(16-995 A)
—— MRSD: Independent Study Final Report ——
Applied Machine Learning

Slip Estimation using Proprioceptive Sensors on Planetary Rovers

---

Under the supervision of: Dimitrios (Dimi) Apostolopoulos

By: Sambuddha Sarkar

November 2017

Contents

1 Introduction

1.1 Motivation

Calculating and correcting wheel slip in planetary rovers is crucial to any unmanned mission. Wheel slip can cause pre-planned trajectories to be violated during critical mission procedures. Slip estimation using proprioception is a new concept where-by relatively cheaper inertial sensors and existing sensors related to drive mechanisms can be utilized to infer indirectly slip instead of expensive instrumentation clusters added onto the wheel to measure slip directly. One way to do it is to use machine learning to predict slip. The primary aim of this project is to effectively estimate and classify **wheel slip** of a vehicle in a particular terrain. The wheel slip can be classified into three classes namely: **Low**, **Moderate** and **High**. It is important to classify slip instead of just predicting the slip value because the traction controller is designed according to the slip categories. The goal is to perform Supervised Classification and Detection of rover slip in different soil types and terrains.

This report contains the **Slip Estimation Pipeline** that was developed to aid in the appropriate model selection which can effectively classify slip and some results pertaining to that. The procedure of model selection is not an exact science and is heavily dependent on the engineer's understanding on the machine learning models and also the intuition gained from observing the data.

1.2 Data Description

1.2.1 Baseline Data: MIT Single Wheel Test Bed

The baseline data is from a setup at MIT's terra-mechanics lab, known as the *MIT single wheel test bed*, whereby a replica of the wheels used on the mars rover: Curiosity, has been provided by NASA. The dataset is provided under special permission by the researchers at MIT and is not publicly available. The data has ground truth labels where the true slip value has been calculated using a specialized setup of torque and force sensors.

There are 131,060 instances of data in the dataset and there are 4 attributes of the dataset namely: **Motor Torque, IMU Acceleration along Z-axis, IMU Pitch Angle along X-axis, IMU Acceleration along Y-axis**. The data has been visualized (Moderate: Blue, Low:Red, High:Cyan) in Figure 1.

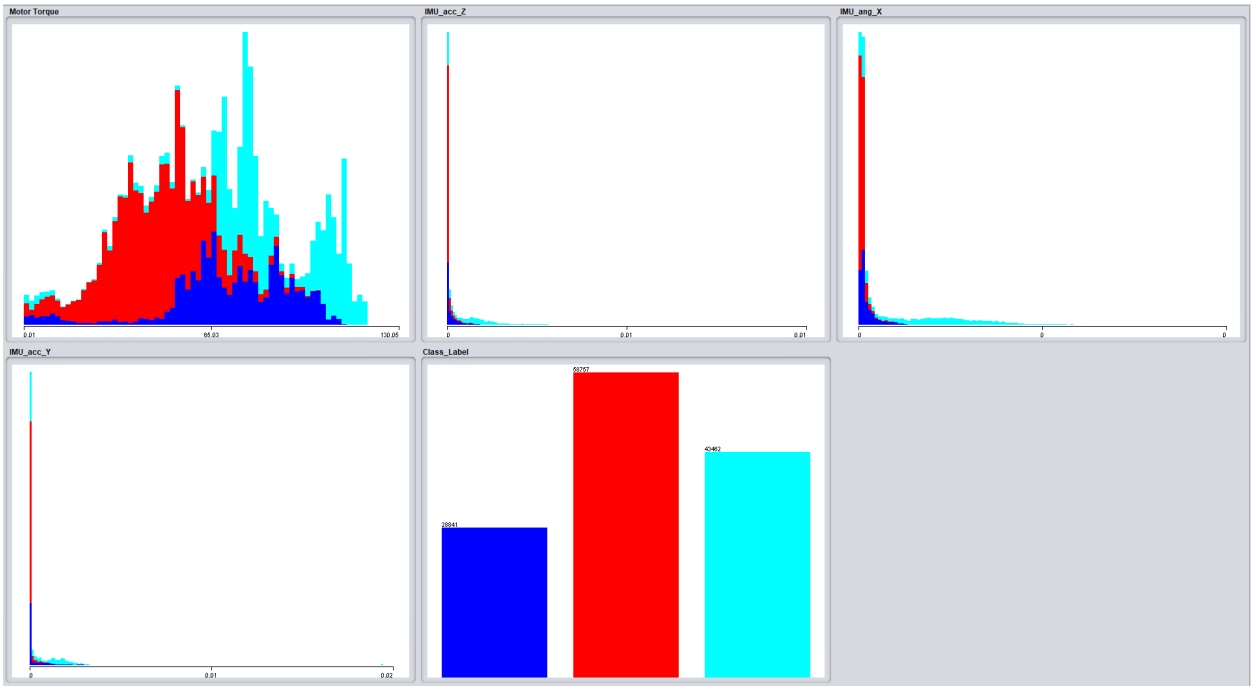

Figure 1: Attributes visualization according to classes.

