Exoplanet Detection Task IITP Report

Srivatsa RJ **P Kulkarni**
Sophomore year
IIT-Patna

A.Raj **A.Drolia**
Junior Year
IIT-Patna

Abstract

The following report is prepared for the proceedings of Exoplanet Detection task of the 6th InterIIT TechMeet - 2018.

This report and the relevant code will be available at `https://github.com/rjs211/Exoplanet-IITP`.

1 Introduction

NASA′s Kepler mission to search for extra solar planets uses Transit photometry, which detects the transit of a planet in front of a star as transient drops in stellar intensity , but spurious intensity dips and other noise in the data due to non-planetary stellar variability has led to high false-positive rates for detecting transits. This Report provides a novel ensemble solution to the problem, which uses RNN,CNN and SVMs to reduce the error rate in Exoplanet candidate identification.

2 Dataset

The training Data provided by the organizing committee had 3960 samples with labels, each with 3197 timestamps, denoting the observed intensity at periodic intervals of time. The dataset was highly skewed with only 33 positive samples, which makes the task of preventing over-fitting extremely difficult, while trying to reduce the false positive rate. The Test Data contained 2000 Samples without label for which the correct class labels have to be predicted by the model.

3 Approach and Pre-Processing

The given data is a time series and the actual value of the flux may have a meaning. The task is to predict the existence of exoplanet in the observed system. The existence is predicted using photometry by employing the fact that the periodicity of dips in intensity is caused by the exoplanets blocking the light between the observer and the light source the exoplanet revolves around , causing periodic dips. There may be relative motion between out solar system and the observed system and which may be the cause of trend in the observed values over a period of time. To tackle this the samples were De-trended by using Median Filter. The data may have noise indistinguishable from the dips but the local maximas can be clipped and thus clipping was done .Since the model must work for all samples, each sample was normalized/scaled.This pre-processed data is oversampled and used for training ANN model (subsection 4.1). As mentioned earlier we must predict the periodicity of the dips and hence, converting the time domain to frequency domain becomes crucial and for that purpose, Fourier Transforms were used (Bloomfield, 2004). SMOTE sampling(Chawla et al., 2002) was used on the Fourier Transformed Data for training SVMs as described in Subsection 4.2. The ANN[1] and SVM [2] models were combined by using an MLP based ensemble approach(subsection 4.3).

4 Model Architecture Description

We developed a Multi-Layer Perceptron (MLP) based ensemble approach which learns on top of Two Models . We separately train and tune all the models and then feed the prediction scores of each model as input to an MLP for ensembling (in Subsection 4.3). Training and tuning of this system is performed separately. The resultant pipeline is used for the Final Prediction.

4.1 Model 1: ANN (RNN+CNN)

As mentioned in section 3 , The actual values of the flux may have a hidden semantic meaning. Recurrent Neural Networks(RNN) have known to unravel semantic meaning (Dieng et al., 2016) and

[1] Built using Tensorflow (Abadi et al., 2015)
[2] Built using Scikit-Learn (Pedregosa et al., 2011)

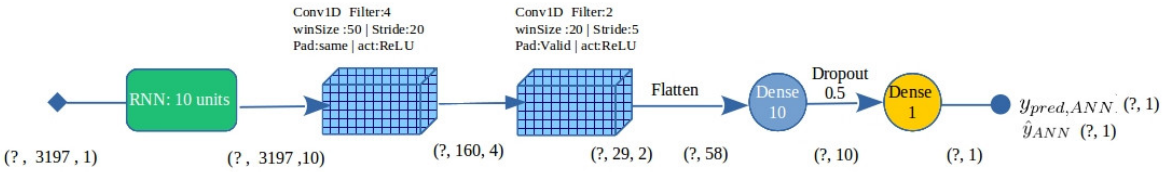

Figure 1: ANN Model.

