 10000, and samples per symbol of 16. The data points are generated with a discrete uniform distribution. For modulation, the carrier frequency and phase count are randomly generated to cover a wide range of potential signals with ranges of [10000, 20000) and [2, 10), respectively. Our train, validation, and test percentage spread is 40%, 15%, and 45%, respectively. This spread was used to demonstrate the ability of our model to apply learning to data it had never encountered.

5.3. RADIO ESTIMATION MODEL

First, we must show that we can accurately simulate a radio and estimate radio signals' appearance. We used an LSTM model trained for 200 epochs on our generated

known signal set. Our model consists of the linear activation function, ADAM optimization, and mean-squared error as a loss function. Initially, the model is trained with an environment of one radio, which takes input as ten-time steps of a time-series signal and outputs the next ten time steps. The known radio scenario is trained 100 times with Monte Carlo-generated signals and performed similar training throughout each execution.

Learning one radio signal is achieved with a simple deep-learning model. The main issue with one model per radio becomes apparent when considering UAV environments where multiple known radios exist within the same airspace. We avoid treating these radios as interfering radios. For this case, we tested training a single model on the complete known radio set signal, represented as one signal where each radio has interfered with the other signals; it should be noted that the same model is used for this case as the previously mentioned single radio scenario. Despite the slight increment in the incorrect estimations and the degree of incorrect, the model could still correctly estimate specific signals when trained on the complete known radio set's signal. Demonstrating transferable learning allows our methodology to scale to various known radios. It can be applied to one model per radio problem or environments where a combination of their signals represents all radios. Conclusively, we can train our model on a signal where each known radio is interfered with one another and use it to estimate an expected signal even when only one of the original radios is operational within the same environment.

The estimated signals obtained by our proposed scheme over the expected signals are reported in Figures 3-4. In Figure 3, we compare an expected signal estimated by our LSTM and the accurate signal generated via Monte Carlo methods. In contrast, Figure 4

shows the signal split into sub-signals to assist with visualizing the similarity. The signal similarity and Mean Squared Error (MSE) between the expected and estimated signals are mentioned above each sub-figure. The MSE and similarity values are calculated via the Dynamic Time Warping (DTW) algorithm by analyzing two sequential datasets outside of phase.

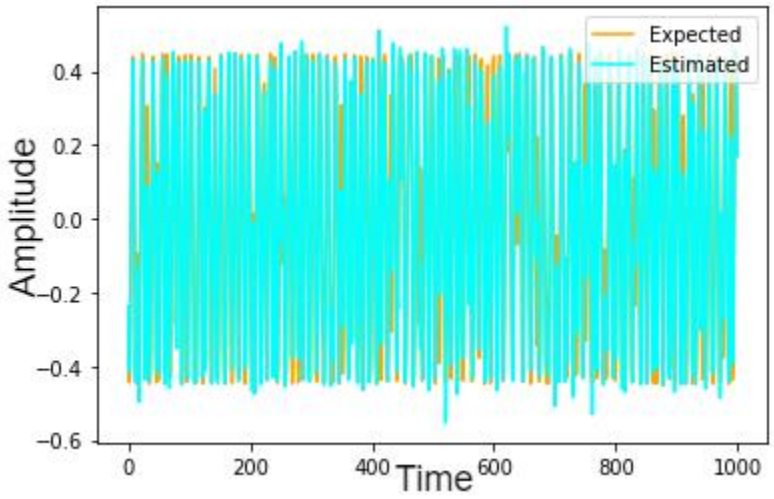

Figure 4: Signal estimation when trained over known signals.

5.4. INTERFERENCE DETECTION

Continuing the scenario mentioned earlier, our approach for RFI detection can accurately determine the locations of interference that begins and ends actively. We perform detection analysis after successfully evaluating the similarity between expected and estimated signals in the previous section. We have generated a few hundred signals, all of which perform to a similar degree as those seen in this paper. As seen below in

Figure 5, it is quite apparent where the interference occurs as our approach dramatically increases the visibility of the interference to a point where it can be mathematically determined.

Our overall approach, RaFID, for detection starts by utilizing batch-wise mean aggregation to reduce the dimensionality and overall complexity of the data into an indicative value for each batch of the received signal. It is important to note that the initial batch's mean is treated as having no noise and stored in a list. After the first batch, batch-wise mean aggregation is performed on each batch. The standard deviation of each batch-wise mean aggregate is then calculated about the average of the list of stored means. The standard deviation is saved in a history variable that keeps track of the three most recent standard deviations. If a calculated standard deviation is less than 0.2 and the average of the history variable is less than 0.3, the batch-wise mean aggregate for the current received signal batch is appended to the stored mean list; else, it a flag is raised, and the index of the current batch is stored as a starting point for interference. Once the standard deviation conditions are satisfied again, the current index is stored as an ending point of interference. This algorithm continues to run this way for as long as desired.

