Astronomy & Astrophysics manuscript no. 0head_ML_3FGL
April 15, 2020

Machine learning methods for probabilistic catalogs

A. Bhat [⋆1] and D. Malyshev [⋆⋆1]

Erlangen Centre for Astroparticle Physics, Erwin-Rommel-Str. 1, Erlangen, Germany

Received September 15, 1996; accepted March 16, 1997

ABSTRACT

Context. Classification of sources is one of the most important tasks in astronomy. Sources detected in one wavelength band, e.g., in gamma rays, may have several possible associations in other wavebands or there may be no plausible association candidates.
Aims. In this work, we take unassociated sources in the third *Fermi*-LAT point source catalog (3FGL) and suggest associations to known classes of gamma-ray sources using machine learning methods trained on associated sources in the 3FGL.
Methods. We use several machine learning methods to separate *Fermi*-LAT sources into two major classes: pulsars and active galactic nuclei (AGNs). We evaluate the dependence of results on meta-parameters of the ML methods, such as the depth of the tree in tree-based classification methods and the number of layers in neural networks. We test the performance of the methods with a test sample drawn from the associated sources in 3FGL. We compare the predictions with the preliminary forth *Fermi*-LAT catalog (4FGL).
Results. Summary of results

Key words. Methods: statistical – Catalogs

Contents

1. Introduction

Catalogs of gamma-ray point sources are typically designed to have low false detection rate. Nevertheless, 469 sources out of 3033 in the third *Fermi*-LAT catalog (3FGL) [3FGL] have no associations in the forth *Fermi*-LAT catalog (4FGL) [4FGL]. This is much larger than the expected false detection rate in 3FGL arising from statistical fluctuations. For the majority of sources in 3FGL, which have no associations in 4FGL, the problem is not the false detection, but rather the association. For example, some sources can be detected due to deficiencies in the Galactic diffuse emission model. In this case, the statistical significance of the detection is high, but the association is wrong: the sources should be classified as a part of the Galactic diffuse emission rather than point-like sources. Another reason could be that two (or more) point-like sources in 3FGL are associated to a single extended source in 4FGL, or a single source

⋆ e-mail: aakash.bhat@fau.de
⋆⋆ e-mail: dmitry.mayshev@fau.de

is resolved into two sources. Again, this is a problem of classification (or association) rather than false detection.

Another reason for an absence of a previously detected source in a new catalog is variability. In particular, flat spectrum radio quasars (FSRQs) are highly variable active galactic nuclei (AGNs). If a source was active during the observation time of 3FGL but inactive afterwards, then its significance in the 4FGL can be below the detection threshold. The problem here is connected to a selection of a hard detection threshold of $TS = 16$ for 3FGL and 4FGL catalogs. Selection of a lower detection threshold could help to keep the variable sources inside the catalog, but it will not solve the problem, since the variable sources near the lower threshold can also disappear in the new catalog. Moreover, lower threshold would lead to more false detections due to fluctuations of the background. Thus, on the one hand, lower threshold can be useful in studies, where a more complete list of sources is desirable, while the higher false detection rate is not a problem. On the other hand, lower threshold can be problematic for studies where a clean sample is necessary.

The problems of the miss-classification of the sources and the detection threshold can be ameliorated with the development of a probabilistic catalog. In this catalog, each point-like object detected above a certain threshold would be classified into several classes with a set of probabilities, rather than associated to one class (or deemed unassociated). The classes can be various types of Galactic and extra-galactic sources, diffuse emission deficiency, extended source. Even the statistical fluctuation of the background can be viewed as one of the classes: in this case, objects with small statistical significance will have a high probability to the associated to the statistical fluctuation class. A user of such a catalog will have the freedom to choose the probability threshold for the class that he or she is interested in.

In this paper we will construct a probabilistic catalog using as an example classification of unassociated sources in the *Fermi*-LAT catalogs. We will start with the Third *Fermi*-LAT catalog (3FGL) and classify the unassociated sources into pulsars and AGNs using the associated sources in 3FGL for training of the classification algorithms. We will use several machine learning algorithms for the classification, e.g., random forest, boosted decision trees, logistic regression, and neural networks (since the number of features and the training sample are small, the neural networks will be rather shallow). We will show applications of the probabilistic catalog for predicting the number of pulsars among the unassociated source and in construction of the source counts as a function of their flux, dN/dS. Since unassociated sources on average have smaller flux than the associated ones, the dN/dS distribution for the probabilistic catalog extends to lower fluxes relative to counting only the associated sources. We will compare the prediction for the number of pulsars and the dN/dS functions with the 4FGL.

