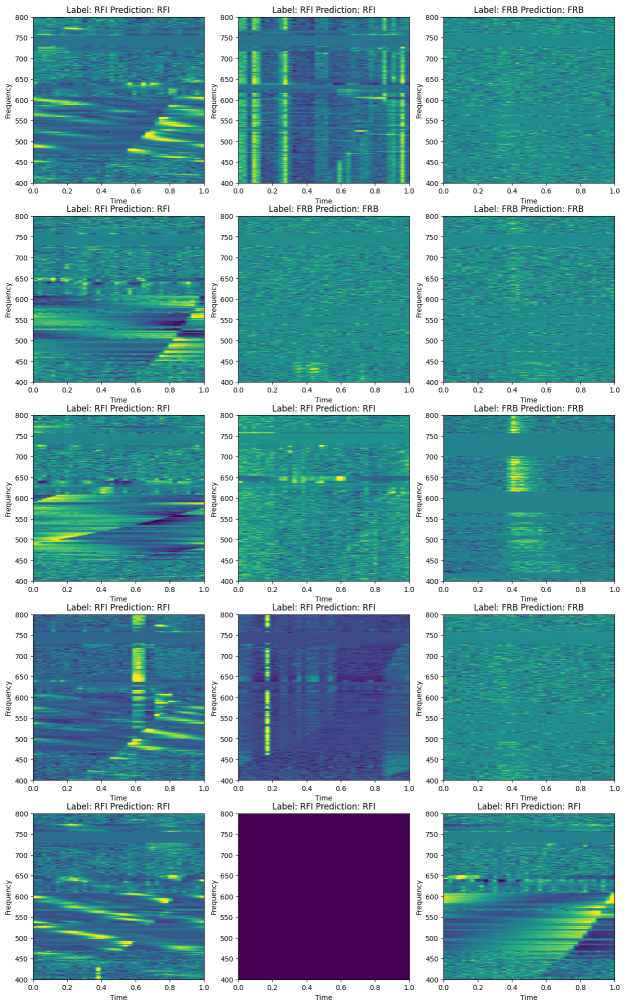
Phys 310: Classifying Fast Radio Bursts from Radio Frequency Interference for the CHIME Telescope

Nicholas Bratvold

April 20, 2024



Contents

1 Introduction **4**

2 Background **4**

2.1 Fast Radio Bursts 4

2.1.1 Dispersion Measure 4

2.2 Convolutional Layers 5

2.2.1 Layers 5

2.2.2 Activation Functions 5

2.2.3 Optimizers and Loss Functions 6

3 Methods **6**

3.1 Data 6

3.1.1 FRB Data 6

3.1.2 RFI Data 8

3.1.3 Preprocessing the Data 9

3.2 Convolutional Network 11

3.2.1 Model Architecture 12

4 Results **12**

4.1 Dataset 12

4.2 Model 1 13

4.3 Model 2 13

5 Discussion and Summary **15**

6 Appendix **15**

List of Figures

1 A comparison of data in the two datasets. The first plot is from the FRB catalogue and the second is from raw interference data. 6
2 Two plots of an example FRB. The first plot is the waterfall data and model waterfall data. The second plot contains the calibrated waterfall data. 7
3 RFI data. 8
4 The 'unprocessed' data for a FRB signal. From left to right: waterfall, model waterfall, calibrated waterfall, RFI example. 9
5 A series of transformations to make the raw RFI data similar to the FRB data. 10
6 Burst Examples. 11
7 Interference Examples. 11
8 Model 1 training curves. There is evidence of over fitting in the loss plot. . . 13
9 Incorrect Prediction 14
10 Model 2 training curves. 14
11 Test predictions with no dedispersion applied. 16
12 Test predictions with dedispersion applied 17

1 Introduction

The Canadian Hydrogen Intensity Mapping Experiment (CHIME) is finding hundreds of Fast Radio Bursts (FRBs) using radio data between 400 and 800 MHz. These bursts have high "Dispersion Measures" (DMs) corresponding to the density of electrons between us and the FRB. FRBs with smaller DMs come from galactic sources, such as pulsars emitting bright single pulses. The DMs can also be large, indicating FRBs of extra-galactic distances away. Although CHIME is located in a radio free zone, there are still many human-generated radio-frequency interference (RFI) that are identified as FRBs. These RFI signals can be distinguished from a true FRB by a human, but this takes a lot of time. By implementing a convolutional neural net (CNN) classifier the true signals can be classified accurately and efficiently.

2 Background

2.1 Fast Radio Bursts

Fast Radio Bursts are bright, millisecond flashes with a broadband signal. The frequencies of each burst are delayed by different amounts of time depending on the wavelength. This delay is referred to as a dispersion measure. The detected bursts have DMs $\leq$ 100pc cm$^{-}3$, much larger than expected for a source inside the Milky Way galaxy. Due to this, the bursts are conjectured to be a extragalatic origin. Studying the FRBs will allow astronomer's to study and map the universe.