Interference Detection with Varying SNR. As shown in Figure 5, when supplied with an SNR of 1.75, the points where interference begins and ends can be accurately and actively detected with our approach. As we now have a functioning model, we explore and perform edge-case analysis to test the conditions where RaFID breaks, specifically regarding noise. We experimented with SNR values of 1.5, 1.75, 2, and 5 with a noise type of white Gaussian noise, and the obtained signals are depicted in Figure 6. This allows us to know how applicable our approach is when encountering

various amounts of noise. Specifically, low-noise environments where the SNR will be a large number of high-noise environments where the SNR will be quite small.

Shown in Figure 6, with an SNR of at least 1.75, our proposed approach adequately and distinctly detects interference in multi-radio environments where noise is present. Other comparable works [25] can detect interference, but these schemes are presented for scenarios in which signals have a sufficiently large SNR (low-noise environments). However, for non-white noise, our approach performs similarly for white Gaussian with an SNR of more than 4. This is different from white Gaussian noise but is expected as non-white noise could be treated as an interfering signal. To this end, it is essential to classify the noise native to known radios.

5.5. INTERFERENCE DETECTION WITH VARIOUS NON-WHITE NOISE

For the case of non-white noise, we opted to test out Brownian noise SB, Pink noise SP, and Blue noise SBL, defined in Eqs. 8-10 respectively, with SNR values of 1.25, 1.75, 2, and 4. Non-white noise calculations can be thought of as equations representative of the relationship between power level and frequency. Based upon the desired non-white noise the relationship between power level and frequency is calculated.

$$S_B = \int_t^0 \frac{dW(\tau)}{d\tau}\, d \tag{8}$$

$$S_P = \frac{W(\tau)}{f} \tag{9}$$

$$S_{BL} = \sqrt{f} * W(\tau) \tag{10}$$

W indicates White Gaussian noise with equal power at each frequency and f denotes the frequency of the signal.