2. Choice of methods

2.1. General methodology

We will use data-driven approach in constructing the probabilistic catalog. The result depends on two inputs: data used for constructing the model and the choice of the method for classification. The data in our case will be sources in 3FGL with known classes. We will split the data into training and testing subsets. For the methods, we will consider four machine learning algorithms: boosted decision trees (BDTs), random forests (RF), logistic regression (LG), and neural networks (NN). The resulting probabilities of classification depend on the choice of the classification algorithm. Although some algorithms have slightly better performance on the test sample then others, the overall performance is relatively similar. As a result, we will report the classification probabilities for all four algorithms in the catalog, instead of selecting the "best" one. The difference among the predictions will serve as a measure of modeling uncertainty related to the choice of the classification algorithm.

2.2. Discussion of the choice of the classification algorithms

Decision trees One of the most simple and transparent algorithms for classification is a decision tree. In this algorithm, at each step the sample is split into two subsets using one of the input features. The choice of the feature and the separating value are determined by minimizing an objective function, such as misclassification error, Gini index, or cross-entropy. This method is very intuitive, since at each step the results can be described in words, for example, at the first step, the sources can be split in "mostly" Galactic and extragalactic by a cut on the Galactic latitude. At the next step, the high latitude sources can be further subsplit into millisecond pulsars and other sources, buy a cut on the spectral index around 1 GeV (pulsars have a hard spectrum below a few GeV) etc. One problem with decision trees is overfitting: if the tree is too deep, then it will pick up particular cases of the training sample, while too shallow tree would not be able to describe the data well. As a result, one needs to be very careful in selecting the depth of the tree. This problem can be avoided if a random subset of features is used to find a division at each node. This is the basis of the RF algorithm, where the final classification is given by an average of several trees with random subsets of

features used at each node. Another problem with the simple trees is that it can miss the classification of some subsets of data. In BDT algorithms, the final classification is given by a collection of trees, where each new tree is created by increasing the weights of misclassified samples of the previous step. Finally, simple trees predict classes for the data samples, while we would like to have probabilities of classes (also known as soft classification). RF and BDT algorithms, by virtue of averaging, provided probabilities. As a result, we will use RF and BDT algorithms rather than simple trees in this paper.

Tree-based algorithms, even after averaging in RF and BDT methods, have sharp edges among domains with different probabilities. In LR algorithm, the probabilities of classes are by construction smooth functions of features. In particular, for two-class classification the probability of class 0, given the set of features x, is modeled by sigmoid (logit) function

$$p_0(x) = \frac{1}{1 + e^{m(x)}}. \quad (1)$$

The probability of class 1 is then modeled as $1 - p_0(x)$. If $m(x)$ is a linear function of features, then the boundary between the domains, defined, e.g., as $p_0(x) = 0.5$, will be linear. More complicated boundaries can be modeled by taking non-linear functions $m(x)$. Unknown parameters of the function $m(x)$ are determined by maximizing the log likelihood of the model given the known classes of the data in training sample. A nice feature of the LR method is that it, by construction, provides probabilities of classes with smooth transitions among domains of different classes. A limitation is that the form of the probability function is limited by the sigmoid function in Equation (1).

We notice that if $m(x)$ is a linear function of features x, then the logistic regression model is obtained by an application of sigmoid function to a linear combination of input features. This is in fact a single layer perceptron, or a neural network without hidden layers, with several input nodes (each node corresponds to a features) and one output node, which corresponds to $p_0(x)$. The output value is obtained by a non-linear transformation (sigmoid) of a linear combination of features. Neural network with several hidden layers is obtained by a sequence of nonlinear transformations of linear combinations of features. In particular, the values in the first hidden layer are obtained by a non-linear transformation of linear combinations of input features. Then the values in second hidden layer are non-linear transformations of linear combinations of values in the first hidden layer etc. In the context of neural networks, the non-linear transformations are called activation functions. If the activation function for the output layer is sigmoid, then the output value (values) can be interpreted as probabilities. We notice that in this case the neural network is can be expressed by a logistic regression for some function $m(x)$, i.e., the neural network is then a particular way of constructing $m(x)$. Thus the only difference between LR and NN for the classification problems is the construction of the function $m(x)$. In this paper, for LR $m(x)$ will be constructed as a combination of low-order polynomials of the input features, while for NN, $m(x)$ will be constructed by taking linear input features and several hidden layers, e.g., 4 or 5, in a fully connected neural network.

