Finding Pulsars
- Final Project of Pattern Recognition Course

Ziyuan Feng

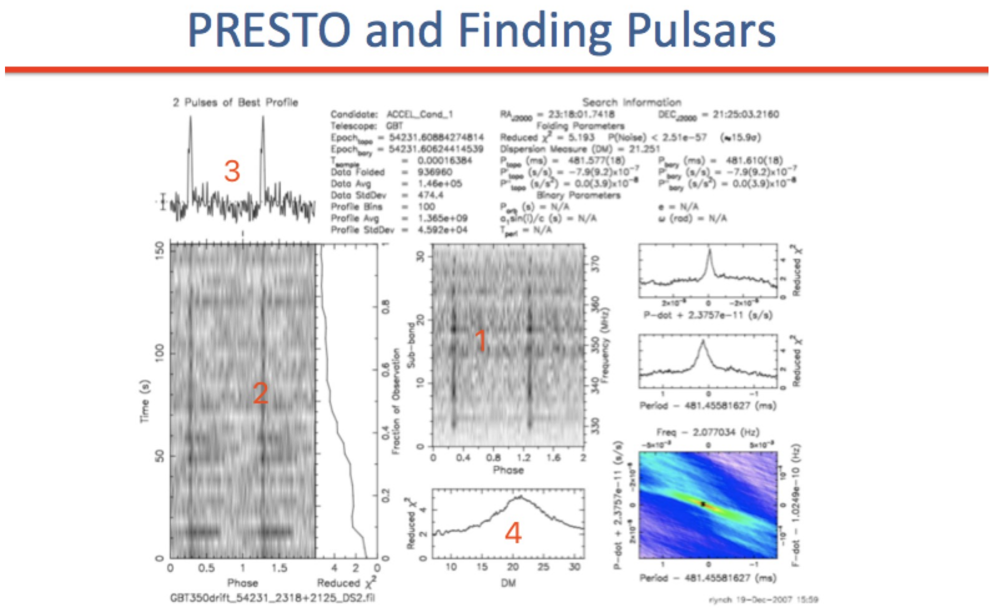
January 7, 2018

1 Abstract

In this project, we are going to explore the pattern of pulsars which differentiates them from the non-pulsars in a large dataset. Statistical methods are used to illustrate the overall features of each data sample, followed by typical machine learning techniques which fit the pattern with powerful models, and the deep learning method as a convenient tool to deal with complicated features. This model achieves satisfying performance in the test set.

2 Problem Analysis

There are three different-source pulsar datasets: `HTRU1`, `p309`, and `PMPS`. Since HTRU1 is highly imbalanced (which has far more negative samples then the positive ones), this experiment only explore the data from `p309` and `PMPS` (both of which are in "PFD" data format). `p309p` contains 2698 pulsar samples and 1655 non-pulsar samples. `PMPS` contains 1001 pulsar samples and 1001 non-pulsar samples.

