Sensitivity Test on FRNN code

Sunny Qin

September 12, 2017

1 Overview

The FRNN code can predict the disruption event 30 ms ahead with very high accuracy (93.5% True Positive Rate and 7.5% False Positive rate). Despite the good output result from the machine algorithm, we understand very little about the physics behind. Which signal is FRNN most sensitive to and what is the physics behind this observation? Furthermore, can we find additional signals or leaving out some current signals to improve the performance of the learning output? This research project is intended to explore possible answers to these problems, and this report will summarize the preliminary result I achieved this summer with the generous help from the FRNN team.

I will first briefly summarize the FRNN machine learning algorithm used since at the beginning of the summer project, I had very little knowledge about machine learning, especially recurrent neural network. I spent a large part of summer research time reading the code, understanding recurrent neural network. This section has no new information but only presents the understanding I have gained.

I will then present the results I have obtained from the sensitivity test, a serious of experiments we did to understand which signal is the most important in the performance of the FRNN algorithm. In the next section, I will present the physics behind disruption and try to relate them with the results observed in the previous section. These understanding mostly comes from the relevant paper and books I read. I am planning on exploring this aspect in detail this fall semester, as the main topic for my junior research.

2 Machine Learning Basics for FRNN code

Why RNN? FRNN code is not the first machine learning code that has been implemented to predict the occurrence of disruptions in JET. One example is a class of supervised machine learning class: the support vector machine method. The SVM method, learning from nine zero-D data from JET, can predict the disruption events up to 90% accuracy. The SVM, a supervised machine learning method, is capable of separating the input vectors (data input with 9 signals) in higher-dimension vector space. However, to incorporate time into the analysis, APODIS code divided the data into several time intervals and trained SVM model for each interval. The RNN, on the other hand, is a new class of neural networks that exploit the sequential nature of the input. Such class of neural networks is designed to predict the occurrence of an element in the sequence that is dependent on the elements that appeared before it. The difference between RNN and the common neural network is that RNN method does not assume that the inputs are independent of each other. From a physics perspective, such independent assumption cannot, in fact, be made for the disruption-predicting problem since the physical state of the tokamak does depend on the previous state. The model we train from the RNN method will predict the occurrence of the disruption event by looking at the input data in a time sequence. Therefore, from a theoretical perspective, the RNN method is a suitable one to solve the problem we face.

Simple v.s. LSTM LSTM stands for Long Short Term Memory, and LSTM RNN is a variation of RNN method that works better than the simple RNN code when solving problems that have some long-term dependencies. LSTM resolves the vanishing gradient problem of the simple RNN by using a more complicated mathematic model in the hidden state. Therefore, implementing LSTM is an optimal choice for our problem.

Normalization Methods The normalization currently implemented in the FRNN code is VarNormalizer: for each signal input, we first get the standard deviation for each machine from a random subset of data input array. Then each data input of that signal from that machine is divided by the standard deviation calculated. In other words, the mean is not subtracted from the data input and therefore, it is not a conventional way of normalizing data for machine learning. The other possible ways of normalization include MeanVarNormalizer, AveragingNormalizer, and MinMaxNormalizer. The MeanVarNormalizer implements the standard normalization method: subtracting the mean and divide by the standard deviation of the input data. The MinMaxNormalizer utilize the range of each signal input from each machine, where range means the difference between maximum value of that signal from that machine and the minimum. For each signal from each machine, the data is subtracted by the minimum and divided by the range. VarNormalizer is chosen because it gives the best testing result but we have not found a explanation for this empirical observation.

Model Specifics and other hyper-parameters If some one-D signals are included in the input data, the FRNN code will add a number of layers of convolutional neural networks, with that number specified by the user. Otherwise, the FRNN code will build a recurrent neural network with a specified number of layers and specified size. For now, the FRNN code has 2 layers of RNN layers and the size of each layer is 200. These parameters are tuned experimentally to reach the best performance of the code. If the number of RNN layers is specified to zero, the FRNN code will build three dense layers of neural network. Finally, there are also other model specifics and hyperparameters: optimizer, drop-out probability, learning rate, etc. They are chosen to optimize the performance of the FRNN code.

