Draft version April 18, 2022
Typeset using LATEX default style in AASTeX631

Machine Learning and Simulation Strategies To Improve Fast Radio Burst Detection

Peter Xiangyuan Ma

ABSTRACT

In the last decade of radio astronomy, Fast Radio Bursts (FRBs) have been of acute interest to astronomers in part due to their highly energetic behaviour, and because of the mysterious processes that give rise to such exotic events. To help uncover these mysteries, one principal task is to discover more of these FRBs using radio telescopes like the Canadian Hydrogen Intensity Mapping Experiment (CHIME). In this paper we attempt to advance the search for FRBs by tackling two major problems: firstly developing simulations for FRBs in order to investigate false positive rates of citizen science volunteers, and secondly, developing machine learning based tools for radio frequency interference (RFI) mitigation. In this project, we conclude that FRB simulations for citizen science could be solved simply if we have an offline version of BONSAI, which is an algorithm running at CHIME searching for FRBs. However, we also concluded that the actual implementation proves too difficult for this current setting. In terms of developing the machine learning RFI sifter, we now believe there exists promising algorithms that can replace existing models. This new model implements an XGBoost model in combination with the current Support Vector Machine (SVM) to produce the best RFI Sifter to date with an order of magnitude improvement in both false positive rate and false negative rate. This upgraded model now runs live on the CHIME pipeline.

1. INTRODUCTION

Fast Radio Bursts (FRBs) are defined to be bright millisecond radio pulses originating from (mostly) extragalactic sources and were first discovered in 2007 by (Lorimer et al. 2007). The initial surprise of discovering the first FRB was due to the mystery around this exotic event and the high amounts of energy released through a mysterious process. It is estimated that FRBs release energy at a rate of 10^{35} W (Petroff et al. 2019). Even today, the astrophysical process for these bursts are not completely understood. However due to collective efforts in studying FRBs we have developed candidate theories. These mysterious events pique the interest of astronomers around the world, firstly they help researchers study the extremes of astrophysics and secondly FRBs can help probe seemingly invisible parts of the universe due to their extragalactic origin and highly energetic nature (Petroff et al. 2019).

The study of FRBs has forced astronomers to reexamine our current understanding of how high energy events occur in the Universe. Some of the early conjectures were that FRBs were caused by cataclysmic processes like supernovae, as the energy output was within a few magnitudes of such events (Bhandari et al. 2020). However the theory was found to be incomplete as the discovery of repeating FRBs contradicted that idea (Collaboration 2019). Had the object been destroyed in the event, it should not repeat (Collaboration 2019). Another conjecture was that these were signs of technosignatures which originate from an extraterrestrial intelligent civilization (Lingam & Loeb 2017). One such contradiction was that after studying FRBs they appear to come from naturally occurring sources rather than signs of engineered technology (Lorimer 2018). Yet another conjecture is that they occur from merging compact objects, like neutron stars (Totani 2013). The main justification was that the rate of FRBs matches the binary neutron star merger rate and that these mergers should produce observable FRBs (Totani 2013). Astronomers have suggested that the cause was magnetic braking created when the magnetic fields of neutron stars interact during coalescence (Totani 2013) (Petroff et al. 2019). Currently a more widely adopted theory is that FRBs originate from magnetars, which are magnetically powered neutron stars with very strong magnetic fields, specifically from either starquakes or the rearrangement of magnetic fields (CHIME/FRB Collaboration 2020)(Bochenek et al. 2020). The justification is that these events produce enough energy to behave like an FRB and their physical size agrees to the variability argument (Zhang 2020). The variability argument says that the object emitting a pulse must be no larger in length than the time it takes for the pulse to travel during the duration of the pulse (Zhang 2020). Overall, great

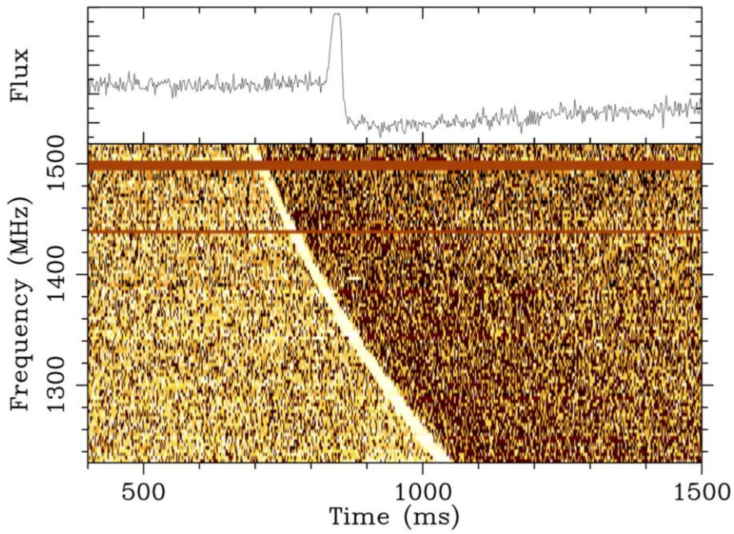

Figure 1. This is the dynamic spectra of a detected FRB. We see that that there is a delay in the time of arrival as a function of frequency which is caused by the ISM that the signal traveled through. (Petroff et al. 2019)

effort has been made by researchers around the world to study and understand these extreme events in our Universe.

Researchers have also attempted to use FRBs as probes for studying invisible parts of the universe (Petroff et al. 2019). One simple example is the study of the interstellar medium (ISM) using FRBs by looking at their dispersion (Macquart 2018). As FRBs travel through the ISM, the light disperses as it interacts with the medium, causing a change in speed and effectively, a delay in its time of arrival to our receivers (Macquart 2018). This change in speed is caused by the cold plasma in the ISM, which contains free electrons from all the ionised hydrogen (Condon & Ransom 2016). [Specifically, the free electrons act like a transmitting medium with an index of refraction that decreases with increasing frequency, this causes certain pulses to arrive sooner at high frequencies. (Condon & Ransom 2016)] This dispersion can be numerically quantified by the dispersion measure (DM) and can be computed directly from the data (de-dispersing dynamic spectra (Bassa et al. 2016)). The physical interpretation of the dispersion measure, is that it is a path integral by summing up the number electron density along the light of sight (Condon & Ransom 2016). More concretely, this DM value helps us gauge firstly how much "stuff" there is in that part of the universe and secondly, helps us gauge how far or where source might be coming from (Condon & Ransom 2016). For example, we now know that high DM events are most likely extragalactic because if they were to have originated from the galactic plane, there currently is not enough matter along that line of sight to have accounted for it (Petroff et al. 2019). Here is an example of a dynamic spectra with a dispersed FRB figure 1. There are other properties such as scintillations, scattering etc, that also help us study these seemingly invisible parts of the universe (Petroff et al. 2019).

Since FRBs are of interest to astronomers around the world, radio telescopes like the Canadian Hydrogen Intensity Mapping Experiment (CHIME) have been used to hunt for these exotic events (Petroff et al. 2019). Initially CHIME came online for cosmology experiments, but it soon proved to be effective in finding FRBs because of its design. CHIME is built with 4 semi cylindrical interferometers and was designed to study the 21cm line specifically to probe the accelerating expansion of the universe over a redshift range (FRB Collaboration et al. 2018). By design CHIME has a large field of view 100 deg by 2 deg (FRB Collaboration et al. 2018). For reference the Greenbank Telescope has a field of view of only $\sim$ 10 arcmin at 1GHz (Gajjar et al. 2018). A big part of why CHIME is good at finding transient events like FRBs is in part due to its large field of view combined with the wide frequency bandwidth, high sensitivity, and a powerful correlator, making CHIME a FRB hunting machine (FRB Collaboration et al. 2018).