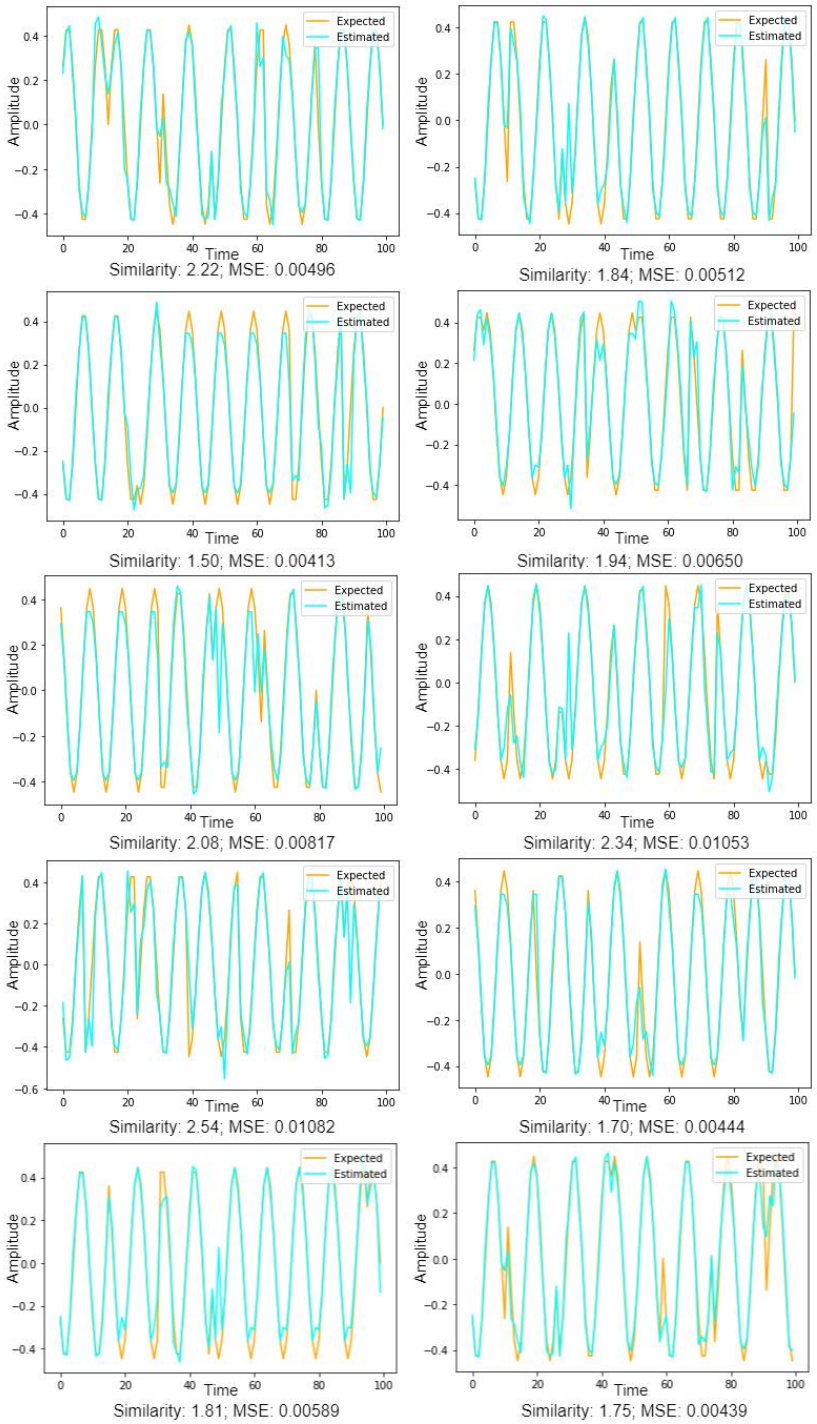

Figure 5: Batches of size 100 to illustrate the estimation ability of RaFID. The Dynamic Time Warping algorithm determines Similarity.

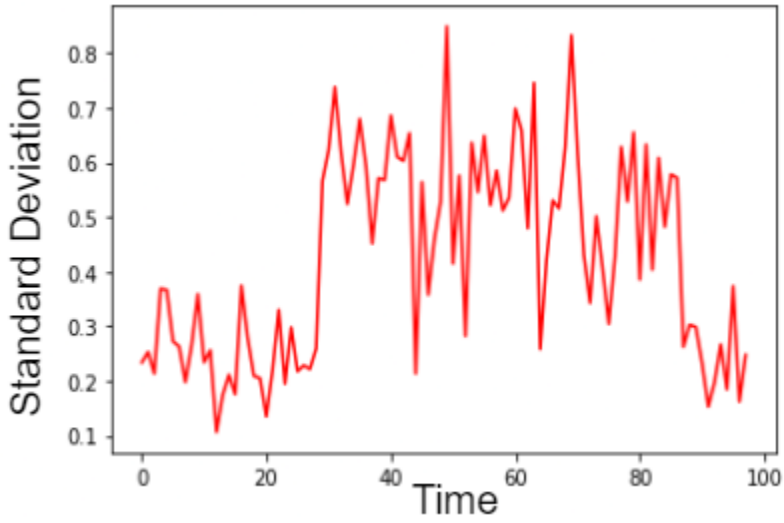

Figure 6: Our proposed scheme detects the interference occurrence between points 31 and 90 (or time steps 3100 and 9000 of our received signal).

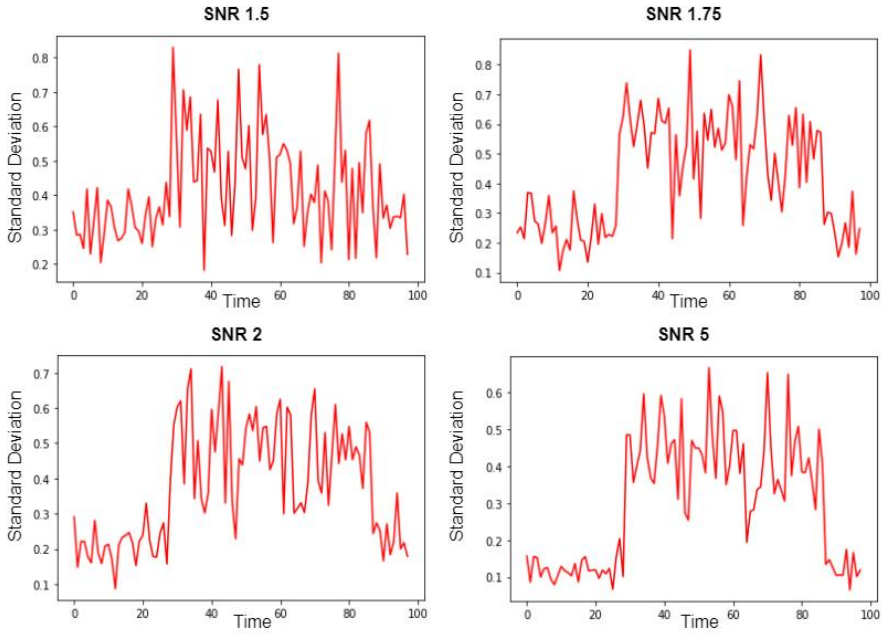

Figure 7: Detection Results with various SNR values. The higher the SNR value, the greater the difference in power level between the signal and associated noise.