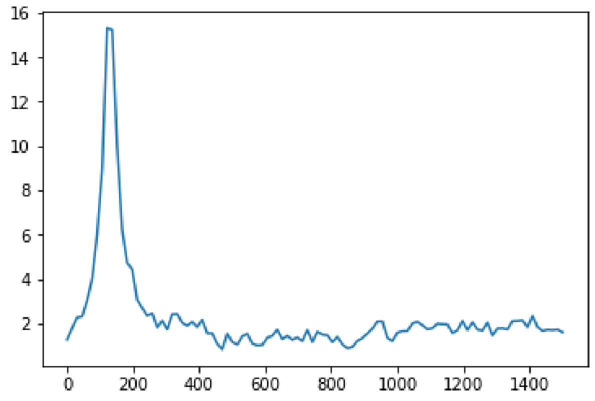

PRESTO and Finding Pulsars

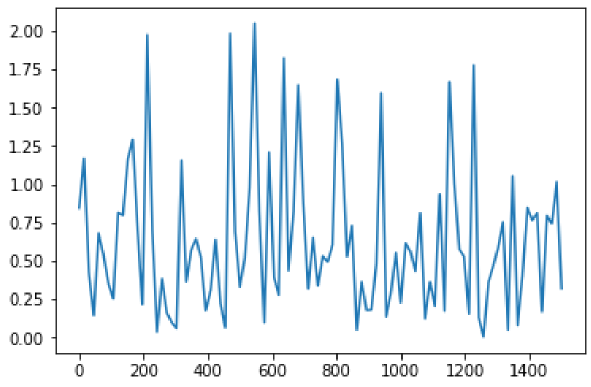
The basic features in these datasets are shown by the graph above.

- *profiles* are peaks of the magnetic field figure, presented by a list of float numbers (length not fixed). (sub-graph 3)
- *the calculated reduced-chi$\hat{2}$* of the current summed profiles, presented by a float number.
- *the interval-summed profiles versus subband*, presented by a 64×n matrix (denoted as *sub-bands*) and a 32×n matrix (denoted as *sub-ints*). (n not fixed) (sub-graph 1 & 2)
- *two arrays showing the reduced-chi$\hat{2}$ versus DM*, one is a counter sequence, the other shows the value. (sub-graph 4)

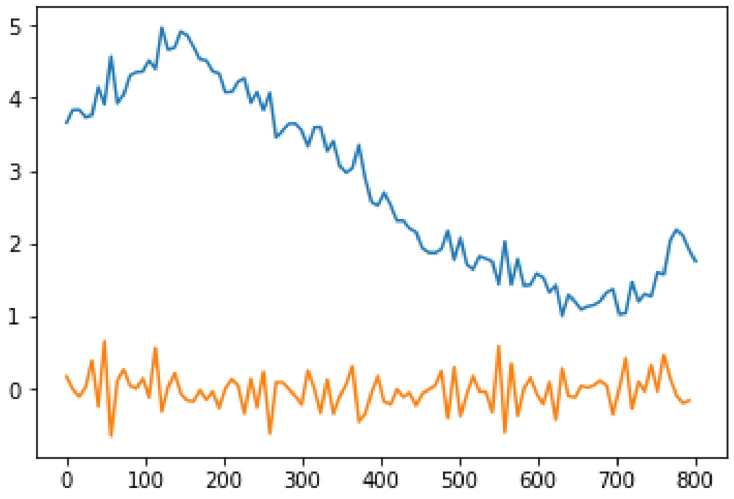
It is possible to plot some relevant features to observe the difference between pulsars and non-pulsars.

These are two main patterns of *the reduced-chi$\hat{2}$ versus DM*. One fluctuates intensively while the other has only one peak.

