This document describe How we develop code and model for Detecting Continues Gravitational Wave (CGW) from Two signal in which one signal was Noise and other was Noise+Continues Gravitational Wave(CGW). We used signal frequency 100Hz as F0 and Band 1Hz for noise and 100Hz as F0 in CGW. We used detector H1,L1,V1 to generate CGW We used singal/noise data for training and testing from PyFstat (<https://github.com/PyFstat>)

Kaggle challange reference https://www.kaggle.com/competitions/g2net-detecting-continuous-gravitational-waves/overview

We used 1 year data for CGW+Noise and Noise generated from PyFstat. There were 17520 amplitude for per frequency (1800 different frequency within 1 Hz band for freqency mid around 100Hz). This 17520 ampilitude obtain from Fourier tranform. And 17520 amplitude used were complex number. We took complex number and then applied np.absolute on it. Then applied 10 \* np.log() on both Noise and CGW+Noise data.

We plot graph with X axis as Frequency and Y as Tranform data of CGW+Noise amplitude. We did same for Noise data. Graph was clearily showed Spike in mid frequency in CGW+Noise Data while Noise data was plan(no spike at all)

We Found frequency where applitude had spike (we took max value of amplitude at every frequency and compare with >-210 ) and we filtered that frequency and put it as additional feature where we put classifier values as 1 while at same frequency we 0 as classifier value for Noise. Same for Noise.

We added that frequency ( 114 frequecy) as additional feature to 17520 column which had value 1 or 0. For data which has not any of 114 frequency we put -1 for that frequency which not present in Noise Data and CGW+Noise Data ( I mean among 114 frequency where applitude is greater than -210 in CGW+Noise/Noise data are not present (after extracting max values greater -210) in test data of CGW+Noise are marked 0 and if same frequency are present in Noise data then it will 1 in noise else if not present in both (CGW+Noise and Noise data) test data as whole it will -1 for both

Then after computing 114 Frequency as feature/column in data we got 17634 column of data where 17520 was complex data transform with np.absolute. These 17634 feature trained with Perceptron Model and SVM Model.

We generated test data with same method as train. We tested many times test data with Perceptron and We got 100% result with 0 misclassified data