Name	Value/units/hyperparameters
RNN	units : 10
1-D Conv (I)	filters:4 | kernel:50 | strides: 20 | pad:Same | act:ReLU
1-D Conv (II)	filters:2 | kernel:20 | strides: 5 | pad:Valid | act:ReLU
Fully Conn.(I)	units : 10
Dropout	keep_prob : 0.5
Fully Conn.(II)	units : 1

Table 1: ANN Hyperparameters

hence , we use Dynamic RNNs with BasicRNN-Cell. We can process a sequence of vectors x by applying a recurrence formula at every time step t:

$$h_t = f_W(h_{t-1}, x_t) \tag{1}$$

Where t denotes the current timestamp , h_t the current state, f_w some function with parameter W , h_{t-1} the previous state and x_t the current input vector.

or more accurately

$$h_t = tanh(W_{hh}h_{t-1} + W_{xh}x_t) \tag{2}$$

and the output sequence,

$$y_t = W_{hy}h_t \tag{3}$$

The output sequence for each timestamp is collected and is interpreted to be an encoded form of the flux at every observation.

The sequence is passed into two 1-D Convolution layer with ReLU activation and different parameters.Convolution layers, which are extensively used for pattern recognition (Karpathy, 2015) convolves filter along the time axis for extracting time patterns.No Max-Pooling was applied as the dips are more important and max-pooling may lead to loss of valuable information. The flattened output of the convolution layers is passed to a Fully Connected (Dense) layer ,followed by to a dropout layer and finally to the output layer (Sigmoid Activation).The regularization effect of dropout layer (Srivastava et al., 2014) also helps in preventing overfitting. The output of the final layer when sigmoid is used, can be interpreted as the probability ($\hat{y}_{ANN}$) of the sample being True. The probability is rounded off to either 0 or 1 ($y_{pred,ANN}$).

Weighted cross entropy, a special case of binary cross entropy is used as a partial solution the class imbalance problem.Adam Optimizer was used to minimize the loss function. The Details of various hyperparamters used are listed in Table 1 .

4.2 Model 2: SVM

An addition to the maximal margin Classifier, Support Vector Machines (SVM) work on the idea of the Kernel Trick, which is to project linearly inseparable data to a higher dimension (N) to find a $N-1$ dimensional hyperplane to find the decision boundary. Various kernels employ various paradigms for this transformation and classification. The data used here is the Fourier Transformed data, from which intiuitively, one must be able to conclude on periodicity. So in order to detect and classify based on dips at small intervals of time, Linear kernel is used to predict the labels ($y_{pred,SVM}$).

For tackling the class imbalance problem,Synthetic Minority Over-sampling Technique or SMOTE is used. SMOTE intelligently over-samples the data by the use of K-nearest neighbors and bootstrapping i.e. random sampling with replacement.

4.3 Ensemble Model

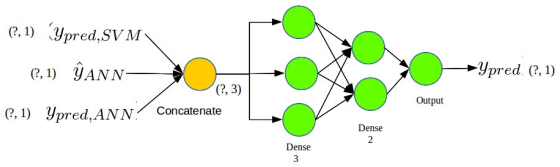

Figure 2: Ensemble Model Model.

Ensemble of various systems is an effective technique to improve the overall performance, by one model fulfilling the defects of the other. Ensembling is known to reduce the generalization error, which in turn reduces over-fitting (Ghosal et al., 2017). The input to the Ensemble model comprises of:

1. SVM Prediction : Binary ($y_{pred,SVM}$)
2. ANN Probability : ($\hat{y}_{ANN}$) $0 \leq \hat{y}_{ANN} \leq 1$
3. ANN Prediction : Binary ($y_{pred,ANN}$)

For every sample,the above features were concatenated and fed as input to two fully connected layers and with no activation function and different units and finally into the output layer (sigmoid activation). The rounded Sigmoid output of the Ensemble model is considered as the Final Prediction (y_{pred}). The Details of various hyperparamters used are listed in Table 2 .

Name	No. of Units
Fully Conn.(I)	units : 3
Fully Conn.(II)	units : 2
Fully Conn.(II)	units : 1

Table 2: Ensembler Hyperparameter

5 Training and Validation

The given dataset contains 3960 samples. Since no separate Training and evaluation data was provided, the given dataset , after preprocessing uniformly was split in [2:1] ratio of [training :validation] samples ie [2640:1320] , with nearly proportional positive and negative Samples.The validation data is kept untouched since.