Despite having discovered $\sim$ 500 FRBs, we are still limited by the number of FRB detections (FRB Collaboration et al. 2021). Discovering more FRBs remains one of the principal challenges for CHIME/FRB project. There currently exist 3 big directions one can take in tackling this problem, the engineering aspect of improving the physical telescope, improving the software and automation pipeline and scaling the human detection aspect either with volunteers or through other computational means. In this project we attempt to tackle the latter two problems. These problems consist developing simulations for citizen science volunteers and improving an automated sifter model called the RFI Sifter.

In the subsequent sections we first explore the simulation problem for citizen science. We introduce the status quo in section 2.1 and the problem statement in section 2.2. Then we describe the methods and possible solutions in section 2.3 and we comment about the results in section 2.4. With respect to the RFI Sifter, we once again introduce the status quo in section 3.1 along with the problem statement in section 3.2. We highlight the experimental approaches and setup with section 3.3 and we discuss the results in section 3.4. We elaborate on the interpretations of the results in section 3.5 and we finally discuss the live results in sections 3.6.

2. FAST RADIO BURST SIMULATIONS

2.1. *Citizen Science Introduction*

Finding FRBs are fundamentally a transient search problem, made difficult by the torrent of data collected from observations. Most of the detection process flows through automated algorithms such as BONSAI, which is an algorithm used to determine the significance of an event by computing physical parameters of an event and to gauge whether or not to pass the result for human inspection by the tsar's. One metric to determine the significance is the Signal to Noise Ratio (SNR) of an event. However, due to the vast quantity of data we only send high SNR events about 8.5 SNR or higher are sent to the tsar's (FRB Collaboration et al. 2018). We want to expand this search to lower SNR, and so collaborators at CHIME decided to divert the flow of lower SNR events, 7.5 to 10, for classification by citizen scientists on sites such as Zooniverse. Zooniverse is a citizen science web portal in an effort to crowd source public efforts in conducting science with big data. Despite the public's excitement, and the vast quantity of now classified events by volunteers, quantifying the **quality** of the classification in a robust statistical manner has not been done before.

2.2. *Problem Statement*

The problem is: **"how can we build realistic controlled tests to evaluate the quality of volunteer classifications?"** We currently lack robust statistical metrics on the quality of the events classified by volunteers. As of now, we have diverted and received 777,887 classified candidates from the volunteers, however we can't do meaningful science without gauging the quality of these classifications. This can be solved if we can test the volunteer's classification ability. This however is currently impossible since we cannot control whether a given observation contains a burst or not. Thus this becomes a simulation problem. Thankfully, there currently exists tool for creating simulated bursts. However there does not exist tools for creating false events, i.e candidates with no bursts. Thus we need to devise a method to create observational data where we can guarantee with certainty that no real FRB appears in a given event. Thus we will focus on specifically: **How can we simulate data where it is a guaranteed false event?**

2.3. *Methods*

Initially there were many possible routes in addressing this issue. However by the end, almost all of them had its own unsolved problems and remains unsolved. Here were the attempts:

1. Take classified events vetted by the tsar's as "not FRB" and give them back to the volunteers.
   This was the first naive solution. The problem with this approach is that nobody, not even the tsar's themselves can guarantee there is truly no real FRBs in the data. Tsar's have a potential in mistaking an event to be false while there could have very well been a real FRB. In other words, tsar's have a false negative rate which would be an error that is very difficult to propagate into the statistics done on citizen science work. Furthermore, we do not even know what this false negative rate is to begin with.

2. We could manipulate real data to make any real signal nonphysical (data that cannot be created from a naturally occurring source) and then hand it back to volunteers. *This initially is a sound approach. An easy means of*

creating this nonphysical data is reversing the ordering of the frequency channels in the dynamic spectra like the data shown in fig 1. We know that naturally occurring signals have a dependence on frequency and so breaking that dependence makes it nonphysical. The caveat is that, all observations given to citizen scientists have already passed through the automated BONSAI *algorithm. BONSAI is a FRB search algorithm that works by attempting to computing physical parameters of a candidate event (FRB Collaboration et al. 2018). If we manipulate the data downstream of that checkpoint, we can no longer guarantee that this synthetic event would have passed BONSAI. In other words, the volunteers may never normally see these kinds of data! This would be an unfair test for the citizen scientists to do.*

3. Take synthetic data, feed it back to BONSAI to double check that it would have passed BONSAI in realtime. *Once again this was a straightforward solution. However,* BONSAI *is built to run in* ***realtime****. Specifically there are realtime search parameters unique to the telescope during observation, such as the variance estimation for the telescopes noise, which are not captured and saved with the each event triggered by* BONSAI. *In other words, there is vital information necessary for running* BONSAI *in offline mode that is lost to the forever.*

4. We could save the unique parameters from BONSAI and repeat approach 2,3. *This is problematic. After talking to multiple CHIME collaborators [Kendrick Smith, Dustin Lang, Ziggy Pleunis, Chitrang Patel, Mike Walmsley], it was determined, almost unanimously, that this would be a difficult undertaking which would involve the work of multiple collaborators and would stretch past the duration of this project time line.*

5. Last Attempt: BONSAI Comparison test. We feed BONSAI a real observation, then we feed BONSAI a simulated observation. If the returned parameters of the fake observation $\sim$ real observation we can conclude the fake would have probably also passed BONSAI in real time. *This stands as the only possible solution that could potentially satisfy our criteria.*

2.4. *Results and Discussion*

Of the many solutions explored we decided that the last proposed solution appears to have the fewest problems and is feasible without involving multiple CHIME collaborators. However, this attempt was quickly thwarted once again before results could be made due to the decision to pivot the project to a new direction. We decided to pivot before getting results because we claim that the last approach still remains incomplete even if implementation is successful, as it does not incorporate the variance estimation that BONSAI uses.

The consensus is that we still can not truly guarantee that by inputting a substitute variance versus the original variance would not change the relative outputs respective to the real observation and the simulated observation. For example what if variance fluctuates by frequency and if we mismatched the frequency it makes it easy for BONSAI to pick out the simulated signals and rejects it more often? The feedback from multiple collaborators on the project believe that this is an unsatisfactory approach, as BONSAI truly needs accurate variance estimation in order to perform close to how we expect it to perform.

2.5. *Forward*

In terms of forward development, many collaborators have agreed that the ideal path to solving the issue is to properly develop an offline version of BONSAI. This means fully supporting the sensitive input parameters and data. Future development would most likely spearhead this area of development.