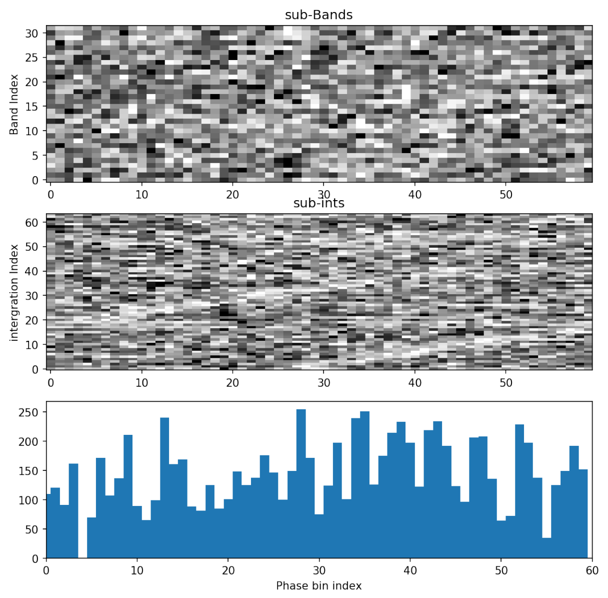


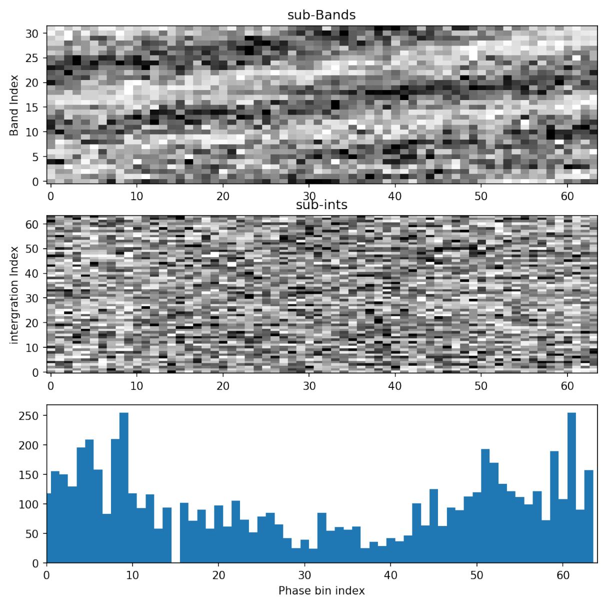


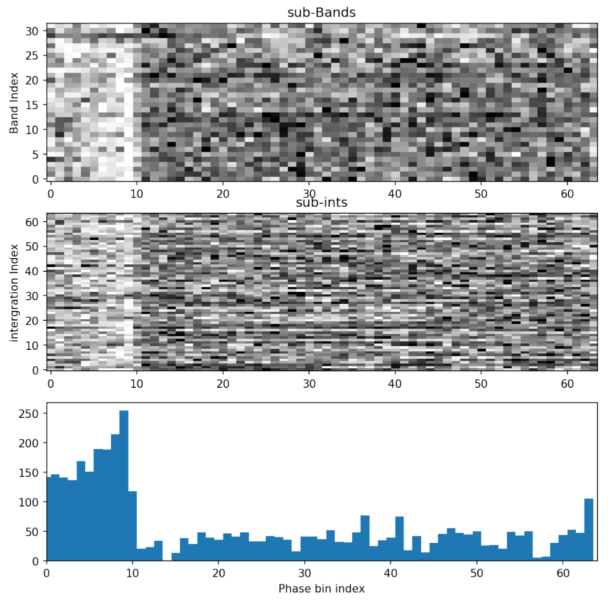
This figure shows a curve of *the reduced-chi$\hat{2}$ versus DM*(Orange) and its first-order difference (Blue).

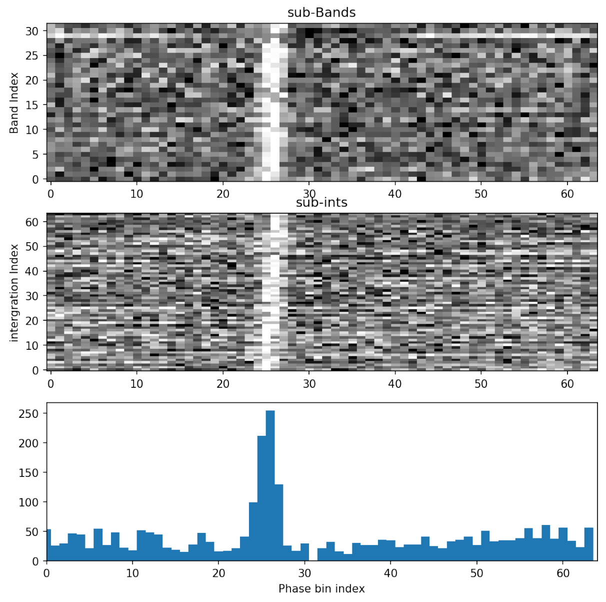


The *sub-bands*, *sub-ints* together with *profiles*.