The detected signals for each noise are reported in Figures 7-8. We observed that for Brownian and Pink noise, with 4 RaFID performs well with SNR values greater than or equal to 4. In contrast, RaFID detects interference with Blue noise for SNR values of more than 2. Given these results, we can say that our model handles RaFID with Blue noise better than that of Brownian or Pink noise as we are looking to determine the smallest SNR in which our approach still functions. Thus, we can infer that there are certain non-white noise types whose power level as a function of frequency are better suited for our approach and others that are not. From the included non-white noises the pattern appears to lie in low-frequency dominance (pink/brownian) vs high-frequency dominance (blue) noise.

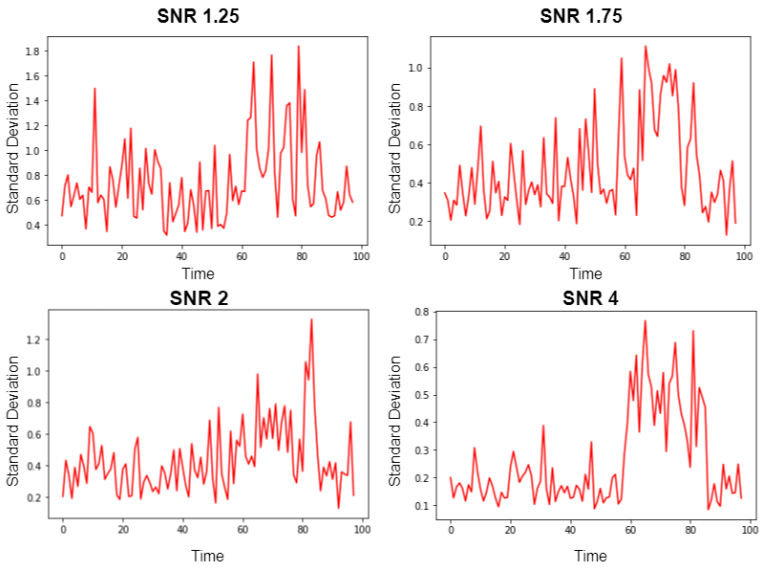

Figure 8: Detected signals using RaFID for Brownian noise.

It happens due to the high false positive rate that non-white noise introduces to varying degrees. The false positive rate results from the varying power levels relative to the frequencies within each type of noise. Across these types of non-white noise, it can be observed that our approach has a higher SNR floor with which it executes successfully. Investigation into determining the prevalent noise type within an interfered signal at the preprocessing step may prove beneficial for RFI detection in non-white noise.

Interference Detection with Varying SIR. Some interfering signals may have a lower or higher power level relative to our known signals. The SIR is considered to further expand upon potential occurrences in real-world signals. To this end, we have tested SIRs of 0.5, 1.25, 2, and 4. These tests are performed after previous experiments and comparisons have been completed and are disjointed. It is important to note that Figures 10-7 used different signal combinations.

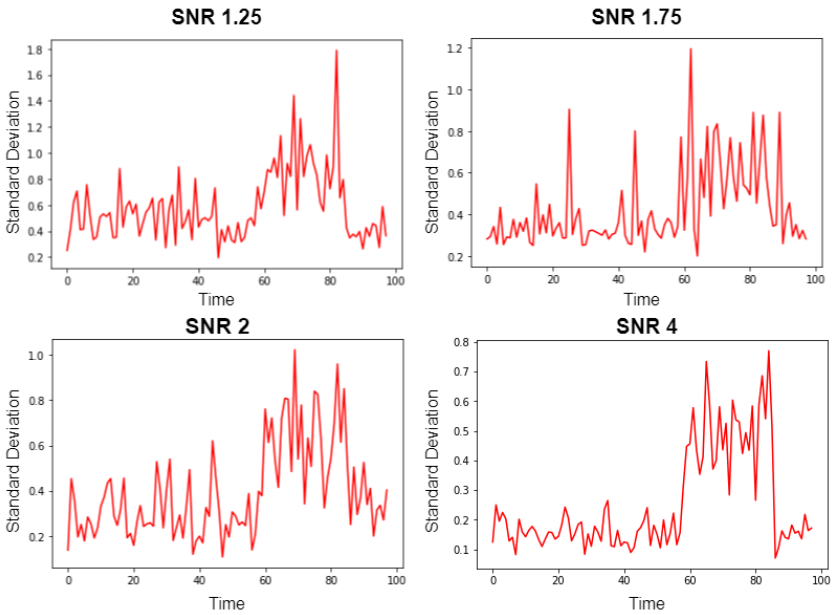

Figure 9: Detected signals using RaFID for Pink noise.