3. MACHINE LEARNING RFI SHIFTER

3.1. *RFI Sifter Introduction*

One of the most scientifically rich means of storing data is storing baseband data. Baseband data is the raw voltage data saved from the detectors and it is the most unprocessed data we can collect, however these data products take up a lot of space on disk (FRB Collaboration et al. 2018). Because of that, we need to save only the highest quality of candidates in this form or else we run into storage issues. Thus we need a model to determine, in real time, what data to persist on disk and what to toss out with a very low false positive rate (on the order of 0.1%). Currently, this decision is made by an algorithm called the RFI Sifter. The data fed into the RFI Sifter is the header data for each

		SVM Prediction	
		False	True
actual value	False	True Neg 35721	False Pos 11
	True	False Neg 952	True Positive 60913

Table 1. Support Vector Machine confusion matrix.

event, which includes features like: SNR vs DM, DM, and time/date etc. Currently this RFI Sifter uses a model called a support vector machine (SVM) to classify (Cristianini & Ricci 2008) the header data for each observation as either true for astrophysical events or false for RFI. The SVM, in short, is based on a classification scheme where we fit some kind of a line or polynomial or in the general a hyperplane that divides the classified data in some high dimensional feature space (Cristianini & Ricci 2008).

We tested the existing model on $\sim$ 100k hand classified events taken in the month of January 2022[1]. The current performance is given by the following confusion matrix, table 1. [Dataset specification elaborated in section 3.3]

3.2. *Problem Statement*

The problem is: **"can we build a new machine learning (ML) model that can improve the existing SVM model's false negative rate, while also maintaining, if not improving, the false positive rate?"** The current SVM model is rejecting nearly 2% of all possible real astrophysical events shown in table 1, and the reason we use this model in production is because it is good at maintaining a very low false positive rate. However there exist more sophisticated machine learning based approaches that could potentially outperform the support vector machine. The SVM approach is quite simplistic which lends itself to areas of improvement in both pure performance and in interpretability (SVM models are "black box" and difficult to understand).

3.3. *Methods*

3.3.1. *Training and Benchmarking Dataset*

The dataset was provide by Adam Dong who hand classified 97,607 of events for testing and 406,092 samples for training. This is the same dataset used to test the SVM model previously. The data contains 28 features of which we throw out the date, the DM, SNR, and similar features, as a means of making its performance as close to the SVM as possible. We also want to force the model to not discriminate against high DM or high SNR candidates, as high DM sources can be both RFI and real FRB's.

Here is the complete list of features used for the training and model execution. The feature list is described in more detail in section 5.2 of appendix. *'max_coherent_snr','incoherent_snr', 'max_to_second_snr_ratio', 'max_level1_grade', 'mean_level1_grade', 'snr_weighted_level1_grade', 'snr_weighted_tree_index_weighted_level1_grade', 'std_level1_grade', 'min_tree_index', 'mean_tree_index', 'snr_weighted_tree_index', 'snr_vs_dm', 'std_tree_index', 'ew_extent', 'ns_extent', 'group_density', 'max_snr_ns_beam', 'snr', 'beam_activity', 'coh_dm_activity', 'incoh_dm_activity', 'avg_l1_grade', 'event_no'*

We can also visualize data that is either a positive event or a negative event. Typically it is NOT illuminating to look at the raw numbers, it is illuminating to specifically look at DM with respect to time which allows us to easily

[1] Data was hand classified by Adam Dong

visually verify a positive or negative event. An example of such a comparison is figure 5. On the left hand side bottom corner we see that vertical streaks are indications of RFI, where as horizontal streaks are indications of astrophysical events. The justification is that RFI tends to have clusters of sporadic changes in DM and are short in time duration. Pulsars, typically remain the same in DM for long periods of time. FRBs are also high in DM but appear very short in duration and so the clumps of vertical streaks aren't FRBs because they occupy too large of a window in time.

3.3.2. *Experimented Solutions*

There were many possible routes considered for addressing this issue.

1. Apply a traditional deep neural network or multilayered perceptrons as described in (Rumelhart et al. 1986). *This was the first naive solution. Applying a deep neural net on the order of* 10^5 *parameters on a dataset of* 22 *features. These neural networks are built from compositions of single layer perceptrons. We believed that the ability for neural networks to approximate nonlinear behaviours of arbitrary functions would help improve the simplistic approach of the linear kernel in an SVM.*

2. Apply a clustering algorithm (Mannor et al. 2011). *This was the second naive solution. We chose to experiment with this approach because we wanted to see if there existed any obvious clusters in the feature space that could make the classification problem simpler.*

3. Applying a Random Forest Model (Breiman 2001). *Random forests belong to the class of ensemble learning approaches. Ensemble models are models that orchestrate multiple simpler models, such as individual decision trees, and aggregate the results of all the simpler models for a final classification. Random forests build an ensemble of decision trees, hence a forest, and uses each tree to act as a vote of confidence for the final classification making it robust against over fitting the data (Breiman 2001). Our model builds decision trees with a max depth of 22. The max depth was chosen because there are only 22 features. This approach has been known to be a powerful predictor on tabluar data in ML literature (Breiman 2001).*

4. Applying an XGBoost Model. *This approach is the same as the random forests model except its trained via gradient boosting and is meant to improve training time on large tabular data and also improve run time (Chen & Guestrin 2016). Gradient boosting is when we iteratively build trees by incoorporating the information of the errors made by previous trees in constructing the new tree (Chen & Guestrin 2016).*

5. Applying a Hybrid XGBoost Model (Chen & Guestrin 2016). *This approach leverages both the predictive power of the SVM model and the XGBoost model. In this classification scheme we take the prediction of both the SVM and XGBoost model and guage the relative confidence between each inference. This confidence threshold is determined empirically to be 0.8, in section 3.5. Should the SVM decide, with high probability, that a given event is most likely a target of interest, we override the decisions made by the XGBoost and take in the original SVM classification. We chose to experiment with this technique in order to tackle the FRB recall problem discussed in section 3.4.2*

3.4. *Results*

Of the many solutions attempted we will briefly outline the results from each of the models we tested. After deciding the algorithm to move forward with we then apply a parameter search and determine the optimal parameters for a given model (this is quite computationally expensive). Optimization is covered in appendix section 5.3.

3.4.1. *Benchmarks*

We evaluate the true positive, true negative, false positive and false negative rates in the following confusion matrices.

1 - We applied a traditional deep neural network (Rumelhart et al. 1986) with Tensorflow API[2], which is a Python 3 module in developing deep learning models , and achieved the following result below in a confusion matrix, table 2. The model has a false positive rate of 0.25% and a false negative rate of 0.07%. Model parameters can be found here [3]

[2] https://github.com/tensorflow

[3] https://github.com/CHIMEFRB/FRBgen/blob/main/Combined_Benchmarking_RFI_Shifter.ipynb

		Deep Neural Net Prediction	
		False	**True**
actual value	**False**	True Neg 35634	False Pos 98
	True	False Neg 49	True Positive 61816

Table 2. Deep neural network model's confusion matrix. We see that the false positive rate is an order of magnitude higher than the original SVM making this approach difficult to implement.

		Kmeans Prediction	
		False	**True**
actual value	**False**	True Neg 34072	False Pos 1660
	True	False Neg 27	True Positive 61838

Table 3. Kmeans model's confusion matrix. This model has the highest false negative rate with a 100x increase than the SVM model making it difficult to use once again.

Apply clustering algorithm (Mannor et al. 2011) 2 - We tried multiple clustering algorithms and we report back only the highest performing model which was the kmeans clustering approach giving us the following confusion matrix found here table 3. The model has a false positive rate of 4.7% and a false negative rate of 0.04%. This model approach appears to perform empirically worse than that of the neural network. [model parameters can be found here [3]]

Applying Random Forest Model (Breiman 2001) 3 - This was the second best performing model and using 1000 estimators we get the following performance on table 4 The model has a false positive rate of 0.09% and a false negative rate of 0.07%. This model appears to perform better both in terms of false positives and false negatives.[3]

.