As we can see in the figures above, compared to a non-pulsar, a pulsar may have:

- one high peak in its *the reduced-chi$\hat{2}$ versus DM* curve.
- one or more white bars in its *sub-bands* and *sub-ints* figures.
- peak profiles in the same position as the white bars.

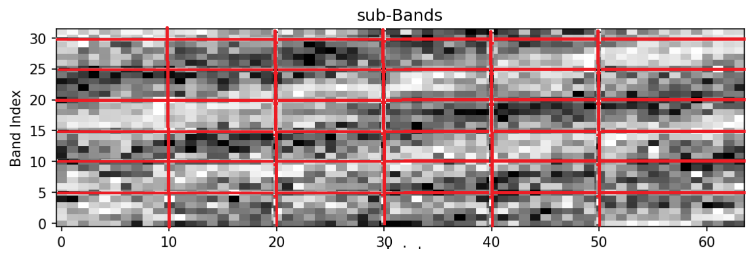
The observations give us some hints about the pattern of a pulsar.

3 Feature Extraction

Directly using the basic features above to build a model is a little bit impossible, because there are too few features for a machine learning problem and some 2-D or sequence-like features such as the sub-bands and DM curves are difficult to handle as ordinary features. How to extract more features based on the basic infomation is a tough task. And this can be done by creating statistical indicators.

1. Three features illustrate infomation about *sub-ints*.
   - the second dimension of *sub-ints*.
   - the average (scalar) of *sub-ints*.
   - the median (scalar) of *sub-ints*.
2. Two features illustrate infomation about the profile.
   - the number of profile - the length of the profile list.
   - the average of profile - the average of all peak values.
3. Features illustrate the *reduced-chi$\hat{2}$ versus DM curve* and how it fluctuates.
   - the variance of *reduced-chi$\hat{2}$*.
   - the maximum of the first-order difference of *reduced-chi$\hat{2}$*.
   - the minimum of the first-order difference.
   - the average of the first-order difference.
   - the variance of the first-order difference.
   - the number of *reduced-chi$\hat{2}$* whose absolute value grater than 0.5.
4. Others
   - *the calculated reduced-chi$\hat{2}$*, the basic feature.

With these 12 features, it is possible to build a primary model to recognize the pattern of a pulsar. However, there is an important information that this model is not involved: the plot figure of *sub-bands* is actually noisy or well-structured, which shall be used by the model. Therefore, it is possible to slice the whole image into several regions like this.



The mean of all pixels in a region is calculated as a representative of this region. Therefore there are 48 mean numbers in total (Notice: dispose of the top small pieces), which are 48 features added into the 12 features to become a feature vector of length 60.

Before deep learning methods are used, these features help us build a machine learning model to find pulsars.