2.1.1 Dispersion Measure

The amount of dispersion can be measured by the cold-plasm dispersion law [4]:

$$DM = \frac{2\pi m_e c \Delta t}{e^2(f_{low}^{-2} - f_{high}^{-2})} \tag{1}$$

where m_e and e are the mass and charge of an electron. c is the speed of light. Δt is the difference in time between the low and high frequencies. [7]

We can simplify this equation for computational purposes by writing it as:

$$DM = = \frac{a^{-1}\Delta t}{f_{low}^{-2} - f_{high}^{-2}} \tag{2}$$

where [3]

$$a = \frac{e^2}{2\pi m_e c} = 4.15 GHz^2 cm^3 pc^{-1} ms \tag{3}$$

The CHIME telescope L1 stage predicts an estimate DM for a detected FRB. The estimated DM can be used to dedisperse the FRB to align the frequencies together.

2.2 Convolutional Layers

Convolutional neural networks(CNN) are a type of machine learning architecture used to adaptively learn spatial features from data. This model is applicable for use on classifying FRBs because of the spacial properties. The bursts are indicated by a thin high intensity strip. This spatial information can be extracted and used to classify FRBs from RFIs.

The CNN used in this project is made up of the following components:

2.2.1 Layers

Layers are used to transform the input data into the required output shape. While doing this, the neurons in the layers adjust their weights to accuratly predict the output from a given input.

- Convolutional Layers: The layers apply convolutional filters to extract local spatial features such as edges. Different filters can be applied to extract different features.
- Pooling layers: Pooling layers reduce the spatial dimensions of the data space. This allows for important features to be extracted into a smaller space making the network more computationally efficient.
- Dropout layers: During training a set ratio of random neurons are set to 0. This helps to regularize the model to reduce overfitting. This also effectively adds noise to the training data allowing for more robust features to be extracted.
- Dense layers: These are fully connected layers where each neuron is connected to every neuron in the previous layer. This allows for complex and relationships in the data to be extracted.
- Flatten layer: This is a transition layer that reshapes the tensor output from previous layers into a one dimensional vector. This is necessary for the dense layers as they require a one dimensional input.

2.2.2 Activation Functions

Neurons are activated by various activation functions to introduce non-linearity to the network.

- Rectified Linear Unit (ReLU) $ReLU(x) = max(0, x)$. This introduces sparsity into the network and is computationally efficient. It is the most popular activation function. Some variants allow a small negative weight to leak through to stop neurons getting stuck with 0 weight.
- Sigmoid $\sigma(x) = \frac{1}{1+e^{-x}}$. This activation is usually used for binary classification. It causes the output to remain between 0 and 1.

2.2.3 Optimizers and Loss Functions

Optimizers are used to train neural networks. The Adam optimizer is a popular algorithm that adaptively adjusts learning rates during training allowing for faster convergence and better generalization to unseen data.

The optimizer is used to improve the score from the loss function. In binary classification problems the Binary Cross-Entropy Loss is often used. It is defined as:

$$L = -(y \log(p) + (1 - y) \log(1 - p)) \tag{4}$$

Where p is the predicted class and y is the true class.

All CNN information was obtained from Viviana Acquaviva's text book [1].

3 Methods

3.1 Data

For this project two data sets were given. A catalogue of confirmed FRB signals which are highly processed, provided by CANFAR [2], and raw interference data provided by Ingrid Stairs. Figure 1 showcases the differences between the data. Applying any machine learning to the differing data sets is a futile thing.

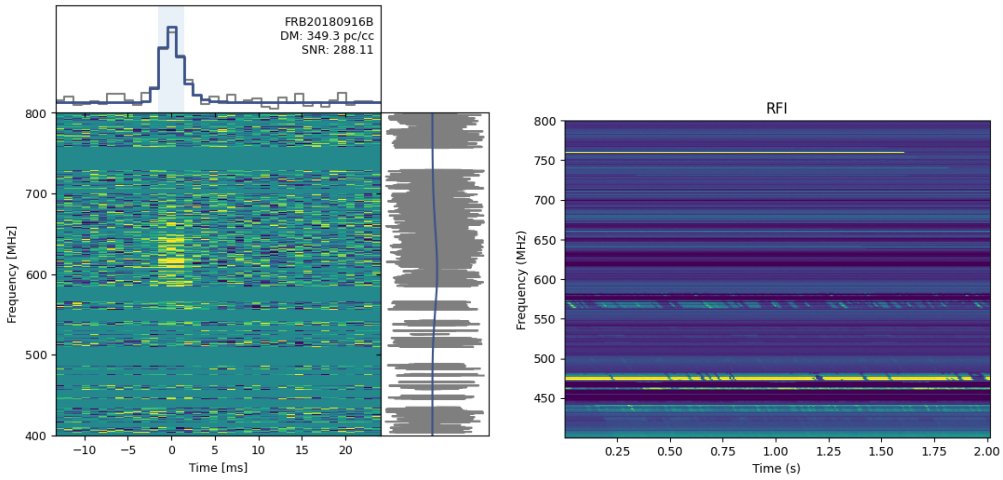


Figure 1: A comparison of data in the two datasets. The first plot is from the FRB catalogue and the second is from raw interference data.