Applying XGBoost Model (Chen & Guestrin 2016) 4 - This the best model which is similar to the random forest model and gave the following confusion matrix, table 6. The model has a false positive rate of 0.02% and a false negative rate of 0.2%. From empirical tests this model performs the best out of all the experiments we have tried. [model parameters can be found here [3]]

Applying Hybrid XGBoost Model (Chen & Guestrin 2016) 5 - This is the second best model with a false positive 0.02% and a false negative rate of 0.2%. We see clearly that when applying the hybrid approach we sacrifice a bit of the predictive power by allowing the original SVM model to make a part of the predictions. This model performs the second best when testing on the same dataset. Model weights can be found here[3]

3.4.2. *FRB Recall Problem*

		Random Forest Prediction	
		False	**True**
actual value	**False**	True Neg 35698	False Pos 34
	True	False Neg 46	True Positive 61819

Table 4. Random Forest model confusion matrix. The false positive rate is almost onpar with the SVM and has a much lower false negative rate.

		XGboost Prediction	
		False	**True**
actual value	**False**	True Neg 35725	False Pos 7
	True	False Neg 134	True Positive 61731

Table 5. XGBoost model confusion matrix. Currently this is best performing model when only comparing the results. Very low false positive rate and false negative rate with respect to the SVM.

		Hybrid XGboost Prediction	
		False	**True**
actual value	**False**	True Neg 35721	False Pos 11
	True	False Neg 137	True Positive 61729

Table 6. Hybrid XGBoost model confusion matrix. Similar performance as the standalone XGBoost model but a bit worse in false positives and false negatives. Still major improvements with respect to the SVM.

Despite the clear indications that the XGBoost model outperforms the original SVM model and every other benched-mark model on the $\sim$ 100k hand classified data set, the XGBoost fails in one important domain. When testing the XGBoost model on known FRBs the model discriminates against them on an order of 0.5-1% in table 7.

However, the issue nearly disappears when we test the hybrid model on known FRBs. By looking at another confusion matrix found here in table 8 we see a clear indication that when combining the original SVM model we recall more known FRBs in comparison to the standalone model.

SVM Prediction

actual value		**False**	**True**
	False	True Neg 0	False Pos 0
	True	False Neg 0	True Positive 92

XGBoost Prediction

actual value		**False**	**True**
	False	True Neg 0	False Pos 0
	True	False Neg 1	True Positive 91

Table 7. We have the SVM performance on the left and the XGBoost performance on the right. Here when we tested the model on known FRBs from 2022 onwards, we see we have approximately a 1% error rate! This is performance is not acceptable as this is what we are trying to detect.

SVM Prediction

actual value		**False**	**True**
	False	True Neg 0	False Pos 0
	True	False Neg 11	True Positive 1114

Hybrid Prediction

actual value		**False**	**True**
	False	True Neg 0	False Pos 0
	True	False Neg 13	True Positive 1112

XGBoost Prediction

actual value		**False**	**True**
	False	True Neg 0	False Pos 0
	True	False Neg 49	True Positive 1076

Table 8. Here when we tested the model on known FRBs from 2021 and on wards we see a drastic improvement in performance and is on part with the original SVM.

3.5. *Discussion*

3.5.1. *Theory and Model Analysis*

Before we make final decisions, we need to verify how the model makes decisions and interpret the results. Every scientist should always question "why does this model work" and "why does that model not work"? Unfortunately, interpretability is one of the largest unsolved problems in modern machine learning. And the current state of the field is just unprepared to answer this question formally with any satisfying rigor. Thus my analysis on the machine learning methods are ad hoc and founded on empirical evidence only. As Howard Stark best put it "I am limited by the technology of my time".

Firstly, we investigate the design. The XGBoost model has no real architectural differences with the Random Forest model (Chen & Guestrin 2016)(Breiman 2001), thus for discussion at the moment we stick with the random forest for simplicity. A plain random forest has proven to outperform other approaches in our tests on table 4, and we hypothesize it is because the model is better designed for the problem. Specifically the model builds decision trees from the ground up and so it has the choice to select which features are important for inference. The model can choose if a feature is used or not when building each tree. The act of choosing occurs if one places restrictions to the maximum depth of a tree. In other words, if you tell the model to make decisions by looking at 10 out of the 22 total features the model *must choose* which features to optimally build these trees. This approach is fundamentally different than SVM and Kmeans, where the choice is *instead* being made by the model. Typically we do not want to make the choice for the model unless we understand precisely the underlying relationships between these features. In this case, it is not well understood. For example, we do not know for sure if SNR is truly more important than SNR vs DM when making the final decision if it is RFI or Astrophysical. Furthermore the design of the model allows the features to be weighted differently. The hierarchy of a tree is the weighted importance of each feature. In other words, higher up in the decision tree, the more effect that decision has on the final outcome. The ability for the model to weigh features in the design of the architecture is crucial to its success. We have empirically proven this in section 3.4.1 by testing the random forest model against models that do not have the ability to weight features. The result is they tend to perform worse, shown in the benchmarking section of the paper. Section 3.4.1.

So why specifically do the SVM and KMeans algorithms perform poorly? These two algorithms make the key assumption that input features are uncorrelated and thus orthogonal. This assumption is made as the KMeans model is optimised using a euclidean norm as a metric (Mannor et al. 2011), similarly for the SVM (Cristianini & Ricci 2008). Thus, they perform worse against models that do not make that assumption. However, the reason why SVM outperforms the K-Means is that the SVM took in pruned data. Before feeding the input data, previous engineers decided to cut out redundant features or features that are correlated with other features. The input data for the SVM is only a 10 dimensional feature vector whereas the K-Means was a 22 dimensional vector. The reason why we chose to no longer do this dimensionality reduction step prior to building a model is that we did not want to make decisions that we do not have too. Thus with this situation, it's clear why a decision tree architecture would outperform these other simpler approaches.

We have also shown that the random forest outperforms the classical neural network in table 2 and 4, in part due to an overfitting problem. Neural networks are designed as universal function approximators (Rumelhart et al. 1986). In theory this is a one-size fits all solution, but in practice that is far from the case. Approximating patterns in training does not guarantee generalizability. Historically, plain neural networks perform poorly without introducing some kind of inductive biases from specific problem domains (Mitchell 1980). Inductive biases are assumptions made about the data / problem before training and baking this assumption into the design of the architecture. For example, if a neural network is meant to work with images, we attach a convolutional layer to aggregate spatial data using a sliding kernel. This is because we know images have spatial relations and to help the neural network we tell it to make that assumption without explicitly learning it. In practice, plain neural networks do not work universally and thus they either underfit the data, or we fill a network with lots of parameters and the model overfits the data. This is