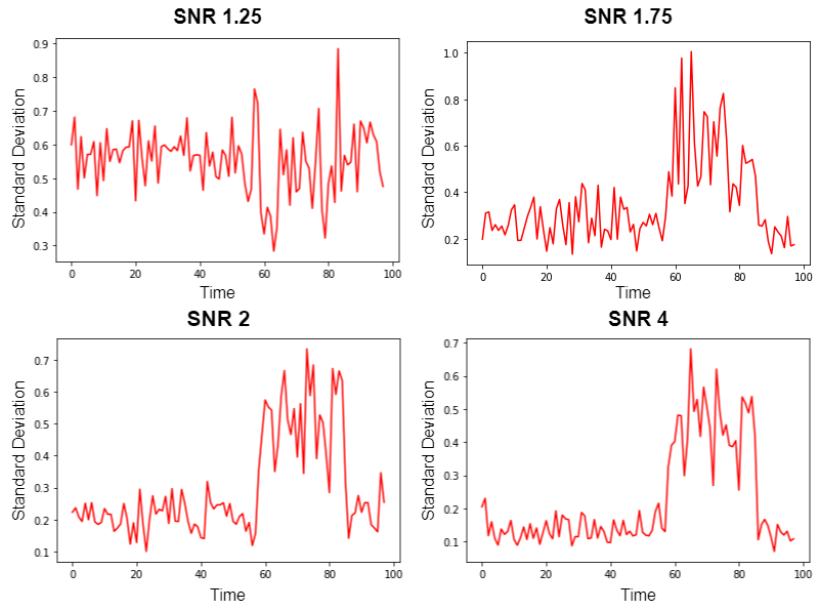

Figure 10: Detected signals using RaFID for Blue noise.

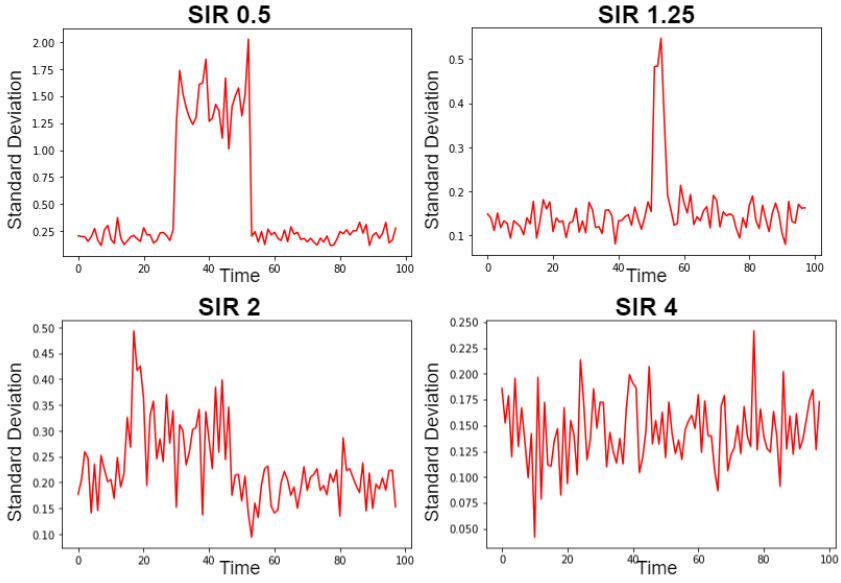

Figure 11: Detection Results with various SIR values. The higher the SIR value, the greater the power level of the known signal vs. the interfering signal, and vice versa.

As shown in Figure 10, with the interference's power level getting smaller than that of our known signal, the model begins to not detect interference as well with a SIR as low as 2. This makes logical sense, however, as the greater the SIR, the less impact the interfering signal has upon our known signal. It is also demonstrated that, inversely, with increasingly smaller SIR values, the model performs with a greater degree of success.

As is demonstrated in Figure 11, interference is reliably detected in the windows of occurrence. One important observation is how it operates across the two representations: overlapping and non-overlapping signals. In the case of non-overlapping signals, our approach can accurately indicate where the interference for each radio of this type begins and ends. However, in the case of overlapping interference, our solution finds the complete time frame where it exists until such a time that interference is not present. We observe that it is an acceptable outcome as this problem looks to solve RFI detection with a byproduct of localization, not to determine the number of interfering devices present.

6. CONCLUSION

This paper proposes a lightweight deep-learning model to detect interference upon a known radio set. The proposed methodology utilizes an LSTM to generate an expected signal and defines our desired signal as our determined distribution. The determined distribution is then compared to the received signal, and interference is detected as per Algorithm 1. Our simulations have shown that deep learning techniques can be used with statistical analysis to accurately and distinctly detect when RFI begins and ends within the time domain. This method's degree of success depends upon the relative SNR and

SIR values, as sufficiently low SNR and sufficiently high SIR begin to demonstrate a reduction in overall detection ability. This paper provides an evidence-driven basis for utilizing both deep learning and statistical analysis to offset the limitations apparent in time-domain RFI detection.