3.1.1 FRB Data

The FRB catalogue contains a ton of details about each FRB. Some relevant info is in Table 1 In brief, it has data with 16384 frequency channels ranging from 400 to 800 MHz and anywhere from 19 to over 200 time points at various time steps. The data has already been

Index	Name	Shape
0	calibrated_wfall	(16384, 192)
1	extent	(4,)
2	model_spec	(16384,)
3	model_ts	(38,)
4	model_wfall	(16384, 38)
5	plot_freq	(16384,)
6	plot_time	(38,)
7	spec	(16384,)
8	ts	(38,)
9	wfall	(16384, 38)
10	dm	(1,)

Table 1: Description of Fast Radio Burst Data

processed in some known and some unknown ways. From reading [7], [2] and [5], it is assumed that the data has been dedispersed, normalized, and has had certain frequency bands removed. It is the specifics and order in which these are done that is unknown.

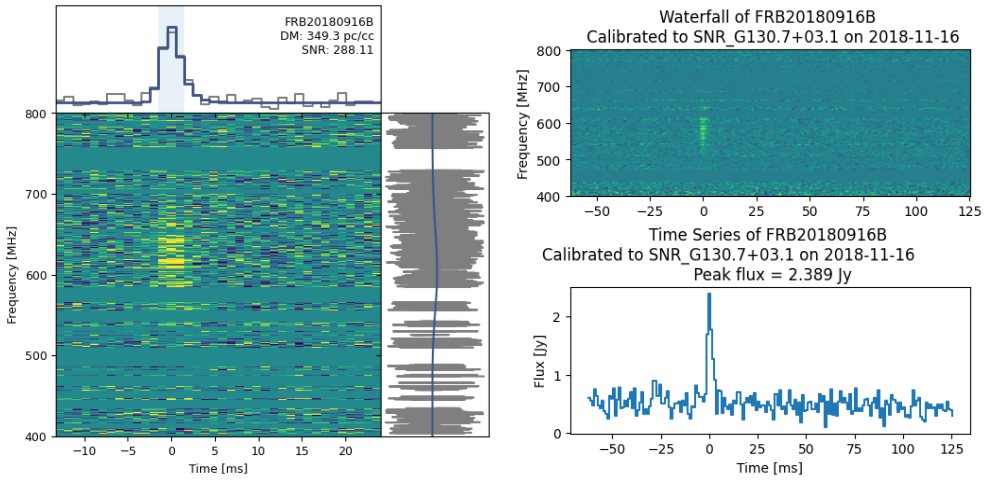


Figure 2: Two plots of an example FRB. The first plot is the waterfall data and model waterfall data. The second plot contains the calibrated waterfall data.

Figure 2 showcases what the given CFOD [6] code is capable of after a few bug fixes. The FRB signal is aligned and prominent with a model signal, frequency spectrum, and extraction of the FRB peak. Notice how the calibrated waterfall plot and the standard waterfall plot have the frequency data flipped. I assume that the standard waterfall plot is correct since it is what is posted in the papers.

3.1.2 RFI Data

The RFI catalogue also contains a ton of details about each RFI. However, the details are very different from the FRB data. The details are displayed in Table 2

Index	Name	Value/Shape
0	intensity	(16384, 256)
1	fbottom	400.195312
2	df	0.024414
3	tstart	0.0
4	dt	0.000983
5	dm	20.623087
6	nchan	16384
7	nsamp	256
8	ftop	800.195312
9	bandwidth	400.0
10	frequencies	(16384,)
11	center_frequencies	(16384,)
12	tend	3.019899
13	times	(3072,)
14	center_times	(3072,)
15	weights	(16384, 256)
16	beam_no	()

Table 2: Description of RFI Data

In brief, the data has 16384 frequency channels ranging from 400 to 800 MHz with 256 time points at various time steps. This data is unprocessed but has an estimated DM from the CHIME telescope L1 stage that can be used. This data is plotted using the script given by Ingrid Stairs and is shown in Figure 3. Notice how different frequency channels have different intensities and a few channels have a very bright signal.

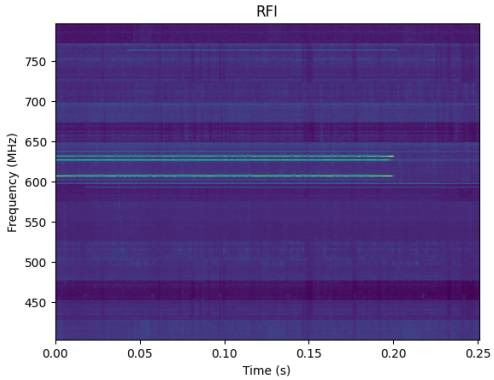


Figure 3: RFI data.

3.1.3 Preprocessing the Data

The goal is to make the raw RFI data and the processed FRB data as similar as possible so a meaningful comparison can be made. Unfortunately, there is no raw FRB data available.

Comparing Raw Data First, the FRB data had code that does some processing. It removes outlier frequency bands and bins the frequency from 16384 channel to 1024 channels. The RFI removal code and frequency binning was taken out and the waterfall, model waterfall, calibrated waterfall, and an RFI example was plotted. The resulting image is shown in Figure 4.