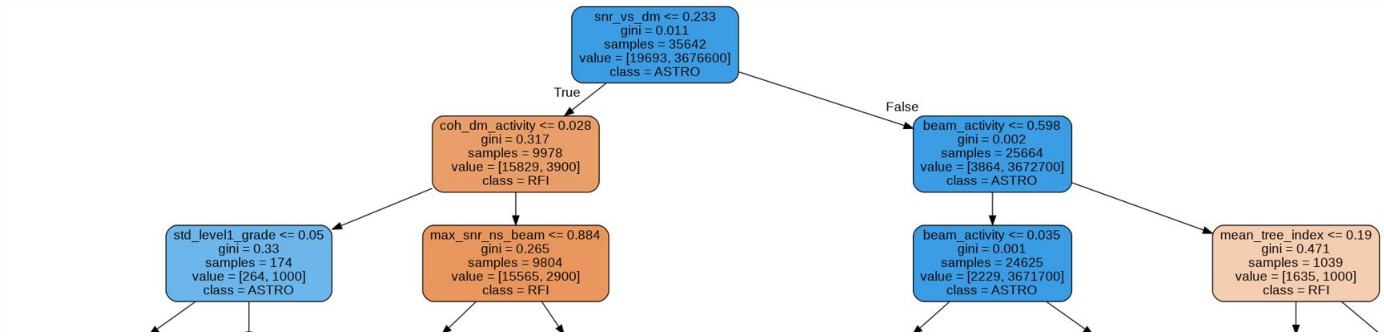

Figure 2. This is the branches of a single decision tree in the model. The first line in each node describes what feature we look at and the number associated is the threshold in which we gauge the decision. We can see that the graph made a cut at the first node, which was SNR vs DM. We see that it makes a cut at a threshold of 0.23 between SNR and DM ratio. This makes relative sense because RFI typically would have a larger SNR to Dm ratio, since SNR would be larger WITH respect to DM. Thus if SNR vs DM is less than that threshold it would be more likely to be Astrophysical indicated by the blue color of the node. One can feasibily continue doing this process, however there are over hundreds of these trees, and each is 23 decision deep. It simply is not feasible to truly understand how the model is making decisions.

exactly the case with this problem. There are patterns and relations tangled with each feature that is not trivial. Not only are the features correlated, some features are continuous and some are discrete even! We did not tell the model beforehand what is important or what relationships can be assumed within the data. Thus we did not get favourable results. Furthermore another key difference is that the neural network works in a "deconstructive approach", whereas the random forest works with a "constructive approach". A "deconstructive approach" like the neural network is when one starts with assuming every feature is equally important and as it goes on training it can decide to drop certain features. In contrast to the "constructive approach", the decision trees work by starting with one feature and growing the tree from a bottom up approach. Intuitively, when humans approach this problem, we clearly opt for the latter as we know not all features are useful, figuring out what is useful first makes more sense than figuring out all the combinations of things that are not useful. It is now clear why a neural network would have been poorly suited for the problem.

Having justified why random forests would in theory perform better, what kinds of decisions are these forest of trees making? We can look at the internal decision process as a sanity check. However, be wary, as few humans can operate 3-4 orders of decisions deep and so when the random forest produces between 10-23 layers of decisions, we are still left with an uninterpretable mess. However as a sanity check we can still look at the first 2-3 order at fig 2

We have yet to examine why the random forests outperformed by the XGBoost model seen in table 4 and 6. The key difference between the XGBoost model and the random forestsis the training process, the XGBoost is sequential and the random forestsis parallel (Chen & Guestrin 2016)(Breiman 2001). In an random forestsmodel, the model builds each tree independently. These trees are grown by randomly sampling data from the training set and building a tree on that sample. In total we have hundreds of these trees each independent from one another. Now suppose that this process produces a strong predictor, i.e a tree that is really good at classifying the data. The performance of that single tree has negligible effect on the overall outcome, because the final decision is made by taking an average of all the other trees. On average we have weaker trees. In other words, random forests would often by chance build a strong predictor but this predictor has little influence because we take overall statistics of the entire forest. However for the XGBoost model, each tree is further built upon sequentially. In other words in a cycle, we take a tree, we compute the error and use the error to update the tree and then grow that tree. If the model by chance gets a strong predictor, this predictor will influence the performance of models downstream because it is sequential in nature. This is the motivation as to why XGBoost is a stronger predictor on average.

Lastly, we need to understand why did the XGBoost model suffered in performance in recalling known FRBs shown in 7? A big reason why this is the case is because the data was trained on candidates from recent observations and when

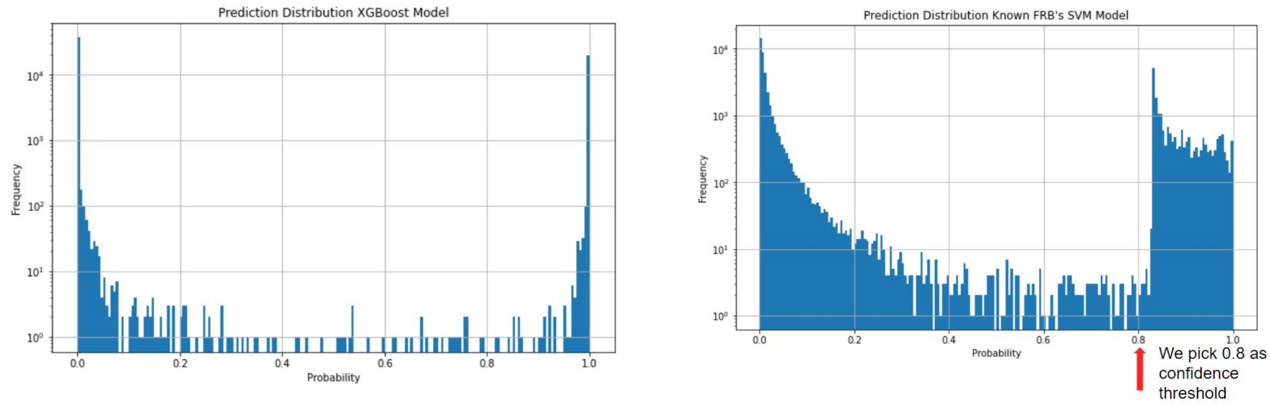

Figure 3. On the left we have the prediction/confidence distribution of the XGBoost model and the right we have the prediction distribution of the SVM model. We see that if we were to index the distribution on the SVM model, we can leverage that models predictive power to our advantage in predicting known FRBs. Details are described in text.

testing known FRBs they can range to 1-2 years in the past; during such time the telescope configurations may have changed. We also believe that having features that correlate with DM might have made the model discriminatory agaisnt high DM events as high DM events tend to be RFI. An example of features that are not orthogonal to DM is SNR to DM. Although we removed DM from the feature list as we know for a fact that will directly discriminate against high DM events, we were not sure how to approach features when they might have correlations with DM.

The reason why the hybrid model outperforms the XGBoost model is because of the tactic of allowing the SVM to override certain XGBoost predictions when the SVM predicts positive events with high confidence/probability. This helps preserve the original SVM's FRB recall ability. This is in part because the SVM has a vastly asymmetric distribution of probability scores when classifying only FRBs. In other words, when given a known FRB, the model gives it a near 99% probability it is of astrophysical origin which is ideal. But we see that just a standalone SVM model can be improved using an XGBoost model. The SVM lacks making correct decision on events that it is less confident on. We see this as there number of samples below the 0.8 threshold greatly outnumbers the XGBoost's distribution in fig 3. Thus in which case we take the XGBoost's prediction as we have empirically proven in section 3.4.1 to correctly classify more real astrophysical events while maintaining the same false positive rate specifically table 8.

Now, the important metric is deciding what is considered "confident". A simple way of deciding is by looking at the distribution of the probability/confidence scores over all the positive samples see (fig 3). We see that the SVM distribution is highly asymmetric, and thus to extract the most value, one would select the threshold where there is a drop in the number of correctly predicted true positives. The rest will be dealt with by the XGBoost model.