This paper also laid the groundwork for the proof of transferable learning between multi-signal and single-signal environments when trained on a signal consisting of all known signals. In our tests to adequately explore and reduce the initial issues of a single-radio model, we discovered that a model trained on a known signal set's combined signal performs comparably to that of having a model per radio. While we can't rule out the possibility that there are cases where this approach may not work, more research can indeed be performed to explore the uses and limitations of this approach. One limitation of this approach is the reliance on the assumption that the initial signal measuring has no interference from outside noise. This assumption can be accounted for by keeping a stored reference point to initialize the detection. We foresee potential future explorations, including interference classification, modulation algorithm expansion, and interference estimation.

REFERENCES

[1] Y. Ghanney and W. Ajib, “Radio frequency interference detection using deep learning,” in 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), 2020, pp. 1–5.

[2] “Radio interference,”https://www.itu.int/en/mediacentre/backgrounders /Pages/radio-interference.aspx, accessed: 2023-2-4.

[3] K. A. Hamdi, “On the statistics of signal-to-interference plus noise ratio in wireless communications,” IEEE Transactions on Communications, vol. 57, no. 11, pp. 3199–3204, 2009.

[4] H. Shahid, “Radio frequency detection, spectrum analysis, and direction finding equipment,” https://www.dhs.gov/sites/default/files/saver-msr-rf-detection_cod-508 10july2019.pdf, accessed: 2023-2-3.

[5] W. A. Baan, P. A. Fridman, and R. P. Millenaar, “Radio frequency interference mitigation at the westerbork synthesis radio telescope: Algorithms, test observations, and system implementation,” The Astronomical Journal, vol. 128, no. 2, p. 933, aug 2004. [Online]. Available: https://dx.doi.org/10.1086/422350

[6] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, “Long short term memory networks for anomaly detection in time series,” 04 2015.

[7] M. Ribeiro, A. E. Lazzaretti, and H. S. Lopes, “A study of deep convolutional auto-encoders for anomaly detection in videos,” Pattern Recognition Letters, vol. 105, pp. 13–22, 2018, Machine Learning and Applications in Artificial Intelligence. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865517302489

[8] B. Sun and L. Ma, “An overview of outliers and detection methods in general for time series from iot devices,” in The 10th International Conference on Computer Engineering and Networks, Q. Liu, X. Liu, T. Shen, and X. Qiu, Eds. Singapore: Springer Singapore, 2021, pp. 1180–1186.

[9] A. J. Schoenwald, S.-J. Kim, and P. N. Mohammed, “Radio frequency interference detection for passive remote sensing using eigenvalue analysis,” in 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2017, pp. 1270–1273.

[10] F. Laricchia, “Forecast number of mobile devices worldwide from 2020 to 2025 (in billions)*,” https://www.statista.com/statistics/245501/multiple-mobile-device-ownership-worldwide/, accessed: 2023-01-28.

[11] P. Taylor, “Average number devices and connections per person world-wide in 2018 and 2023,” https://www.statista.com/statistics/1190270/number-of-devices-and-connections-per-person-worldwide/, accessed: 2023-01-20.

[12] R. Chalapathy and S. Chawla, “Deep learning for anomaly detection: A survey,” 2019. [Online]. Available: https://arxiv.org/abs/1901.03407

[13] Z. Li, C. Yu, J. Xiao, M. Long, and C. Cui, “Detection of radio frequency interference using an improved generative adversarial network,” Astronomy and Computing, vol. 36, p. 100482, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2213133721000366

[14] J. Kerrigan, P. L. Plante, et al, “Optimizing sparse RFI prediction using deep learning,” Monthly Notices of the Royal Astronomical Society, vol. 488, no. 2, pp. 2605–2615, jul 2019. [Online]. Available: https://doi.org/10.1093%2Fmnras%2Fstz1865

[15] Y.-c. Park, J.-G. Jang, and U. Kang, “Fast partial fourier transform,” 2020. [Online]. Available: https://arxiv.org/abs/2008.12559

[16] S. Gopali, F. Abri, S. Siami Namini, and A. Siami Namin, “A comparative study of detecting anomalies in time series data using lstm and tcn models,” arXiv:2112.09293, 12 2021.

[17] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, pp. 1735–80, 12 1997.

[18] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, “Temporal convolutional networks: A unified approach to action segmentation,” in Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14. Springer, 2016, pp. 47–54.