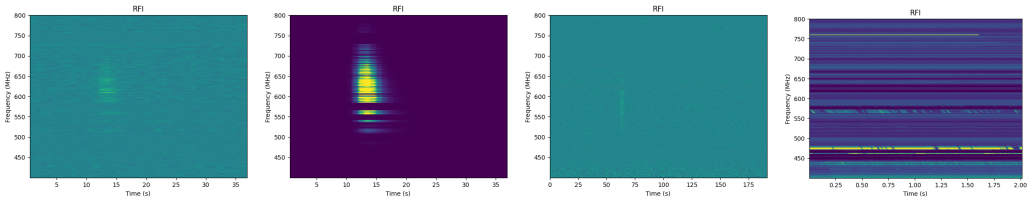


Figure 4: The 'unprocessed' data for a FRB signal. From left to right: waterfall, model waterfall, calibrated waterfall, RFI example.

It is evident, that there is additional processing in the FRB data that can not be undone. The beam has been dedispersed, normalized, the LTE band, 729 to 756 MHz, is removed[5] and other bands are also gone. Since the RFI data does not have a complimentary model waterfall, the model waterfall and calibrated waterfall will not be included in the ultimate training dataset.

Frequency Binning Issue An issue encountered is the way the RFI code and the FRB code bins the frequency data. If you compare Figure 3 and the right image in Figure 4 you will notice they are very different. These come from the exact same data.

The FRB code bins frequencies by

```
data = np.reshape(16384 // 16, 16 , time)
data = np.mean(data, axis=1)
```

The RFI code bins frequencies by

```
data = np.reshape(16384 // 1024, 1024 , time)
data = np.mean(data, axis=0)
```

Using the FRB code on the RFI data does not severely alter the data. I believe the RFI code provided has an error due to numpy's indexing and reshaping on a different axis.

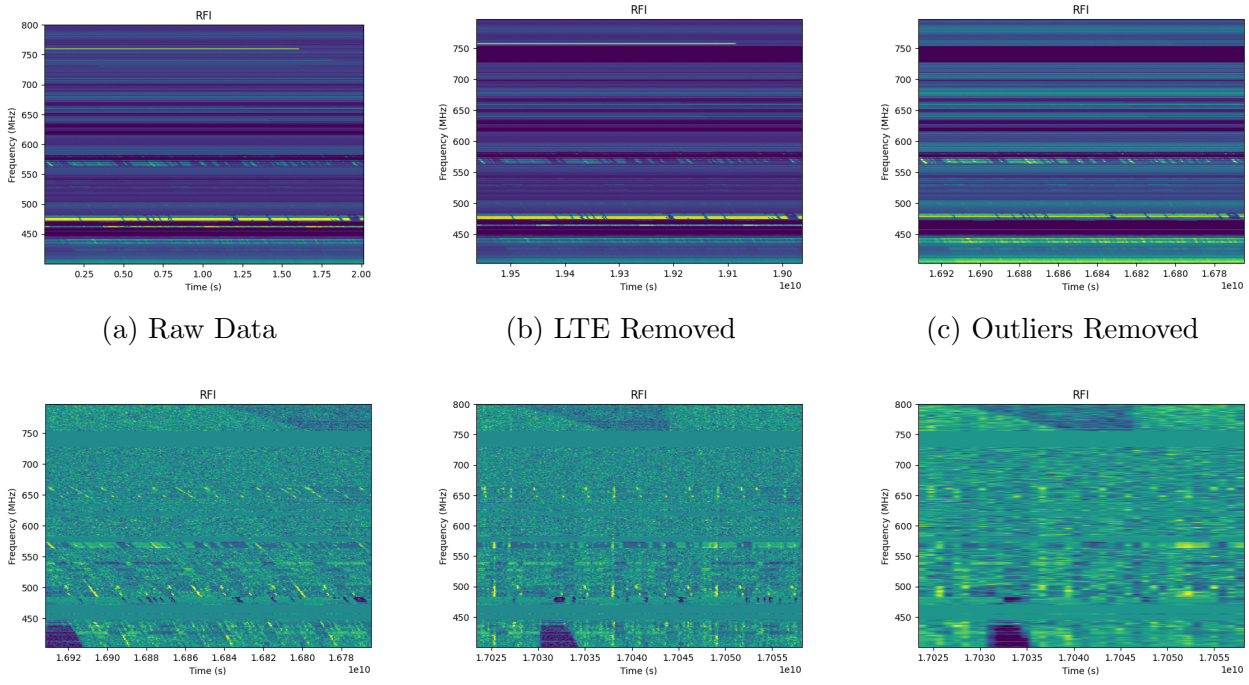
Processing Routine To make the RFI data similar to the FRB data the following needs to be done:

- Remove the LTE band

  - This band is removed in FRB data.

- Remove outlier frequency bands
  - FRBs are broadband. Removing frequency channels more than 3 standard distributions from the mean variance across all frequencies allows for only broadband signals to remain.
- Normalize the data across each frequency channel
  - Keeps each frequency band consistent with each other.
- Dedisperse the RFI with predicted DM
  - Use Equation 2 and custom script to align interference data.
- Down sample frequency to 1024 channels
  - Reduce data size.
- Down sample time to 38 columns
  - 38 is most common time samples in FRB data. The amount of information needs to be the same.