3 Sensitivity Test Summary

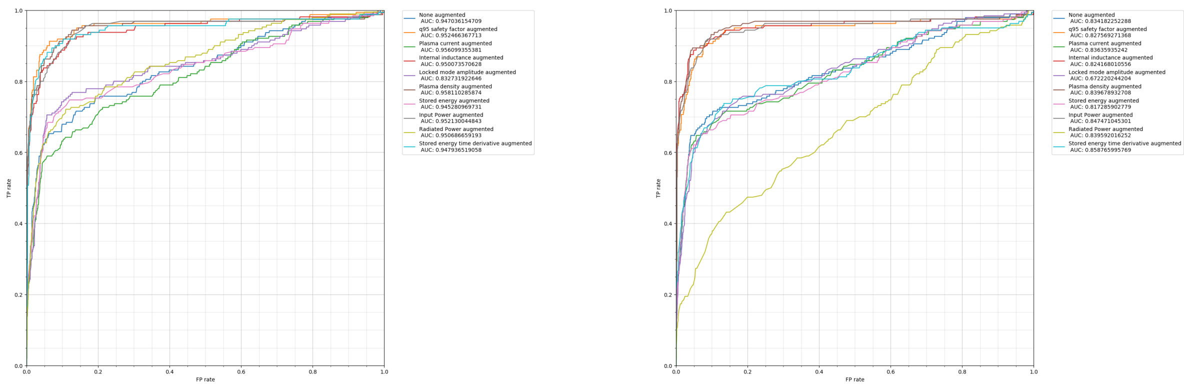
There are two main methods we have attempted to study the sensitivity of the FRNN code. The two methods are the leave-out method and deterministic-augmentation method. In this section, I will present the methodology behind each of them and the results generated. Then I will analyze the results and discuss the current difficulties and problems needed to be resolved. The two methods are:

1 Simply leave one signal out and compare the resulting AUC for each ROC

2 Add noise to one signal - deterministically - for both training and the augmenting process. The noise follows a normal distribution with zero mean and standard deviation equal 100 or to 10 in different trials.

3.1 Leave-out Method

There are nine signals used for the JET data: q95 safety factor, Plasma current, Internal inductance, Locked mode amplitude, Plasma density, Stored energy, Input Power, Radiated Power, Stored energy time derivative. For each set of experiments, I excluded one of them (or none of them) by specifying "specific signals" in conf.yaml. Then I produced the ROC plot for both the training and the testing processes, shown below:



(a) Test ROC

(b) Train ROC

Figure 1: ROC Plots for the Leave-out Method

In the next step, I turned off the random.seed in mpi_run.py and ran the code multiple times to get the uncertainty of the experiment. After repeating the experiment 10 times, I was able to obtain the mean and the error for AUC for each signal (signal = 'Plasma density' means Plasma density excluded). The graph below shows a summary of results, sorted by the test AUC value for each signal with the numerical results after it.

Left-out Signal	Test AUC	Test std	Train AUC	Train std
Locked mode amplitude	0.834	0.013	0.668	0.016
Stored energy	0.933	0.015	0.798	0.015
Stored energy time derivative	0.946	0.004	0.859	0.008
Plasma current	0.949	0.007	0.829	0.010
Internal inductance	0.951	0.009	0.834	0.015
None	0.951	0.005	0.832	0.007
Radiated Power	0.954	0.004	0.832	0.006
q95 safety factor	0.955	0.003	0.838	0.009
Input Power	0.956	0.005	0.8428	0.007
Plasma density	0.962	0.003	0.839	0.013