However, one may question the actual performance of the model, as this hybrid model still misclassified 2 known FRB's shown in table 8. To rebut this point, we further tested the model on 10 days worth of data. In the meantime we found 2 more promising FRB candidates by eye, to which the Hybrid model also agreed with us but was rejected by the existing SVM model. We believe that if we span the time in which the model is in production across 1 -2 years, the increased predictive power from the XGBoost correctly classifying what the SVM model deems to be RFI would not only make up the 0.2% error but will greatly surpass the SVM's performance. Thus we were confident that this is in fact the best solution to replace the SVM model.

3.6. *Live Production Analysis*

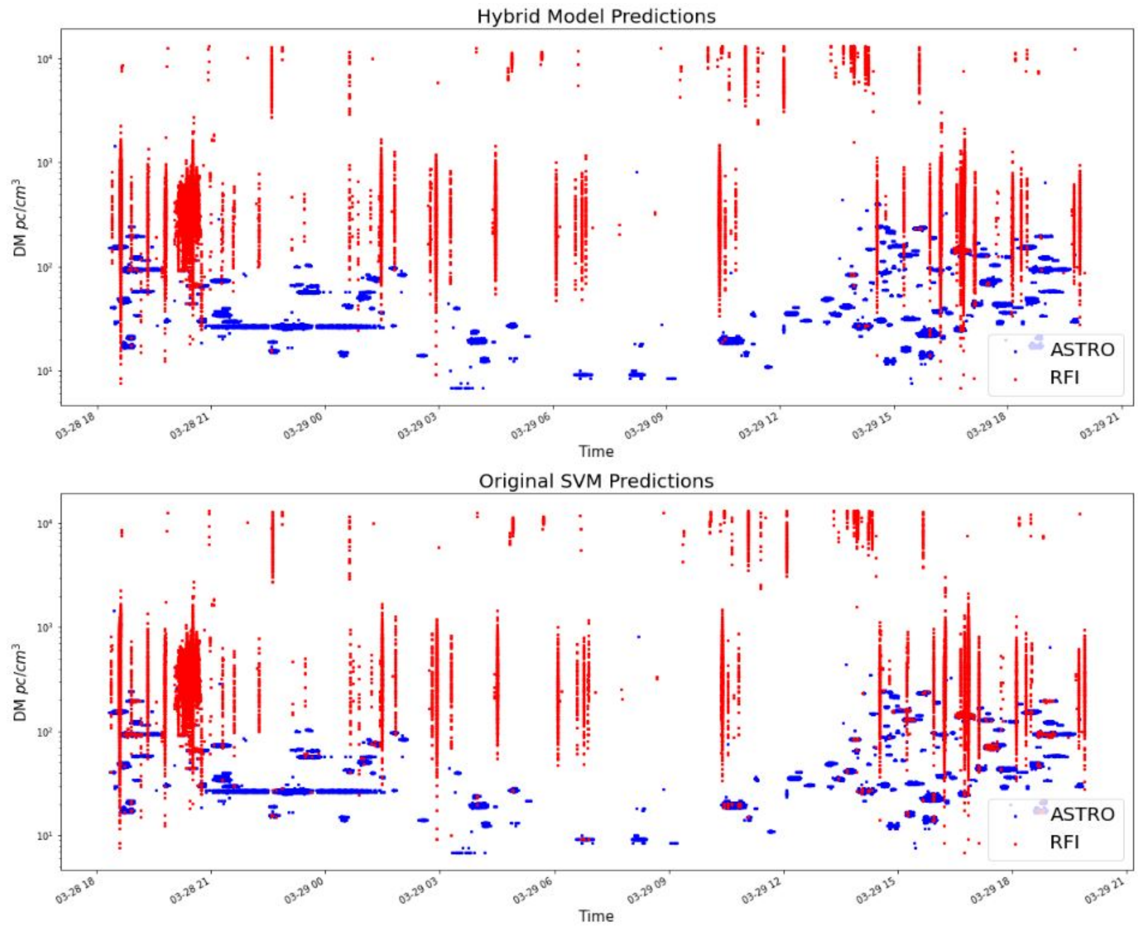

Figure 4. Here is an overview comparison. Notice the vertical clusters are examples of RFI where as the horizontal streaks are pulsar events and high DM singular events are potential FRB candidates.

After deciding that the hybrid approach is ideal, we move into deploying the model into a production testing environment where we collected the outputs to directly verify that the model does still indeed produce more favourable results than the SVM model.

To understand how a human might interpret favourable results, we plot the data with DM as a function of time like in fig 4. Interpreting these relationships, we see that when the DM is relatively constant for a long duration of time, this is an indication of a pulsar event. This is because the repetition of the pulses means we see relatively no change in DM with each pulse. We also see that when the DM forms a long vertical line meaning the duration is relatively short and the event has a changing DM, this is RFI. We locate potential FRB candidates when we see a single high DM event that only lasts for a single event. This indicates that it is fast in time and transient and extragalactic because of the high DM. Although this is trivial for us to pick out after having collected all the data, this problem is nontrivial on the live system, as the pipeline only sees a few events at once and does not have knowledge of past and future events when making a decision in real time.

We want to demonstrate that the model can successfully handle RFI storms. During the testing phase, we were hit with a storm which the hybrid model not only successfully blocked it out, but in a manner that was far more effective than the original SVM model this is show in figure 5. We also looked at its ability to retrieve real astrophysical events. We see that the new hybrid model once again outperforms the original SVM model that's currently in production as shown in figure 5.

On the timescale of 1 day we were able to retrieve ~ 200 new astrophysical results (after manually hand classifying the data) that was never found by the previous RFI Sifter. If we were to also look at the rate of positive events