Figure 5 shows the progress of these steps and their effects on the data.



(a) Raw Data (b) LTE Removed (c) Outliers Removed

(d) Normalized (e) Dedispersed (f) Down Sampled

Figure 5: A series of transformations to make the raw RFI data similar to the FRB data.

With the processing done, all data can be processed into 1024x38 arrays, labeled as RFI or FRB, and split into training and validation data sets.

Here in Figure 6 and Figure 7 are some samples of FBRs and RFIs after being processed. There is still a lot of streaking in the interference data. Perhaps this what needs to be detected.

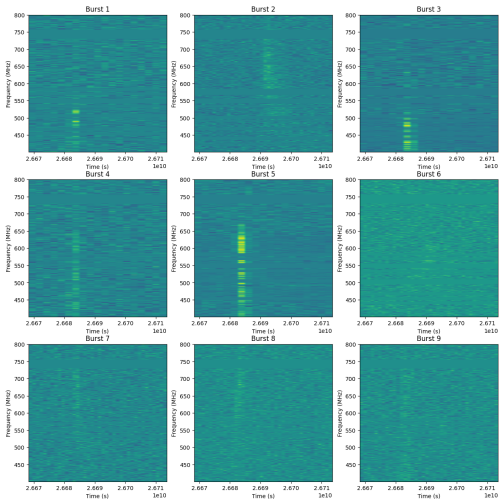


Figure 6: Burst Examples.

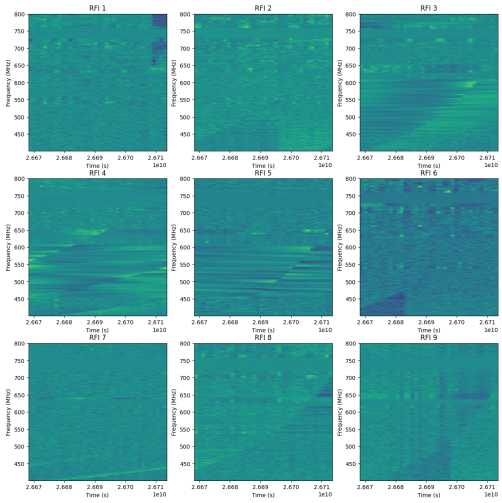


Figure 7: Interference Examples.

3.2 Convolutional Network

With the data processing out of the way we can create a CNN to classify the data as FRBs or RFIs.

A model was constructed and was scored on accuracy, recall, precision, and f1 metrics given the same dataset.

3.2.1 Model Architecture

The model was designed based on a previous project that classified galaxies from images. The layers had to be converted work with 2D data rather than 3D image data. The architecture is as follows:

Layer (type)	**Output Shape**	**Param #**
conv1d (Conv1D)	(None, 1022, 32)	3,680
max_pooling1d (MaxPooling1D)	(None, 511, 32)	0
conv1d_1 (Conv1D)	(None, 509, 64)	6,208
max_pooling1d_1 (MaxPooling1D)	(None, 254, 64)	0
conv1d_2 (Conv1D)	(None, 252, 128)	24,704
max_pooling1d_2 (MaxPooling1D)	(None, 126, 128)	0
conv1d_3 (Conv1D)	(None, 124, 128)	49,280
flatten (Flatten)	(None, 15872)	0
dense (Dense)	(None, 64)	1,015,872
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 1)	65

Table 3: Model Architecture

There are 4 repetitions of convolutional and pooling layers to extract spacial features and down sample the data. Next, the tensor is flattened and sent through two fully connected layers to extract complex relationships in the data. Finally the model outputs a probability of the class through a sigmoid activation function. The models were trained using an Adam optimizer for 100 epochs with a batch size of 32.

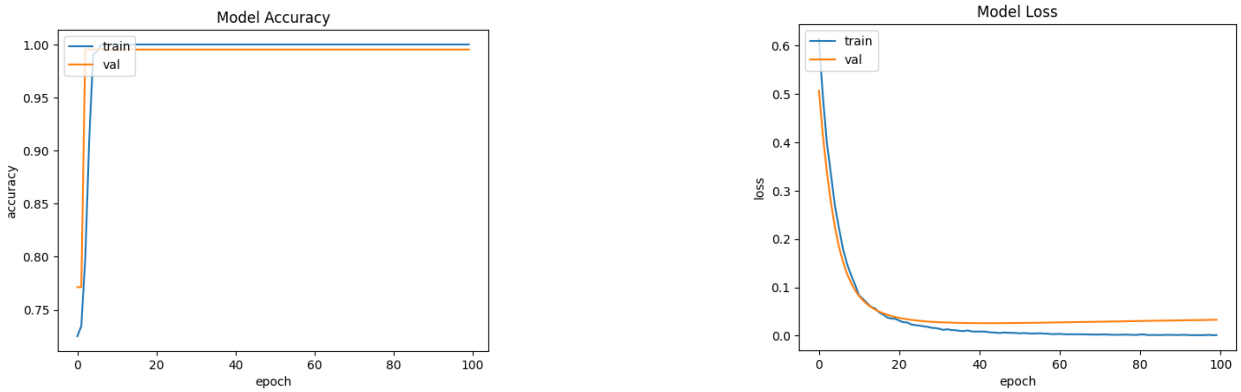
4 Results

4.1 Dataset

The final dataset used has 524 samples. 133 of these are RFIs and 390 are FRBs. There were some issues downloading the RFI files into zip folders. Only 20Gb could be downloaded at once and the data was often corrupted. As a result, 133 RFIs were able to be extracted. Downloading the files individually would work but would take a lot of time. The FRBs were scraped off of the web using a custom 'scraper.py' script. A high 60:40 training validation split was chosen so the model was sure to generalize well to data it had not seen.