2.3. Data and feature selection

As an example of the construction of a probabilistic catalog, we will use with the 3FGL catalog. For training and testing the methods, we use sources which have associations and no missing values in the catalog table. In this paper we will perform a two-class classification to separate PS into pulsars and AGNs. Thus, we subselect the sources, which are associated to either a pulsar or an AGN. After the training of the algorithms, we test the performance with the test sources and predict the classes of sources without associations, but have all features present in the catalog table. The general workflow will have the following steps:

1. Select data for learning and testing.
2. Train algorithms using the learning dataset. Tune hyper-parameters of the algorithms and test the performance on the test dataset, in particular, to avoid overfitting.
3. Choose the most important features and retrain the algorithms using the subset of the most important features.
4. Make prediction for unassociated point sources of the 3FGL. We also apply the classification for associated source. In this case we check if there are any outliers among the associated sources.

As a result of the analysis in this section, we obtain a catalog with probabilistic associations of sources in 3FGL. We will report the classification probabilities for all four algorithms and each source. In the next section we compare the predictions in the catalog with the new 4FGL catalog. We also construct a probabilistic catalog starting with the 4FGL.

2.4. Details on data selection

We restrict attention to associated and unassociated source but without missing values. We use the associated sources which were classified as either AGNs (agn, FSRQ, fsrq, BLL, bll, BCU, bcu, RDG, rdg, NLSY1,

nlsy1, ssrq, and sey) or Pulsars (PSR, psr). (Dima: would be nice to list all labels that we used for AGNs and pulsars in parentheses here), to get a list of 1905 sources. The rest of the sources without problematic values were then used as unassociated sources, which we used later on for testing and prediction.

Our methodology for classification was dependent on two things: The data that we had, which needed to be cleaned and the algorithms that we needed to apply. For this we decided on using the 3rd catalog of F-LAT (3FGL from hereon) for initial training and testing, the 4th catalog (FL8Y from hereon) for further testing and predictions. Our data was similar to that used by Parkinson et. al. We cleaned the 3FGL catalog to have sources which were both associated and unassociated but with no missing values.

The raw data of the catalog had a lot of different features that could be used for classification. However, going by the previous studies, we decided on using the most important features, which included Flux density and the error on it, spectral index, the curvature, hardness ratios (as defined by Parkinson et. al.), variablity, and also the galactic latitude, the last of which was used even in the classification of AGN and Pulsars (as opposed to Parkinson, who used it only for the young and milli-second pulsar distinction). In features where the values were high, we used the logarithmic scale to better seperate the sources. This includes Flux Density, curvature, error on flux, and the variability. The complete list of sources, along with some statistics, is given in the appendix. The influence of the features on the classification, especially the differences in the various methodologies is discussed in much more detail in the next section.

One of the main aims of our project was to understand and optimize the machine learning methods which we were using. So apart from the features which were in the data itself, we also theorized and experimented with the parameters of the algorithms themselves. We wanted to find the fastest and cost-effective way of using certain methods, without going into regimes of under and over-fitting the data. Parameters which we studied range from Depth and Number of trees in Forest based methods to the number of hidden layers and epochs in neural networks. The details are given in the next section, where we discuss our expectations and the resulting behaviour of our algorithms.

2.5. Details of algorithms

All of the machine learning algorithms were taken from the python module sklearn, including Neural Networks. A neural network using Keras was also attempted; however, due to the classification being on only two classes, we discarded it in favour of the sklearn algorithm which was much faster.
One of the main aims of our project was to understand and optimize the machine learning methods which we were using. So apart from the features which were in the data itself, we also theorized and experimented with the parameters of the algorithms themselves. We wanted to find the fastest and cost-effective way of using certain methods, without going into regimes of under and over-fitting the data. Parameters which we studied range from Depth and Number of trees in Forest based methods to the number of hidden layers and epochs in neural networks. The details are given in the next section, where we discuss our expectations and the resulting behaviour of our algorithms.