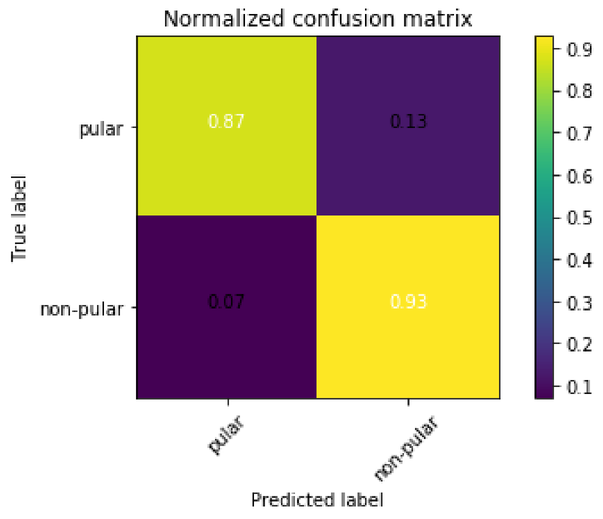
4 Classification & Evaluation

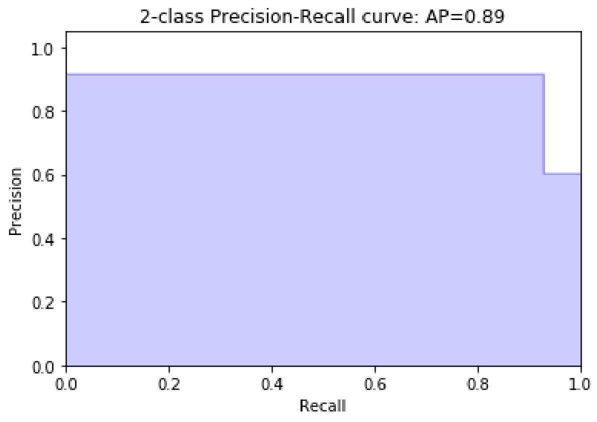
4.1 Bagging

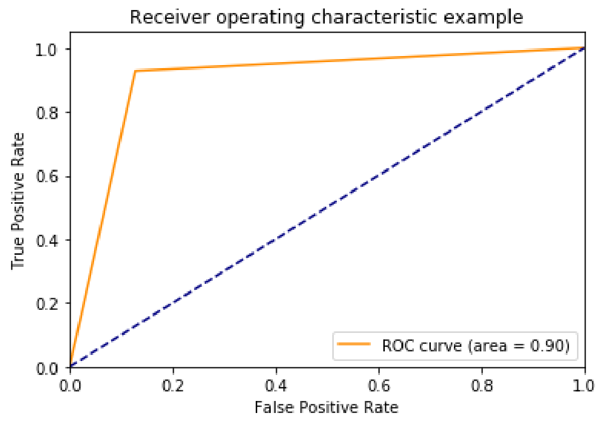
Bagging is a powerful ensemble learning model by combining the results from different classifiers to get a overall result.

A classical bagging model is the **Random Forest**. In this case, 20 estimators (trees) are used to build a random forest. The quality of a tree split is measured by Gini impurity. Samples are boostraped. And the class 1 (pulsars) has higher weight than class 0 (non-pulsars).

The model is trained and test within `p309` dataset. The test results are shown below.







However, the model trained in `p309` dataset does not perform well in `PMPS` dataset. But if the model is trained in a combined dataset of `p309` and `PMPS`, the result is nearly the same as what is shown above.

4.2 Boosting

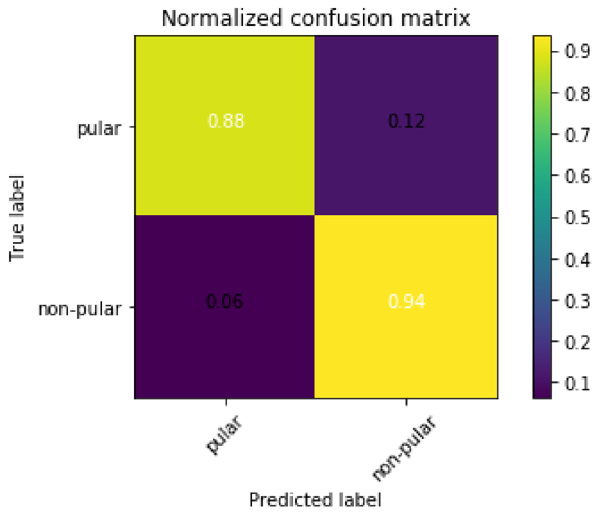
Boosting is also a common ensemble model. In this project, the **gradient boosting decision trees** and **AdaBoost** are used. They both have a good performance.