4.2 Model 1

The model trained extremely well. Figure 8 showcase the accuracy and loss curves. Table 4 and Table 5 demonstrate the model scoring metrics. It is almost perfect with a single incorrect classification as shown in Figure 9. There are some signs of over fitting in the loss plot as the validation loss starts to diverge after 20 epochs.



(a) Model Accuracy

(b) Model Loss

Figure 8: Model 1 training curves. There is evidence of over fitting in the loss plot.

Class	Precision	Recall	F1-score	Support
0	0.99	1.00	1.00	133
1	1.00	1.00	1.00	390
Accuracy				1.00
Macro avg	1.00	1.00	1.00	523
Weighted avg	1.00	1.00	1.00	523

Table 4: Classification Report 1

		Predicted	
		0	1
Actual	0	133	0
	1	1	389

Table 5: Confusion Matrix 1

4.3 Model 2

A small improvement to the model was made by training it again with a smaller learning rate. This now perfectly classifies all data. Figure 10 showcase the accuracy and loss curves. Table 6 and Table 7 demonstrate the model scoring metrics.

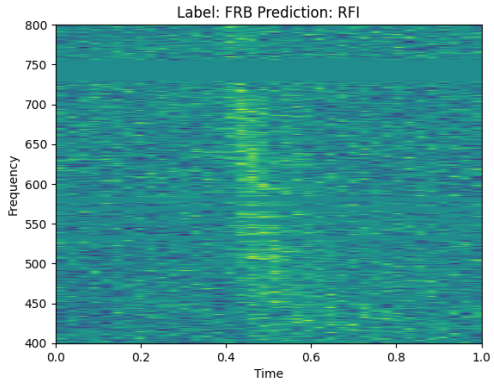
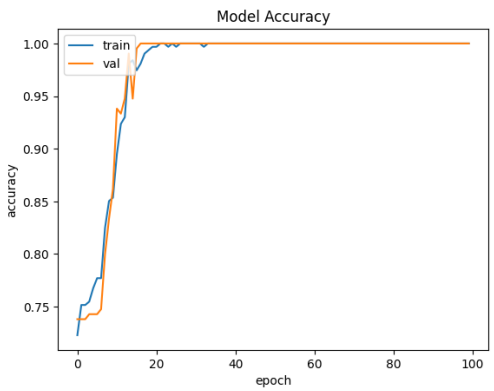
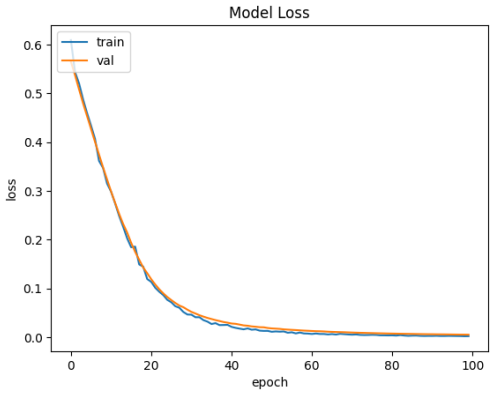


Figure 9: Incorrect Prediction



(a) Model Accuracy



(b) Model Loss

Figure 10: Model 2 training curves.

Class	**Precision**	**Recall**	**F1-score**	**Support**
0	1.00	1.00	1.00	133
1	1.00	1.00	1.00	391
Accuracy				1.00
Macro avg	1.00	1.00	1.00	524
Weighted avg	1.00	1.00	1.00	524

Table 6: Classification Report 2

		Predicted	
		0	1
Actual	0	133	0
	1	0	391

Table 7: Confusion Matrix 2

To further validate the model. A random selection of FRBs and RFIs were downloaded off of the data catalogues. There was some concerns that the dedispersion algorithm was not performing, indicated by the sweeping artifacts in Figure 7. To validate that it was working, the model was evaluated on new data that had the dedispersion applied and to data that didn't have the dedispersion applied. Both the dispersed data in Figure 11 and the dedispersed data in Figure 12 were predicted perfectly by the model.

5 Discussion and Summary

In summary, the CNN model was able to achieve perfect scores in accuracy, precision, recall, and f1 score. The CNN did this through four convolutional and pooling layers into two fully connected layers to produce a binary output.

Honestly, the perfect scoring of the model is concerning. I think the dataset used was unfortunately too different from each other, even after the processing was applied. In the future, the data for the FRBs and RFIs should be collected at the same stage in the CHIME telescope. This data then can be put through the same processing and a better classification could be made. Yadav, the grad student, who also made a model for this classification problem, had access to the CHIME telescope's raw data for their model [7]. They achieved similar results and had Accuracy 99.2%, Precision 99.1%, Recall 99.6%, and F1-Score 99.3% with their model. They also augmented the data by shifting the data left or right, flipping the time axis, and adding noise. Doing these augments allows for more robust features to be extracted from the data. This is a good consideration in the future.