2.6. Old

When applied on the 3FGL known sources, using 1500 sources to train and the rest to test on, we found (for 10 seeds) the following:

Algorithm Name	Parameters	Accuracy
Random Forest	50 trees and 12 max depth	
Neural Network	200 epochs, 20 neurons, tanh	95.97
Gradient Boost	20,5 (0.1 lr)	95.8
Logistic Regression	all solvers	to put

Table 1. Testing Accuracy of 4 algorithms on 3FGL data

2.7. Details of the analysis

2.8. Data and Features

The total number of sources, including unassociated and associated, in the two catalogs is shown below.
[Add Table]

The features used for our analysis follow the same idea as the previous studies. The features, along with statistical and methodological details, are given below. A correlation matrix is presented for the most important features as well. The matrix is important for the case where there might be redundant features, in which case using only one of the two features would be a better idea.

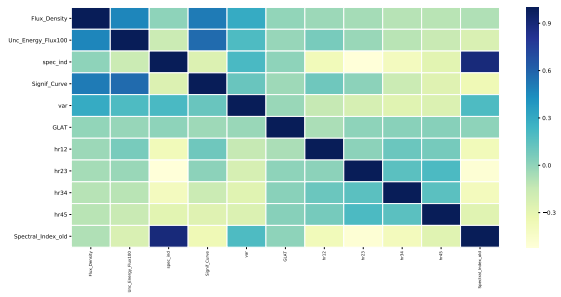

Fig. 1. Correlation matrix for the most important features

[Add Table of features for both catalogs]

Our initial hypothesis was that certain features would be more important for classification than others. For instance, as shown below, one can see a clear distinction between the regimes of AGNs and Pulsars, based on spectral idex and significant curvature. [Add image] While not clearly obvious from the get go, we were also interested in comparing the importance of features based on the algorithms that we were using. Due to the difference in the basic method of Random Forests and Neural Networks, we expected a slight shift in their reliance on certain features. Despite that we hypothesized that features with the most contribution would be among spectral index, variability, and the curvature; as already observed by Parkinson et. al.

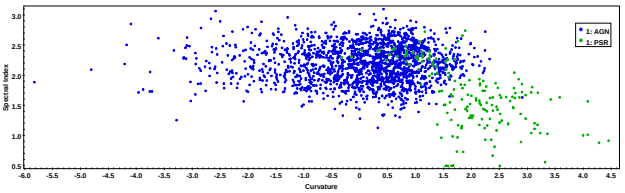

Fig. 2. Differences in AGNs and PSRs from the 3FGL catalog

2.9.

1. Describe the features that we use for the analysis.
2. Describe the objective function for minimization (accuracy of classification on learning sample). Weighted objective function: give more weight to pulsars, since there are fewer of them in the catalog.
3. Learning curve using all features? *Plot: classification accuracy using the total list of features for learning and test sample as a function of complexity parameter.*
4. Selection of the most important features. *Table: features vs algorithms. Columns: algorithms, rows: features, values: significance.*
5. Selection of meta-parameters. *Plot: classification results for the test sample using a subset of features.*
6. Train the final classifier. *Table: classification accuracy of the final classifiers for different algorithms using the test sample from 3FGL.*

Discuss the general features of the optimal algorithms: which features turn out to be important, what is the depth of the trees, the number of trees in random forests, the depth and number of internal nodes in the neural networks.

These importances were found to be consistent for various different algorithm parameters. So while the value might change a bit for different tree architechtures, for instance, the importances of these features were still pronounced.

2.10. Comparison of the classification algorithms

Plot: classification domains for a pair of features (or different pairs of features, e.g., latitude vs index, index vs curvature, latitude vs variability).

Probabilistic classification? Result: probability for a source to belong to a particular class. Result of classification: table of sources with probabilities for different algorithms. Final probability: the probability for one of the algorithms (for the most precise one?) and uncertainties determined from the other algorithms.

Discuss a few examples where algorithms give different predictions (are these sources at the boundaries of the domains).

Discuss examples where algorithms misclassify sources from the test sample.

In the case of test data, we worked with all the different classification algortithms, namely Random Forests, Ada Boost, logistic regression, and Neural Networks. Here we were mostly concerned with tweaking the parameters of the classification algorithms involved, minimizing the cost of computation and aiming for the most efficient way of classification.