For ANN model (sec : 4.1) the Positive samples of the training data (20 in number) is repeatedly (40 times) appended to the training split and shuffled for over-sampling, with the possibility of overfitting the repeated positive samples. For SVM Model(sec : 4.2), the appending operation is repeated only 3 times before SMOTE oversampling.

After the ANN and SVM models are Trained and the models with best hyperparameters are saved[3], the whole training+validation data is made to run on both models to get the features for training the Ensemble model (as mentioned in section 4.3) .

For Ensemble model (sec : 4.3), the features of training+validation data is split again[4] in [2:1] ratio ,and on the training split, repeated oversampling is performed (35 times) before training is started. The precision, recall and f1-score for the validation splits are listed in Table 3 .

Model	Preci.	Recall	F1-Score	TrueSkill
ANN	1.0	0.4166	0.588	0.416
SVM	0.857	0.5	0.631	0.499
Ensemble	**1.0**	**0.933**	**0.965**	**0.933**

Table 3: Training Results

6 Observations and Test Result

Model	Preci.	Recall	F1-Score	TrueSkill
ANN	1.0	0.787	0.881	0.787
SVM	0.964	0.818	0.631	0.817
Ensemble	**0.967**	**0.909**	**0.937**	**0.908**

Table 4: Train + Validation Results (on complete dataset)

As can be seen in Table3 and Table 4 the ensembler outperforms both SVM and and ANN networks. The precision increases on ensembling, which reduces false positives. The increase in recall signifies more positive correct sample predictions. The evidence of the combinational effect

[3]Many iterations are performed and the best among all is chosen as the reference for the ensemble model

[4]This split is not the same split used for training SVM and ANN, since those models are fixed.

of ensembler has been verified during the training+validation check (where SVM and ANN were wrong on 7 instances each, while ensembler was wrong only on 4) and during testing (where, SVM and ANN predicting 6 and 7 positive results each, while ensembler predicted 13 positives) .

7 Other Approaches

KNN, Random Forests and LSTM in the place of RNN were used to perform initial experiments.SVM is found to outperformed Random forest and RNN outperforms LSTM but the F-Scores of KNN were better than that of SVM. KNN approach was discarded as the reason behind its success could be explained.

8 Acknowledgements

We would like to thank Dr. Asif Ekbal, Associate Dean, R&D, IITP for allowing us to participate in the event.We would like thank out Contingent Leader for his extended support.

References

Peter Bloomfield. *Fourier analysis of time series: an introduction*. John Wiley & Sons, 2004.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Adji B Dieng, Chong Wang, Jianfeng Gao, and John Paisley. Topicrnn: A recurrent neural network with long-range semantic dependency. *arXiv preprint arXiv:1611.01702*, 2016.

Andrej Karpathy. Stanford university CS231n: Convolutional neural networks for visual recognition. 2015. URL `http://cs231n.stanford.edu/syllabus.html`.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1): 1929–1958, 2014.

Deepanway Ghosal, Shobhit Bhatnagar, Md Shad Akhtar, Asif Ekbal, and Pushpak Bhattacharyya. Iitp at semeval-2017 task 5: an ensemble of deep learning and feature based models for financial sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 899–903, 2017.

Wikipedia contributors. Kepler object of interest — wikipedia, the free encyclopedia, 2017a. URL `https://en.wikipedia.org/w/index.php?title=Kepler_object_of_interest&oldid=814471996`. [Online; accessed 2-January-2018].

Wikipedia contributors. Transit (astronomy) — wikipedia, the free encyclopedia, 2017b. URL `https://en.wikipedia.org/w/index.php?title=Transit_(astronomy)&oldid=807367548`. [Online; accessed 2-January-2018].

Wikipedia contributors. Methods of detecting exoplanets — wikipedia, the free encyclopedia, 2017c. URL `https://en.wikipedia.org/w/index.php?title=Methods_of_detecting_exoplanets&oldid=817662929`. [Online; accessed 2-January-2018].

Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017. URL `http://jmlr.org/papers/v18/16-365.html`.