Table 1: AUC Summary for Leave-out Method

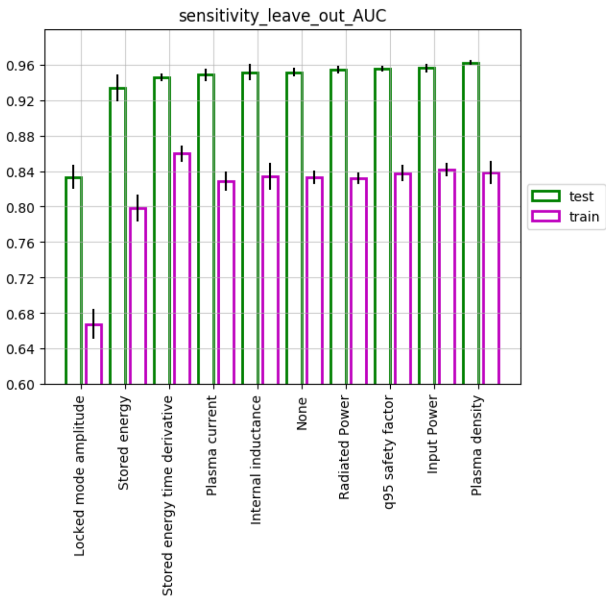


Figure 2: AUC Plot for Leave-one-out method

3.2 Augmentation Method

The augmentation process serves a similar purpose as leaving the signal out: to ask the FRNN to build the model from data input but with one of the signals invalid. The leave-one-out method achieves this purpose by simply deleting the signal whereas the augmentation method achieves this purpose by making this signal follows a random distribution and thus produce no useful information. The augmentation is done in AugmentingVarNormalizer, a class inherent from VarNormalizer and uses the same normalization method as VarNormalizer. Therefore, the data is augmented during the pre-processing period, which means both the training and the inference processes use the augmented signal. Each signal is normalized to standard deviation $\sigma = 1$ after the applying VarNormalizer. We first attempt to set $\sigma = 10$. The resulting ROC curves, however, do not differ too much from each other, meaning the signal did not become random enough after the augmentation. To resolve this issue, we decided to change σ

to 100. We have completed few trials for this method, and the summary AUC plot is included below.

Left-out Signal	Test AUC	Train AUC
Locked mode amplitude	0.375	0.399
Plasma current	0.674	0.574
q95 safety factor	0.920	0.850
Stored energy time derivative	0.926	0.862
Plasma density	0.936	0.855
None	0.938	0.869
Radiated Power	0.945	0.846
Stored energy	0.954	0.837
Input Power	0.958	0.848
Internal inductance	0.960	0.850

Table 2: Summary for Augmentation Method

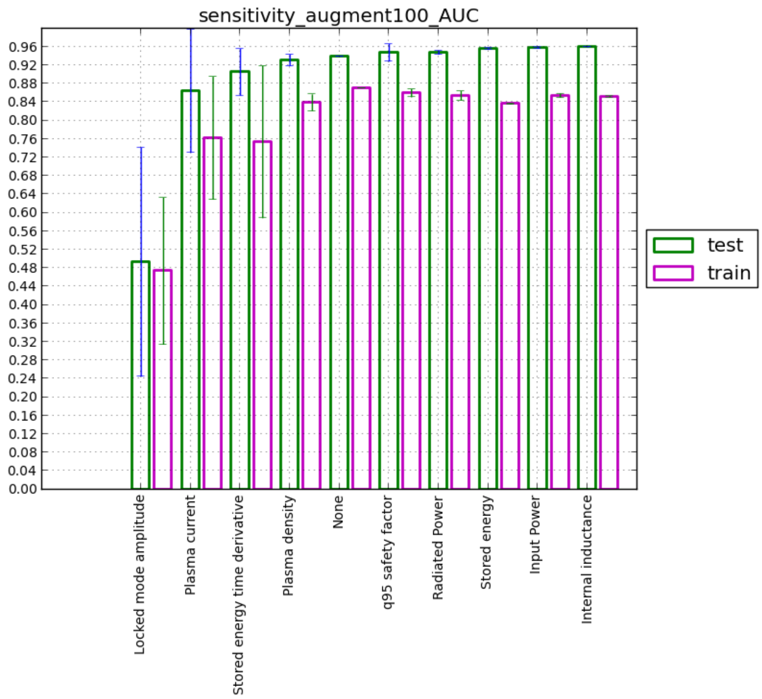


Figure 3: AUC Plot for Augmentation method

However, we noticed that the resulting ROC with a $\sigma = 100$ augmentation has many repetitive data points, which might be caused by the way we augment. The problem of augmenting with a fixed standard deviation is that the augmentation process now relies on the normalization method. In our case, VarNormalizer calculates the standard deviation of each signal from a subset of the data inputs, instead of the complete data set. By using a randomly chosen subset, the standard deviation generated might vary from each trial and would have a different effect on the augmentation process. To resolve such issue, we also attempted to modify the VarNormalier such that it can calculate

Left-out Signal	Test AUC	Test std	Train AUC	Train std
q95 safety factor	0.122	0.0003	0.248	0.0053
Input Power	0.507	0.0043	0.296	0.0017
Locked mode amplitude	0.811	0.0040	0.610	0.0053
None	0.943	0.0003	0.871	0.0005
Stored energy time derivative	0.946	0.0002	0.879	0.0002
Radiated Power	0.950	0.0022	0.850	0.0010
Plasma density	0.952	0.0001	0.870	0.0002
Stored energy	0.956	0.0010	0.838	0.0007
Internal inductance	0.961	0.0000	0.857	0.0001
Plasma current	0.963	0.0000	0.869	0.0005

Table 3: AUC summary for Augmentation method 2

the standard deviation from the entire subset and thus apply the normalization deterministically. The output ROC shows some improvement, but we still need to work more on the augmentation methodology. The following AUC summary is from this set of experiment.