[19] O. Mosiane, N. Oozeer, and B. A. Bassett, “Radio frequency interference detection using machine learning,” in 2016 IEEE Radio and Antenna Days of the Indian Ocean (RADIO), 2016, pp. 1–2.

[20] D. Nelson, “What is an autoencoder?” https://www.unite.ai/what-is-an-autoencoder/, accessed: 2022-12-13.

[21] H. Weytjens and J. D. Weerdt, “Process outcome prediction: CNN vs. LSTM (with attention),” in Business Process Management Workshops. Springer International Publishing, 2020, pp. 321–333. [Online]. Available: https://doi.org/10.1007%2F978-3-030-66498-5_24

[22] C. Ruf, S. Gross, and S. Misra, “Rfi detection and mitigation for microwave radiometry with an agile digital detector,” IEEE Transactions on Geoscience and Remote Sensing, vol. 44, no. 3, pp. 694–706, 2006.

[23] A. J. Schoenwald, A. Gholian, D. C. Bradley, M. Wong, P. N. Mohammed, and J. R. Piepmeier, “Rfi detection and mitigation using independent component analysis as a pre-processor,” in 2016 Radio Frequency Interference (RFI), 2016, pp. 100–104.

[24] G. Cucho-Padin, Y. Wang, E. Li, L. Waldrop, Z. Tian, F. Kamalabadi, and P. Perillat, “Radio frequency interference detection and mitigation using compressive statistical sensing,” Radio Science, vol. 54, no. 11, pp. 986–1001, 2019. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019RS006902

[25] Q. Wu, Z. Sun, and X. Zhou, “Interference detection and recognition based on signal reconstruction using recurrent neural network,” in 2019 IEEE Globecom Workshops (GC Wkshps), 2019, pp. 1–6.

SECTION

3. UNPUBLISHED WORK

The potential of expanding upon this research is to determine signal trust based on noise levels within a known signal. This research was recently started and will not be completed by the time of this publication.

The use of this work in more real-world environments and stress tested within those environments. We started exploring real-world data but that research will not be published by the time of this publication.

4. CONCLUSIONS AND RECOMMENDATIONS

4.1. CONCLUSIONS

This thesis presented a paper introducing time domain learning as an efficient alternative to frequency domain RFI detection utilizing an LSTM signal simulator and a statistical analysis discriminator. This work exhibits that in a dense RF environment, the overall complexity and memory requirements of time domain detection are superior to that of frequency domain detection and can be effectively used with live signals being actively received. Our work also demonstrated the differences between different types of noise and how they affect the detection of interference in an environment.

We also established the validity of transferable learning in time domain signal simulation. The ability to train a single standard LSTM on a set of known signals

interfered with each other and still accurately simulate any subset of those signals will be a boon to future research within RFI detection. This demonstration holds great promise and warrants future exploration.

4.2. RECOMMENDATIONS

We recommend further exploring this approach and its interactions with varying signal modulation schemes, noise types, and RF environments. For instance: QPSK, QAM, etc. While we did begin investigating these other modulation schemes we were unable to complete explorations and comparisons.

We also recommend exploring the use of this approach with more real-world environments as we did find that noise in real-world data does require some additional tuning for comparable results. Overall, from what we were able to find up to this point results were similar despite the large increase in ambient noise.

A final recommendation is to adequately consider and research expanding the use-case(s) of this research to include areas such as trust determination and signal extraction. Trust determination, as described in Unpublished Work, would utilize our approach to determine a signal's trustworthiness based on SNR/SIR levels as compared to our generated signal. The ability to isolate noise is similarly implicit in the design of our approach as signal interference is additive, but the difficulty comes with determining how many signals may be interfered to create the residual signal.

BIBLIOGRAPHY

[1] Y. Ghanney and W. Ajib, "Radio Frequency Interference Detection using Deep Learning," 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), Antwerp, Belgium, 2020, pp. 1-5, doi: 10.1109/VTC2020-Spring48590.2020.9129612.

[2] "Radio interference," https://www.itu.int/en/mediacentre/backgrounders/Pages/radio-interference.aspx, accessed: 2023-2-4.

[3] K. A. Hamdi, "On the statistics of signal-to-interference plus noise ratio in wireless communications," IEEE Transactions on Communications, vol. 57, no. 11, pp. 3199–3204, 2009.

[4] H. Shahid, "Radio frequency detection, spectrum analysis, and direction finding equipment," https://www.dhs.gov/sites/default/files/saver-msr-rf-detection_cod-508 10july2019.pdf, accessed: 2023-2-3.

[5] W. A. Baan, P. A. Fridman, and R. P. Millenaar, "Radio frequency interference mitigation at the westerbork synthesis radio telescope: Algorithms, test observations, and system implementation," The Astronomical Journal, vol. 128, no. 2, p. 933, aug 2004. [Online]. Available: https://dx.doi.org/10.1086/422350

[6] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," 04 2015.