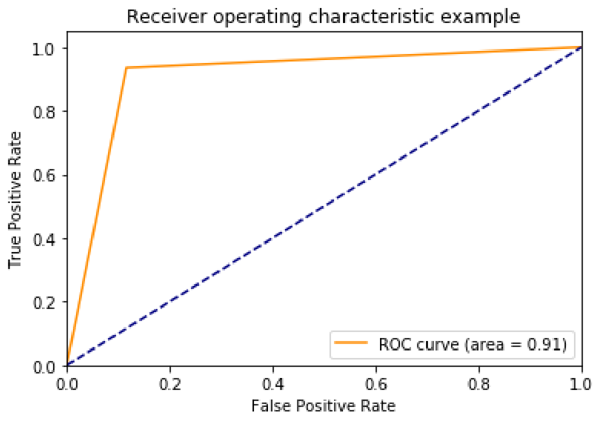
For gradient boosting decision trees, logistic regression for classification with probabilistic

	Random Forest	AdaBoost	GBDT
Accuracy	0.90094	0.91504	0.8907
Precision	0.9144	0.9264	0.9118
Recall	0.9168	0.93348	0.9008
F1	0.9156	0.92994	0.9062
ROC area	0.90	0.91	0.91

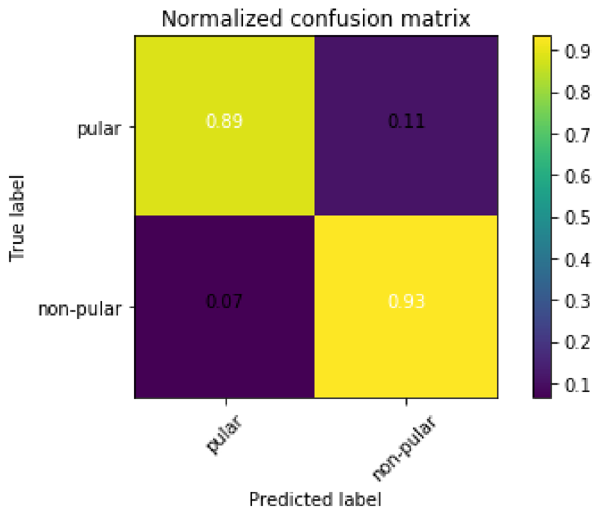
Table 1: Ensemble Classifiers

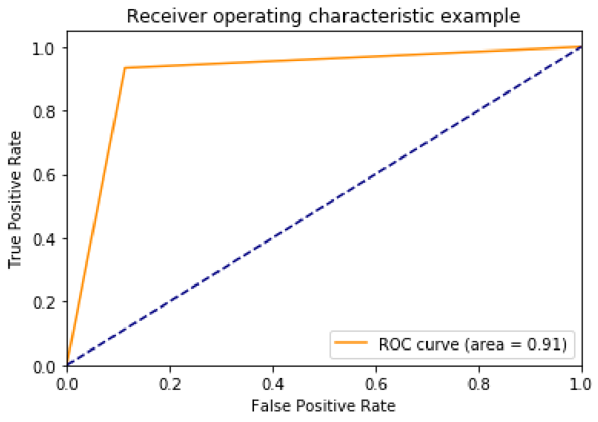
outputs acts as a loss function. Gradient boosting is fairly robust to over-fitting so the model use 200 estimators for boosting with learning rate 0.1. The model is trained and tested in `p309` and `PMPS`.





For Adaboost, the maximum number of estimators is set to be 200 with SAMME algorithm. The base estimator is a 3-depth decison tree. The model is trained and tested in `p309` and `PMPS`.