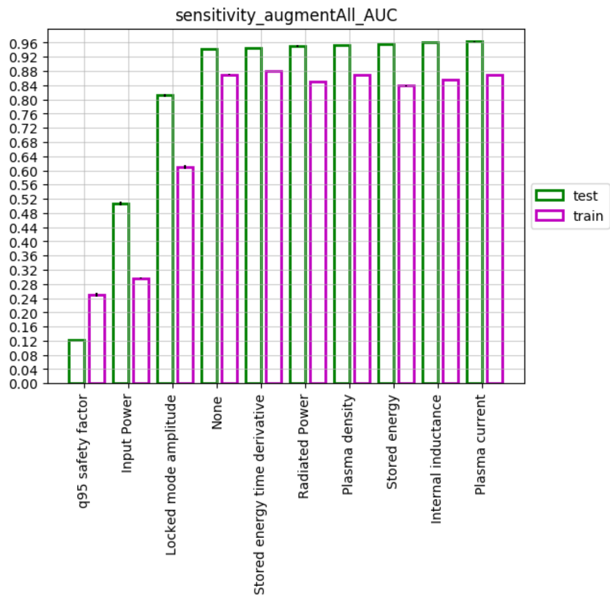


Figure 4: AUC Plot for Modified Augmentation method

3.3 Result Analysis

In this section, I will analyze the three set of experiments individually then compare with them to find similarities and differences between them. Then, I will discuss the possible improvements for future experiments.

Analyzing the result from the leave-out method, we can find the 5 signals that FRNN is most sensitive to; they are locked mode amplitude, stored energy, stored energy time derivative, plasma current, and internal inductance. Among them, locked mode amplitude is the most important signal in predicting the disruption event: without this signal, the test AUC is only 0.834.

Analyzing the result from augmentation method with the VarNormalizer (the original one used in the FRNN), we can find the 5 signals that FRNN is most sensitive to; they are locked mode amplitude, plasma current, q95 safety factor, stored energy time derivative and plasma density. Again, we see locked mode amplitude being the most important signal in the FRNN model and this time, the test AUC dropped to 0.375 with locked mode amplitude augmented. The second important signal according to the test is plasma current: with plasma current augmented, the test AUC is only 0.674. This test results agree with the experiment done by Kyle Felker: he concluded that locked mode and plasma current signals can predict 68% of disruptions. Both methods also agree that stored energy time derivative is also an important signal in FRNN.

Analyzing the result from the augmentation method with the modified VarNomarlizer, we see that locked mode amplitude still plays an important role, this observation comes from three sets of experiments. Besides that, we also notice both augmentation methods show that the FRNN is very sensitive to q_{95} safety factor. The second augmentation also allows us to conclude that input power is more important than locked mode amplitude, and such conclusion contradicts with the results from the first two methods.

The conclusion so far is still preliminary in two aspects: first, we still need to work on the methodology for the sensitivity test and draw a general conclusion from them. For example, the two augmentation test produces ROC curves with many repetitive data points, as a result of which, the AUC we are looking at might lack some accuracy. One step we have taken to resolve this issue is to use the modified VarNormalizer. However, since the FRNN is trained to be optimal with the original VarNormalizer, the model trained with the modified VarNormalizer has an inferior performance compared to the original one. To move forward, we need to find out the reason why there are many repetitive data points for ROC curve after the augmentation process. We are hoping to produce better testing results after resolving the underlying issue. Then, we can repeat the augmentation method several times to produce a comprehensive result.

Another improvement is to separate the augmentation process for train and test so that we can train with one signal augmented randomly and test with one signal augmented deterministically. This method is proposed by Julian and is supposed to test the sensitivity of each signal. However, this might require a larger change in the FRNN code compared to the first two methods.
Besides the possible improvements on the testing methodology, the conclusion so far is preliminary also from another aspect: we do not understand the physics behind these observations: locked mode amplitude is one of the most relevant signals in the FRNN code, agreed by all the sensitivity tests above. It then becomes essential for us to understand why this is the case. The next section will include some research I have done over the summer to understand the

physics behind it. But the main part of this aspect will be explored during the semester, as my junior research topic.