2.10.1. Random Forests

The two main parameters involved in Random Forests are the number of trees and the maximum depth of the trees involved. Figures below shows one instance of the accuracy as a function of maximum depth when the number of trees was kept constant, and as a function of number of trees when the maximum depth was kept constant. We performed 100 runs for different numbers of trees (also called esimators in the code) and for the maximum depth. In the case of training, we found

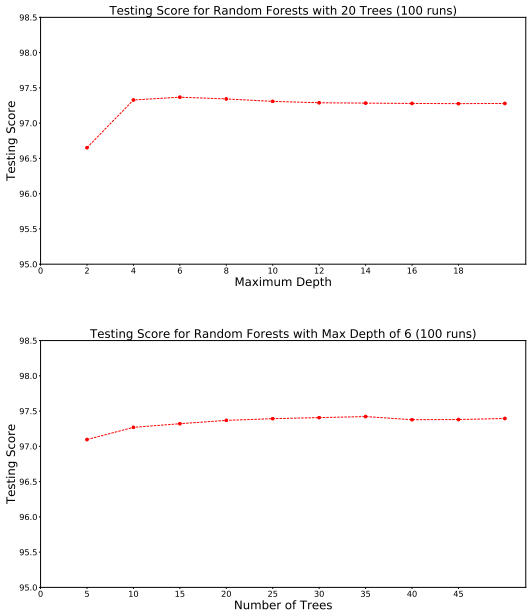

Fig. 3. Random Forests in Training

no significant differences when using weights for our data set. However, we chose to use a inversly weighted dataset since that allows us to generalize the algorithm more. The influence of weights will also be discussed more in the case of unassociated data, in the next section.

After running tests on various architectures we chose a Random Forest with 20 trees to avoid over-fitting, and a maximum depth of 12.

Since Random forests are based on decision trees, they allow us to characterize feature importances based on how helpful a feature was to split a tree. In our case, using all 10 features discussed above we found the following feature importances for two architectures of random forests with similar accuracies of 97.5:

Feature Name	RF (20,12)	RF (50,6)
Flux Density	0	0
Unc Energy Flux100	0	0
Spectral Index	0.13	0.17
Significant curvature	0.33	0.29
var	0.13	0.11
hr12	0.07	0.06
hr23	0.06	0.05
hr34	0.05	0.05
hr45	0.20	0.23
GLAT	0.04	0.04

Table 2. Feature importances for different Algorithm

Our hypothesis about feature importances turned out to be correct, as curvature, variability, and spectral index were the most important features. The last hardness ratio was also seen to be quite important, most probably reflecting the end of the spectrum where the AGNs and PSRs shift from each other.

2.10.2. Neural Networks

In the case of neural networks we were concerned with the number of epochs that one would need to tweak, along with a dependence on the number of neurons in the hidden layers. A final improvement involved checking whether multiple hidden layers would actually add to such a classification algorithm or not.
As can be seen in the figure, a complex network with two hidden layers (100 and 5 neurons) reaches the maximum accuracy pretty fast. The results becoming more consistent at higher epochs. A similar result was found for a network having two hidden layers but with only 20 neurons in the first layer. However, such networks could also lead to overtraining, and therefore it is important to check whether such a high accuracy could perhaps be reached by less complicated algorithms, which would drastically reduce the chances of overtraining and allow for a more flexible classificaition methodology.

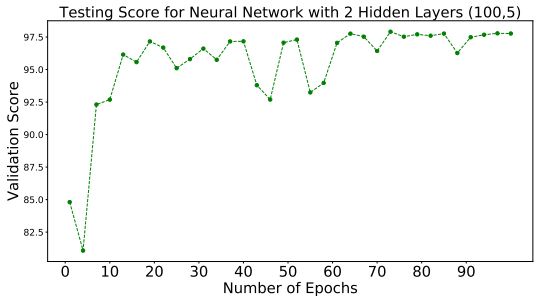

Fig. 4. Example of a figure for one column.