This was an awesome project and course. Thank you. It is also my absolute final project/final ever at UBC and was a very nice one to end it on!

6 Appendix

The following files are included in the submission:

- frb_project.pdf, frb_project.ipynb My log book
- frb_clean_project.pdf, frb_clean_project.ipynb Pure code that can run submitted given data. There is over 60 gigs of raw data that I won't submit.
- data/testdata.zip Data used to generate Figure 12.
- data.pkl Processed data for model 1.
- feats.npy, labs.npy Processed data for model 2.
- 244065364_rfi.npy Example RFI.
- FRB20180916B_waterfall.h5 Example FRB.

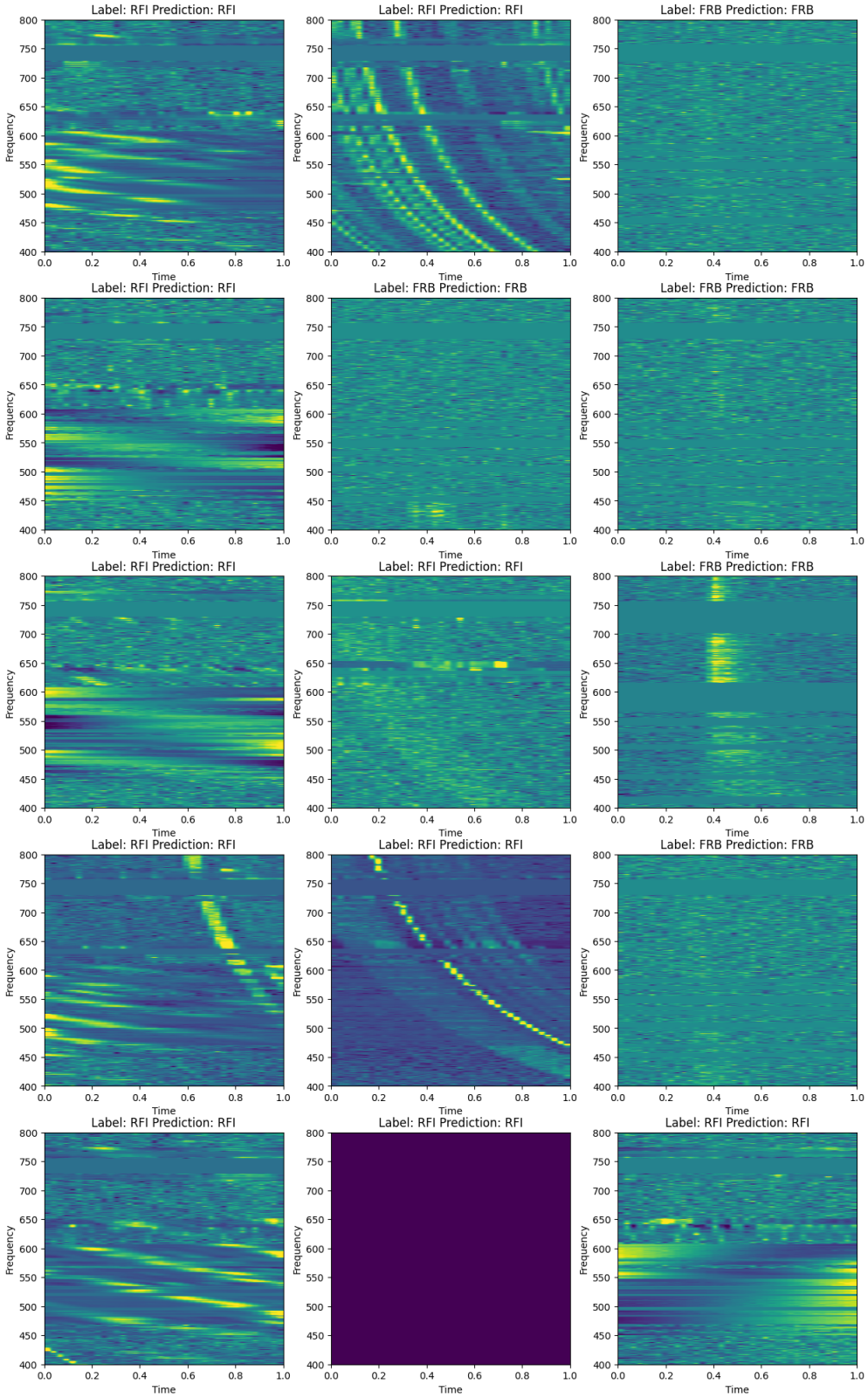


Figure 11: Test predictions with no dedispersion applied.