4 Physics basics for disruption prediction

4.1 Stable Operation of Tokamaks

For the tokamak to achieve a stable operation state, some well-known parameters have to be kept under certain limits: plasma current and plasma density. Plasma current should be limited by the safety factor q_ψ and plasma density should be kept under Greenwald density limit: $n_{GW}(10^{20}m^{-3}) = I(MA)/a^2(m)$. Plasma pressure should also be kept under Troyon normalized beta parameter, but we do not have such data available. If any of these parameters reach the limit, the system is very likely to become unstable and eventually leads to a major disruption. In fact, data analysis has shown that a low q_{95}, high density usually indicate a higher disruption frequency in the tokamak. Thus, we can expect plasma density and q_{95} are two relevant signals for prediction. They are relevant because they set the stable limit for the tokamak.

4.2 Mechanism for disruption

From a theoretical perspective, the main cause for the disruption events tokamak lies on the perturbation on the tokamak configuration. The known helical perturbation is described by poloidal and toroidal angels θ and ϕ and their mode numbers m and n. A perturbation in the form

$$\xi(r,t) = \xi(r) exp\, i(\omega t + m\theta - n\phi)$$

will cause a rapid loss of thermal energy confinement and thus cause the major disruption event. Therefore, we should expect that there is a distinct thermal energy change before the disruption event.

The major disruption usually consists of three phases in timely order: slow thermal quench, fast thermal quench, and current quench. Theoretical analysis shows that the slow thermal quench, the very initial state of the disruption event, is very likely to be the resulting event of the initiation of a locked helical mode.The transition from slow to fast thermal quench is also believed to be caused by the locked mode development. This theoretical analysis can explain what we observed in the last section: locked mode is one of the most important signals in predicting the major disruption.

From an experimental perspective, many experiments have confirmed that $T_e(r)$ is observed to show a flattering in the center before the major disruption. Even such experiments studied thermal energy in 2-D form, but we should still expect the temperature parameter plays a role in predicting the major disruption events.

Other experimental studies also concluded that the weakening of the magnet shear in the chore is often a key indicator of the upcoming disruption event. More importantly, the reasons for this precursor event can be caused by the locked mode development. impurity accumulation, the neoclassical tearing mode can also be the cause for the weakening of the shear. This experimental observation could potentially be used to explain the result of our sensitivity test.

4.3 Signal Importance

From the discussion above, we can see that locked mode, plasma current, plasma density and q_{95} safety factor are the most relevant parameters for the onset of the major disruption event. Such conclusion drawn from a first principle physics perspective does agree with our experiment to a limited degree. Locked mode amplitude is observed to be a very relevant signal for the FRNN is now supported by the theoretical analysis. The leave-out method and the first augmentation method testify the importance of plasma current and the second augmentation shows the importance of q_{95} safety factor.

5 Conclusion and moving forward

There are two aspects of the sensitivity of FRNN project: first to find out which signals FRNN are most sensitive to and the second is to understand the physics behind these observations. For the first part of the project, we have implemented two main sensitivity test methods and produced some results from them. For the second part, I have done some readings on my own and attempted to relate it to the observations. We first conclude that locked mode amplitude is one of the most important signals in the FRNN. My research on the physics behind the tokamak disruption event supports this conclusion: The onset of the thermal quench is caused by the locked mode development. We also concluded from the leave-out and the first augmentation method that plasma current also contributes positively to the performance of the FRNN code. From the physics perspective, plasma current relevant because it is a limit parameter for the stability of the tokamak. Looking at the output individually, we also conclude the importance of q_{95} safety factor and plasma density, and these two observations are both supported by observations from physics experiments.

Both perspectives have a lot of room for improvement and I will use the opportunity of my junior research to work more on this project. The possible improvements and future objectives are lists below:

Testing method
1. Resolve problems associated with the augmentation method and produce final results for this method.
2. Implement the non-deterministic method and produce results from it.
3. Compare the similarities and differences between these methods and understand the underlying causes.

Physics basics
1. Relate the observation with the physic from a more rigorous approach.
2. Add temperature to the FRNN and analyze the performance of the new FRNN. Study the physics behind it.

6 References

1 ITER Physics Basic Editors, ITER Physics Expert Group Chairs and CoChairs, ITER Joint General Team and Physics Integration Unit, ITER Physics Basics, Chapter 1, Overview and Summarty (1999), 27-39

2 ITER Physics Basic Editors, ITER Physics Expert Group Chairs and CoChairs, ITER Joint General Team and Physics Integration Unit, ITER Physics Basics, Chapter 3, MHD stability, operational limits and disruptions (1999), 2253-2367

3 Bill Tang, Advances in Big Data Machine Learning for Fusion Energy (presentation in Xi'an China, 2016)

4 Antonio Gulli; Sujit Pal, Deep Learning with Keras, 2017