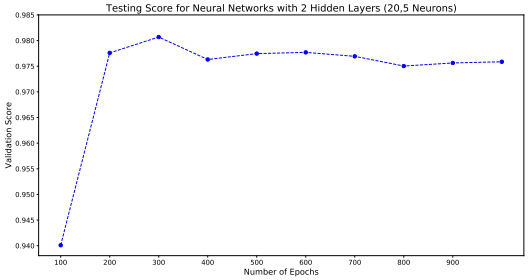

Fig. 5. Neural Network

A consistent and accurate result is found even for networks with only one hidden layer with 20 and 5 neurons in the hidden layer respectively. There seems to be no significant dependence for the number of epochs above 30, and even a simple network with one layer and

5 neurons shows a high accuracy for only 30-40 epochs.

3. Prediction for unassociated sources in 3FGL and comparison with 4FGL

Comment: Create a table with sources which are more likely to be pulsars (select about 20 the most likely candidates). Compare the accuracy of the algorithm for the sources which have an associate now in the 4FGL catalog.

In this section we use the best algorithms from the previous section to predict classes for the unassociated sources in the 3FGL. We then use the associations which exist for some of these sources in the 4FGL to check the accuracy of our methods on the unassociated data. In this section we work only with Random Forests, Neural Networks, AdaBoost, and Logistic Regression.

3.1. 3FGL Unassociated sources with Association in 4FGL

There were a total of 286 sources without associations in 3FGL but which had a corresponding assocation in 4FGL. We trained our algorithms on the entire associated data from the 3FGL, and then used tested our algorithms on these 286 sources. The probabilistic version is discussed in the next section.

The following were the optimized results we obtained for this case:

Algorithm Name	Parameters	Accuracy
Random Forest	50 trees and 12 max depth	96.22
Neural Network	200 epochs, 20 neurons, tanh	95.97
Gradient Boost	20,5	95.8
Logistic Regression	all solvers	to put

Table 3. Testing Accuracy of 4 algorithms on 3FGL unassociated data

As can be seen above, the best accuracies were found with less complicated models, which allowed bias to be low. The models were complicated enough to neither under, nor overtrain.

3.2. 3FGL Probabilistic classification

4. Prediction for unassociated source in the 4FGL catalog

5. Conclusions

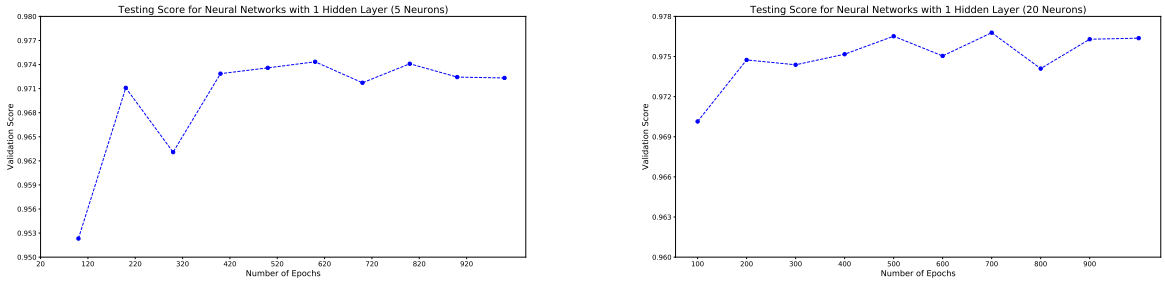

Fig. 6. Neural networks

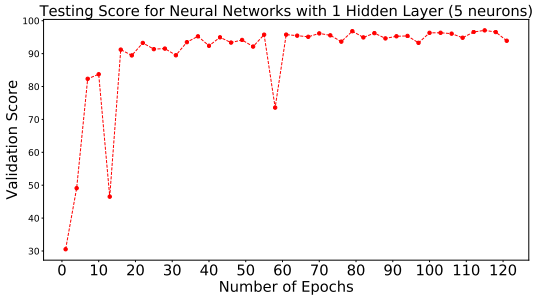

Fig. 7. Neural Network

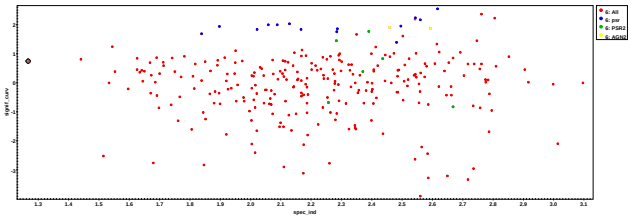

Fig. 8. A comparison of the outliers in the test predictions

Appendix A: Appendix

If we need one.