[7] M. Ribeiro, A. E. Lazzaretti, and H. S. Lopes, "A study of deep convolutional auto-encoders for anomaly detection in videos," Pattern Recognition Letters, vol. 105, pp. 13–22, 2018, Machine Learning and Applications in Artificial Intelligence. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865517302489

[8] B. Sun and L. Ma, "An overview of outliers and detection methods in general for time from iot devices," in The 10th International Conference on Computer Engineering and Networks, Q. Liu, X. Liu, T. Shen, and X. Qiu, Eds. Singapore: Springer Singapore, 2021, pp. 1180–1186.

[9] A. J. Schoenwald, S.-J. Kim, and P. N. Mohammed, "Radio frequency interference detection for passive remote sensing using eigenvalue analysis," in 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2017, pp. 1270–1273.

[10] F. Laricchia, “Forecast number of mobile devices worldwide from 2020 to 2025 (in billions)*,” https://www.statista.com/statistics/245501/multiple-mobile-device-ownership-worldwide/, accessed: 2023-01-28.

[11] P. Taylor, “Average number devices and connections per person world-wide in 2018 and 2023,” https://www.statista.com/statistics/1190270/number-of-devices-and-connections-per-person-worldwide/, accessed: 2023-01-20.

[12] R. Chalapathy and S. Chawla, “Deep learning for anomaly detection: A survey,” 2019. [Online]. Available: https://arxiv.org/abs/1901.03407

[13] Z. Li, C. Yu, J. Xiao, M. Long, and C. Cui, “Detection of radio frequency interference using an improved generative adversarial network,” Astronomy and Computing, vol. 36, p. 100482, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2213133721000366

[14] J. Kerrigan, P. L. Plante, et al, “Optimizing sparse RFI prediction using deep learning,” Monthly Notices of the Royal Astronomical Society, vol. 488, no. 2, pp. 2605–2615, jul 2019. [Online]. Available: https://doi.org/10.1093%2Fmnras%2Fstz1865

[15] Y.-c. Park, J.-G. Jang, and U. Kang, “Fast partial fourier transform,” 2020. [Online]. Available: https://arxiv.org/abs/2008.12559

[16] S. Gopali, F. Abri, S. Siami Namini, and A. Siami Namin, “A comparative study of detecting anomalies in time series data using lstm and tcn models,” arXiv:2112.09293, 12 2021.

[17] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, pp. 1735–80, 12 1997.

[18] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, “Temporal convolutional networks: A unified approach to action segmentation,” in Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14. Springer, 2016, pp. 47–54.

[19] O. Mosiane, N. Oozeer, and B. A. Bassett, “Radio frequency interference detection using machine learning,” in 2016 IEEE Radio and Antenna Days of the Indian Ocean (RADIO), 2016, pp. 1–2.

[20] D. Nelson, “What is an autoencoder?” https://www.unite.ai/what-is-an-autoencoder/, accessed: 2022-12-13.

[21] H. Weytjens and J. D. Weerdt, “Process outcome prediction: CNN vs. LSTM (with attention),” in Business Process Management Workshops. Springer International Publishing, 2020, pp. 321–333. [Online]. Available: https://doi.org/10.1007%2F978-3-030-66498-5_24

[22] C. Ruf, S. Gross, and S. Misra, “Rfi detection and mitigation for microwave radiometry with an agile digital detector,” IEEE Transactions on Geoscience and Remote Sensing, vol. 44, no. 3, pp. 694–706, 2006.

[23] A. J. Schoenwald, A. Gholian, D. C. Bradley, M. Wong, P. N. Mohammed, and J. R. Piepmeier, “Rfi detection and mitigation using independent component analysis as a pre-processor,” in 2016 Radio Frequency Interference (RFI), 2016, pp. 100–104.

[24] G. Cucho-Padin, Y. Wang, E. Li, L. Waldrop, Z. Tian, F. Kamalabadi, and P. Perillat, “Radio frequency interference detection and mitigation using compressive statistical sensing,” Radio Science, vol. 54, no. 11, pp. 986–1001, 2019. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019RS006902

[25] Q. Wu, Z. Sun, and X. Zhou, “Interference detection and recognition based on signal reconstruction using recurrent neural network,” in 2019 IEEE Globecom Workshops (GC Wkshps), 2019, pp. 1–6.

VITA

Luke Andrew Smith was born in Houston, Texas, and raised in Marceline, Missouri. He graduated from Marceline R-V High School in May 2014 where he would then attend Missouri University of Science and Technology from August 2014-May 2019 for his Bachelor of Science in Computer Science. Luke earned a Master of Science in Computer Science from Missouri University of Science and Technology in July 2023.