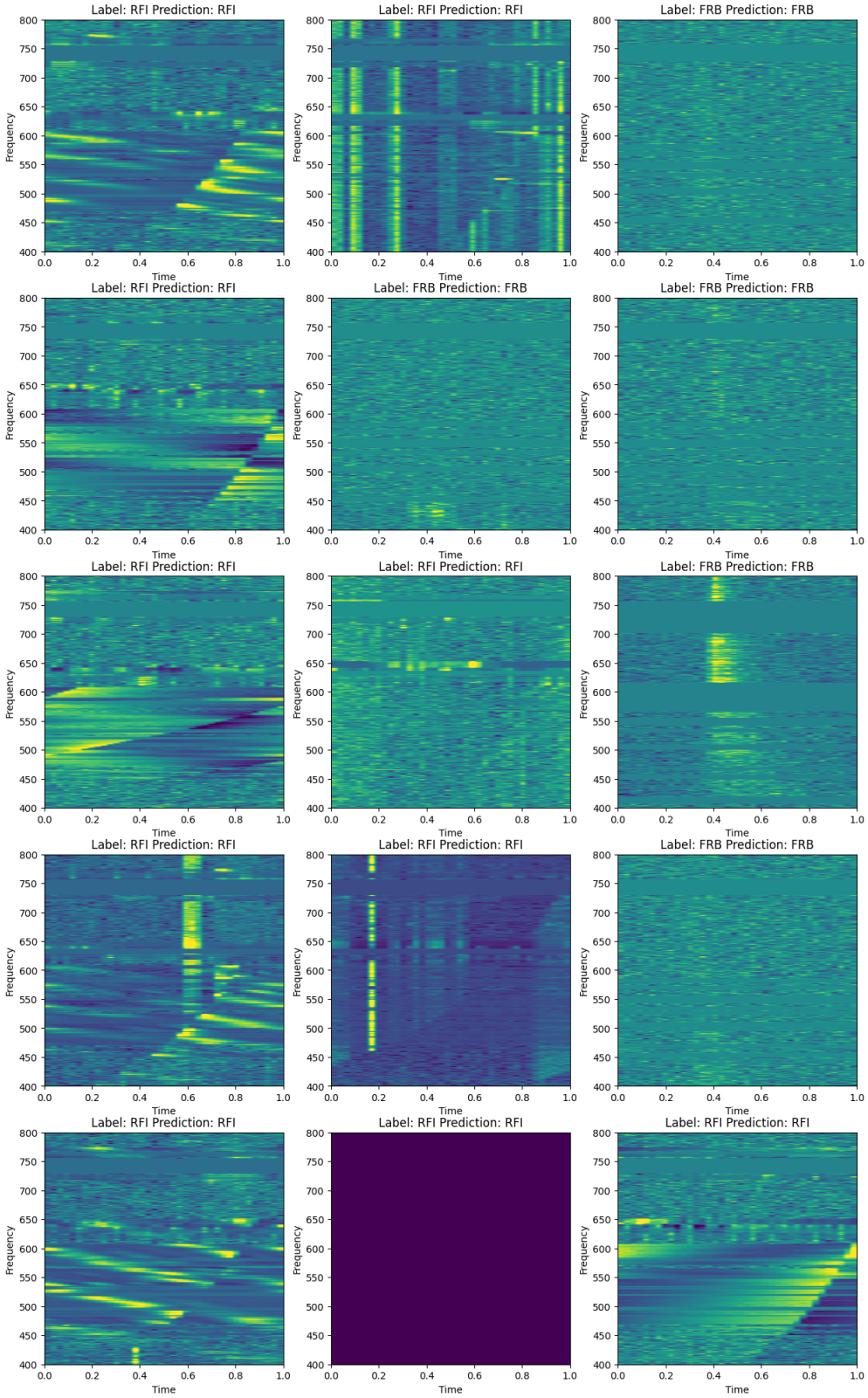


Figure 12: Test predictions with dedispersion applied

References

[1] Viviana Acquaviva. *Machine Learning for Physics and Astronomy*. 2023.

[2] Mandana Amiri et al. "The First CHIME/FRB Fast Radio Burst Catalog". In: *The Astrophysical Journal Supplement Series* 257.2 (Dec. 2021), p. 59. ISSN: 1538-4365. DOI: 10.3847/1538-4365/ac33ab. URL: http://dx.doi.org/10.3847/1538-4365/ac33ab.

[3] S. R. Kulkarni. *Dispersion measure: Confusion, Constants Clarity*. 2020. arXiv: 2007.02886 [astro-ph.HE].

[4] D. R. Lorimer and M. Kramer. *Handbook of Pulsar Astronomy*. Vol. 4. 2004.

[5] "Observations of fast radio bursts at frequencies down to 400 megahertz". In: *Nature* 566.7743 (Jan. 2019), pp. 230–234. ISSN: 1476-4687. DOI: 10.1038/s41586-018-0867-7. URL: http://dx.doi.org/10.1038/s41586-018-0867-7.

[6] Mohammed Chamma Shiny Antonio Herrera Martin. *chime-frb-open-data*. https://github.com/chime-frb-open-data/chime-frb-open-data/tree/master. 2023.

[7] Prateek Yadav. "Applying convolutional neural networks to classify fast radio bursts detected by the CHIME telescope". PhD thesis. University of British Columbia, 2020. DOI: http://dx.doi.org/10.14288/1.0389889. URL: https://open.library.ubc.ca/collections/ubctheses/24/items/1.0389889.