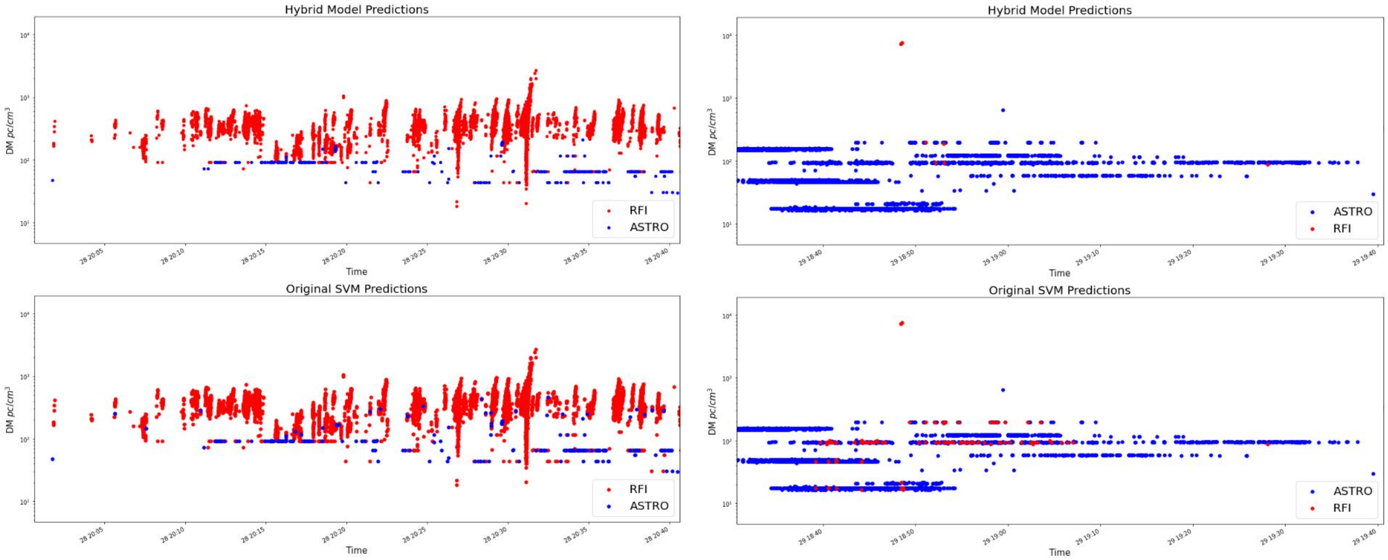

Figure 5. On the left we see an RFI storm which the SVM has misclassified $\sim$ 50 total more events to be astrophysical when in reality all should be classified as RFI since they are vertical streaks, which the hybrid model successful did. We see on the right, that there was a pulsar event indicated by the horizontal clusters, however the SVM decided to classify a number of these as RFI. The hybrid model once again successfully classifies them as astrophysical.

classified we also see they are within close proximity with each other table 9. This is an important consideration as too many positive events could clog up the system in vetting out the false positives.

Cut Off Val.pc/cm^3	SVM	Hybrid
DM>500 SNR>12	3	3
DM>200 SNR>12	573	578
DM>200 SNR>9	1053	1062
DM>100 SNR>12	7800	8008

Table 9. Table showing the recall rate of each model running on the live system with various, DM and SNR cuttoffs.

Looking at the recall using Table 9 it is important to note that on the threshold of DM between $100 - 200pc/cm^3$ the Hybrid model retrieved more results than the SVM. We know that based on the model's tested performances, these are most likely real astrophysical events that the original model is throwing away. This once again demonstrates the model's superior performance.
Furthermore we can assess the quality of the classifications made. Firstly we can look at false positives. During March 28th at 20h we received what appears to be an RFI storm on figure 5. The Hybrid model successfully identified the storm and returned very few positive events here shown in blue. On the other hand, the SVM had a number of misclassifications which demonstrates the robustness of the model to RFI.

Lastly we can assess the quality of the true positive classifications made here. One example of such are the pulsar events on March 29th 18:30-19h mark on figure 5. We see a series of misclassifications from the SVM model but correct classifications from the Hybrid model.

With these results coming in and with the thorough analysis done in the previous sections we are confident that the model outperforms the existing SVM RFI Sifter and we are currently in the process of formally replacing the existing SVM model. This push to replace the existing SVM model was also reflected by the people in the CHIME community as well.

3.7. *Future Work*

The future plan is to let the new and improved RFI Sifter help improve the quality and quantity of detections made by the CHIME software pipeline! An open discussion / problem that future students or researchers could tackle is looking into using the pipeline's buffer space. As mentioned before, the current state of the system does not allow a model to see past or future observations when making inference. This is technically incorrect: the system has a buffer on the order of a couple seconds and thus given +/- a few seconds for a single observation one has a few seconds of future and historical data. One path in improving the detection technique is to leverage this property of the system in some meaningful way.

4. CONCLUSION

For the project regarding citizen science and FRBs we have found a means of solving the false positive problem. However, after thorough investigation, we decided the scope of executing the solution extends beyond the initial plan for the semester and we have deferred solving the problem to a later course. With respect to the machine learning RFI sifter project, we have implemented and found promising algorithms, and have made the decision to replace the existing SVM model with the hybrid XGBoost model with a quoted improvement in false negative by a factor of 4-6x. We hope this new RFI Sifter helps expand the software capabilities of CHIME. In the end, we have successfully addressed the two largest goals of this semester. We hoped to meaningfully move the problems involving the RFI sifter and the statistics of volunteer classifications forward.

5. ACKNOWLEDGEMENTS

P.M would like to thank everyone that has supported him on the project during the semester. Special thanks to Prof Bryan Gaensler who both supervised and gave vital feedback to this project! Furthermore this was also supported by many collaborators at CHIME including but not limited to: Adam Dong, Chitrang Patel, Ziggy Pleunis, Mike Walmsley, Shriharsh Tendulkar, Kendrick Smith and Dustin Lang. Thanks to everyone!

REFERENCES

Bassa, C. G., Pleunis, Z., & Hessels, J. W. T. 2016, Enabling pulsar and fast transient searches using coherent dedispersion, arXiv, doi: 10.48550/ARXIV.1607.00909

Bhandari, S., Sadler, E. M., Prochaska, J. X., et al. 2020, The Astrophysical Journal, 895, L37, doi: 10.3847/2041-8213/ab672e

Bochenek, C. D., Ravi, V., Belov, K. V., et al. 2020, Nature, 587, 59, doi: 10.1038/s41586-020-2872-x

Breiman, L. 2001, Machine Learning, 45, 5, doi: 10.1023/a:1010933404324

Chen, T., & Guestrin, C. 2016, in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (ACM), doi: 10.1145/2939672.2939785

CHIME/FRB Collaboration . 2020, Nature, 587, 54, doi: 10.1038/s41586-020-2863-y

Collaboration, T. C. 2019, Nature, 566, 235, doi: 10.1038/s41586-018-0864-x

Condon, J. J., & Ransom, S. M. 2016, Essential Radio Astronomy (Princeton University Press), doi: 10.1515/9781400881161

Cristianini, N., & Ricci, E. 2008, in Encyclopedia of Algorithms (Springer US), 928–932, doi: 10.1007/978-0-387-30162-4_415

FRB Collaboration, Amiri, M., Bandura, K., et al. 2018, doi: 10.48550/ARXIV.1803.11235

FRB Collaboration, Amiri, M., Andersen, B. C., et al. 2021, doi: 10.48550/ARXIV.2106.04352

Gajjar, V., Siemion, A. P. V., Price, D. C., et al. 2018, The Astrophysical Journal, 863, 2, doi: 10.3847/1538-4357/aad005

Lingam, M., & Loeb, A. 2017, The Astrophysical Journal, 837, L23, doi: 10.3847/2041-8213/aa633e

Lorimer, D. R. 2018, Nature Astronomy, 2, 860, doi: 10.1038/s41550-018-0607-9

Lorimer, D. R., Bailes, M., McLaughlin, M. A., Narkevic, D. J., & Crawford, F. 2007, Science, 318, 777, doi: 10.1126/science.1147532

Macquart, J.-P. 2018, Nature Astronomy, 2, 836, doi: 10.1038/s41550-018-0625-7

Mannor, S., Jin, X., Han, J., et al. 2011, in Encyclopedia of Machine Learning (Springer US), 563–564, doi: 10.1007/978-0-387-30164-8_425

Mitchell, T. M. 1980, The Need for Biases in Learning Generalizations, Tech. rep.