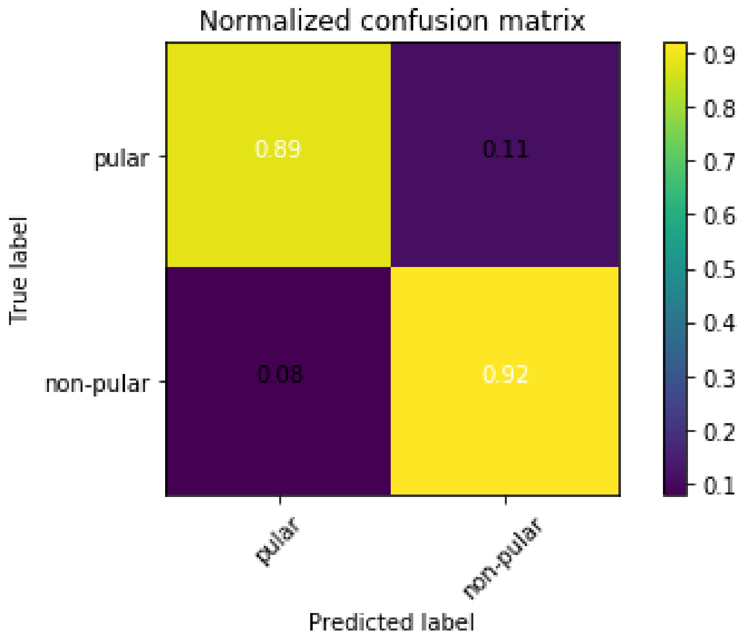


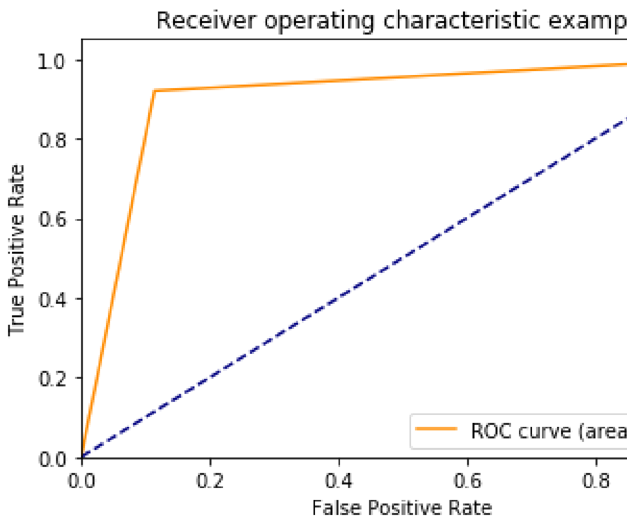
4.3 Convolutional Neural Network

The last method is the deep learning approach to deal with the *sub-ints* and *sub-bands* images. By linear interpolation along rows, *sub-ints* can be presented as a 64×64 matrix and *sub-bands* as a 32×64 matrix. Hence, the architechture of the Convolutional neural network is as follows.

Layer	Tensor Shape
input	[batch_size, height=32, width=64, channels=1]
conv1	filter:[height=3, width=3, in_channels=1, out_channels=5]
pooling	size:[1, 3, 3, 1] stride: [1, 2, 2, 1]
conv2	filter:[height=3, width=3, in_channels=5, out_channels=15]
pooling	size:[1, 3, 3, 1] stride: [1, 2, 2, 1]
conv3	filter:[height=3, width=3, in_channels=15, out_channels=30]
pooling	size:[1, 3, 3, 1] stride: [1, 2, 2, 1]
conv4	filter:[height=1, width=1, in_channels=30, out_channels=60]
pooling	size:[1, 3, 3, 1] stride: [1, 2, 2, 1]
full_connect	weight:[480, 60] bias:[60]
output	weight:[60, 1] bias:[1]

The input is the 32×64 image pixels and the output is a scalar - 0 or 1. The network is implemented using TensorFlow with batch normalization and drop-out technique in order to prevent overfitting. The hyper-paramters are determined by experience and instinct. Maybe there is a better parameter combination that has not been tried in this experiment. By adding the convolutonal neural network into the codes before, the model is a combination of a CNN and a random forest, with *sub-bands* feature only used in CNN. The model achieves satisfying performance. The combined model is trained and tested in `p309` and `PMPS`.





The results are similar when combining CNN with other clssifiers.

	AB+CNN	GBDT+CNN	RF+CNN
Accuracy	0.9270	0.8952	0.9056
Precision	0.9221	0.9348	0.9195
Recall	0.9367	0.9120	0.9195
F1	0.9298	0.9135	0.9195
ROC area	0.91	0.91	0.90

Table 2: CNN + other classifier