Petroff, E., Hessels, J. W. T., & Lorimer, D. R. 2019, The Astronomy and Astrophysics Review, 27, doi: 10.1007/s00159-019-0116-6

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. 1986, Nature, 323, 533, doi: 10.1038/323533a0

Snoek, J., Larochelle, H., & Adams, R. P. 2012, Advances in neural information processing systems, 25

Totani, T. 2013, Publications of the Astronomical Society of Japan, 65, L12, doi: 10.1093/pasj/65.5.l12

Zhang, B. 2020, Nature, 587, 45, doi: 10.1038/s41586-020-2828-1

APPENDIX

5.1. *Implementation of Simulation Technique With Pseudo BONSAI*

This is the details on the implementation of the simulation technique without having to rely on running BONSAI completely offline. This only requires to get a mock version of BONSAI working.

5.1.1. *Getting BONSAI*

First we need to get BONSAI set up locally. Firstly one needs to be on the FRB-Analysis nodes. You need to git pull the following: BONSAI[4]. Then make sure you have GCC for C++11. Then you need to git pull simd_helpers [5], then you need RF_Kernels [6]. You also need python 2.7 [NOTE do not use the latest version of python]. Then you need HDF5 helpers [7]. Next you will have to then go and install each of these. Go into the directory and make install each. Only then, should you install BONSAI. You will most likely need to run so that the system path can find your code.

Once you have done so you can then try using the python2.7 version of the examples pipeline. With it you will need to run a few path commands namely make sure python has the following: python2.7 -m pip install –user –upgrade numpy==1.15.0 and python2.7 -m pip install –user h5py==2.10. Then you can run the unit tests, and fingers crossed nothing blows up![8]

5.1.2. *Reading Real Data into BONSAI*

To interface with real data, from the msgpack file format into memory as a python numpy object we need a package called intensity utils [9]. This package is used to read the data and also apply the correct RFI masks for the data as well. To install it, follow instructions on the readme. You need to export the following and export LANG=en_US.utf-8, this should solve most issues.

If you get an issue with running the example in the readme, Ziggy Pleunis suggested to ignore the example and use the API instead. Follow this notebook [10], this will save you hours of trying to figure it out. Also note, this uses python 3! Which is not compatible with BONSAI python 2.7. So you need to temporarily save the data and then re-read the data in BONSAI. Here is an example script that does that found here [11]

5.1.3. *Manipulating the data*

We decided that the easiest and the most straight forward implementation for this was to just flip the axis for the frequency channels. This would make any thing in the observation non-physical as we know there's exists a frequency dependence for any FRB's should we pick up on any.

5.1.4. *Combining Everything*

We can combine both of these techniques into one old script which I've written and can be found here [12]. Note once again this is written in the ancient and almost forgotten language of python2.7. This gives us the first working version of this comparison trick. WARNING: this does not have the proper detrending method implemented, as the project was cut short by our decision to pivot to a different project direction. For details on the detrending visit the CHIME systems paper where a supposed 2-d cubic spline model was used to detrend and bring it to a mean of 0.

5.2. *Feature Description*

Some feature descriptions were left out as no documentation can be found regarding their descriptions [at the moment]

[4] https://github.com/CHIMEFRB/bonsai
[5] https://github.com/kmsmith137/simd_helpers
[6] https://github.com/kmsmith137/rf_kernels
[7] https://github.com/kmsmith137/sp_hdf5
[8] Shout out to Dustin Lang for all the support on this!
[9] https://github.com/CHIMEFRB/Intensity-Analysis-Utils
[10] https://github.com/CHIMEFRB/Intensity-Analysis-Utils/blob/main/examples/tutorial.ipynb
[11] https://github.com/CHIMEFRB/FRBgen/blob/main/custom_read_write.py
[12] https://github.com/CHIMEFRB/FRBgen/blob/main/bonsai_replica.py

1. 'incoherent_snr' - Incoherent Signal to Noise Ratio (incoherent is when we correlate accounting for geometric time delays)
2. 'max_coherent_snr' - Maximum Coherent Signal to Noise Ratio (coherent is same as incoherent but we also account for clock times/delays) [13]
3. 'max_level1_grade' - Max L1 grade, is a grade given by L1 system which is part of the pipeline that performs per-beam RFI rejection and dedispersion using BONSAI, identifying candidate events in the DM/time plane.(FRB Collaboration et al. 2018).
4. 'mean_level1_grade' - Mean of L1 grade
5. 'snr_weighted_level1_grade' - SNR weighted by the measurement error from L1 system (FRB Collaboration et al. 2018).
6. 'snr_weighted_tree_index_weighted_level1_grade' - SNR weighted by the measurement error and by the tree index from L1 system when performing tree dedispersion (FRB Collaboration et al. 2018).
7. 'std_level1_grade' - standard deviation of L1 grade
8. 'min_tree_index' - minimum tree index from L1 tree dedispersion
9. 'mean_tree_index' - mean tree index
10. 'snr_vs_dm' - signal to noise ratio as a ratio to dm
11. 'group_density' - the number of candidates occupied in a time window
12. 'snr' - signal to noise ratio
13. 'beam_activity'- number of candidates detected in the same time window for beam
14. 'coh_dm_activity' - coherent DM activity
15. 'incoh_dm_activity' - incoherent DM activity
16. 'avg_l1_grade' - average L1 grade

The full list is here *'max_coherent_snr','incoherent_snr', 'max_to_second_snr_ratio', 'max_level1_grade', 'mean_level1_grade', 'snr_weighted_level1_grade', 'snr_weighted_tree_index_weighted_level1_grade', 'std_level1_grade', 'min_tree_index', 'mean_tree_index', 'snr_weighted_tree_index', 'snr_vs_dm', 'std_tree_index', 'ew_extent', 'ns_extent', 'group_density', 'max_snr_ns_beam', 'snr', 'beam_activity', 'coh_dm_activity', 'incoh_dm_activity', 'avg_l1_grade', 'event_no'*

5.3. *Model Parameter Optimization*

Although we eventually chose to implement the hybrid model, we still need to individually optimize the XGBoost models and various other models that we benchmarked. We then had to choose a model parameter search using either a grid search, which brute forces through all permutations of the model parameters, or a Bayesian optimization scheme where, for each evaluated model, we use the performance to update priors that inform which next parameter to then select (Snoek et al. 2012). We initially optimized the random forest model. Our result was that the existing model performs best when taking into account the false positive to false negative rates. The same was done for the XGBoost model

From discussions with multiple collaborators we agree there are two important metrics to consider, (1) we want there to be as few false positive and as few false negatives as possible. And (2) we want there to be relatively the same magnitude for false positives and false negatives. The first confusion matrix 5.3 made by optimized XGBoost has the

[13] https://www.astron.nl/astrowiki/lib/exe/fetch.php?media=ra$_{u}va$: *ra_uva_lecture*10.*pdf*

XGBoost Prediction

		False	**True**	**total**
actual value	**False**	True Neg 35725	False Pos 7	P′
	True	False Neg 134	True Pos-itive 60913	N′
	total	P	N	

RF weight 4 Prediction

		False	**True**	**total**
actual value	**False**	True Neg 35717	False Pos 15	P′
	True	False Neg 75	True Pos-itive 61790	N′
	total	P	N	

Table 10. The RF weighted model on the right shows a great deal of improvement in balancing the error rate in false negatives and false positives than the previous models.

best performance in terms of (1) and the second random forest using a weighted class has the best performance in terms of (2) confusion matrix 10.

To address this issue imbalanced false positives to false negatives, we can weigh the classifications from each XGBoost run. We see in table 10 that the ideal performance at the moment requires a 4 : 1 ratio of negative to positives to achieve the ideal (2) requirement at the sacrifice of more false positives.