1.2.2 Training and Testing Data: LATUV Rover

Now for the purpose of this study and report the dataset being used is generated by field tests of the LATUV Rover at Proto Innovations. The MIT single wheeled test bed data was used as a baseline. The data from the LATUV rover consists of sensor readings from **IMUs, Drive Motors and Steering Motors** which are described in the report on the vehicle's data logging architecture.

1.2.3 Baseline Performance

The plan is to use supervised learning models to predict slip using the sensor inputs as features. To establish a baseline a **Decision Tree : J48** has been used and a reported accuracy of **87.8407%** has been observed where 115214 have been correctly classified as shown in Figure 2.

```
Number of Leaves  :     716

Size of the tree :      1431


Time taken to build model: 6.17 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      115124               87.8407 %
Incorrectly Classified Instances     15936               12.1593 %
Kappa statistic                          0.8066
Mean absolute error                      0.1203
Root mean squared error                  0.2483
Relative absolute error                 28.1753 %
Root relative squared error             53.7424 %
Total Number of Instances           131060

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.658    0.036    0.839      0.658    0.738      0.683    0.931     0.834     Moderate
                 0.957    0.138    0.849      0.957    0.900      0.815    0.959     0.932     Low
                 0.918    0.026    0.945      0.918    0.931      0.898    0.977     0.970     High
Weighted Avg.    0.878    0.078    0.879      0.878    0.875      0.813    0.959     0.923

=== Confusion Matrix ===

     a     b     c   <-- classified as
 18980  7728  2133 |     a = Moderate
  2313 56256   188 |     b = Low
  1323  2251 39888 |     c = High
```

Figure 2: Classification Report.

It is important to note that the boundaries of the classes can be modified, i.e. the percentage slip is the original ground truth and a simple set of boundary conditions has been applied to the percentage slip to split it into three distinct classes. The boundary conditions can be modified and even the number of classes can be modified though for this report the slip boundaries are retained as is i.e. the three slip classes: High, Medium and Low.

1.3 Software Packages for Offline Testing and Online Testing

It is important to note that a custom Python package (based on Python existing python libraries like: Numpy, Scipy, Pandas and Scikit-Learn) was created and developed at PI by me during this Phase to facilitate the model selection procedure. This package follows the coding practices of any other Python library and will be completed with full-fledged API documentation by the end of this Phase. The model selection will be explained in the successive sections as well as the **Offline** and **Online** software architectures.

1.4 Goals

The expected outcomes of this project are:

(a) Find the best performing classifier for slip estimation.

(b) Find the best attributes for slip classification.

(c) Find the optimum tuning parameters for the classifier used for slip estimation.

2 Model Selection

The model selection for the Machine Learning based Slip Estimator is crucial as the ground truth slip value obtained from the wheel velocity and vehicle body velocity from localization systems like GPS is unavailable on Planetary Rovers. The model selection procedure is highlighted in Figure 3. The candidate machine learning models being tested are as follows:

1. DT: Decision Trees (Cart C4.0 -J48)
2. RF: Random Forest Trees (Ensemble)
3. ET: Extra Trees (Ensemble)
4. SVM: Support Vector Machines (Non-linear Kernel)
5. MLP: Multi-layer Perceptron
6. KNN : K- Nearest Neighbors

There had been plans to use to Gaussian Nave Bayes, but as we will observe in the Exploratory Data Analysis section, the distribution of the features is not Gaussian and hence Gaussian Nave Bayes is not well suited for this application as it will learn a learning that is not appropriate for our application.

The model selection procedure is as follows:

2.1 Filtering

We apply appropriate filters (like sliding variance and sliding average with variable window size) to the dataset to get rid of unwanted variations in the incoming data stream from the array of proprioceptive sensors. In all the analysis throughout the report, we use the following filter configuration applied using our custom Filtering Module:
Window size 10 and Type of Filter Sliding Variance as from our testing we found out this filter configuration is most suited for noise suppression in the data.

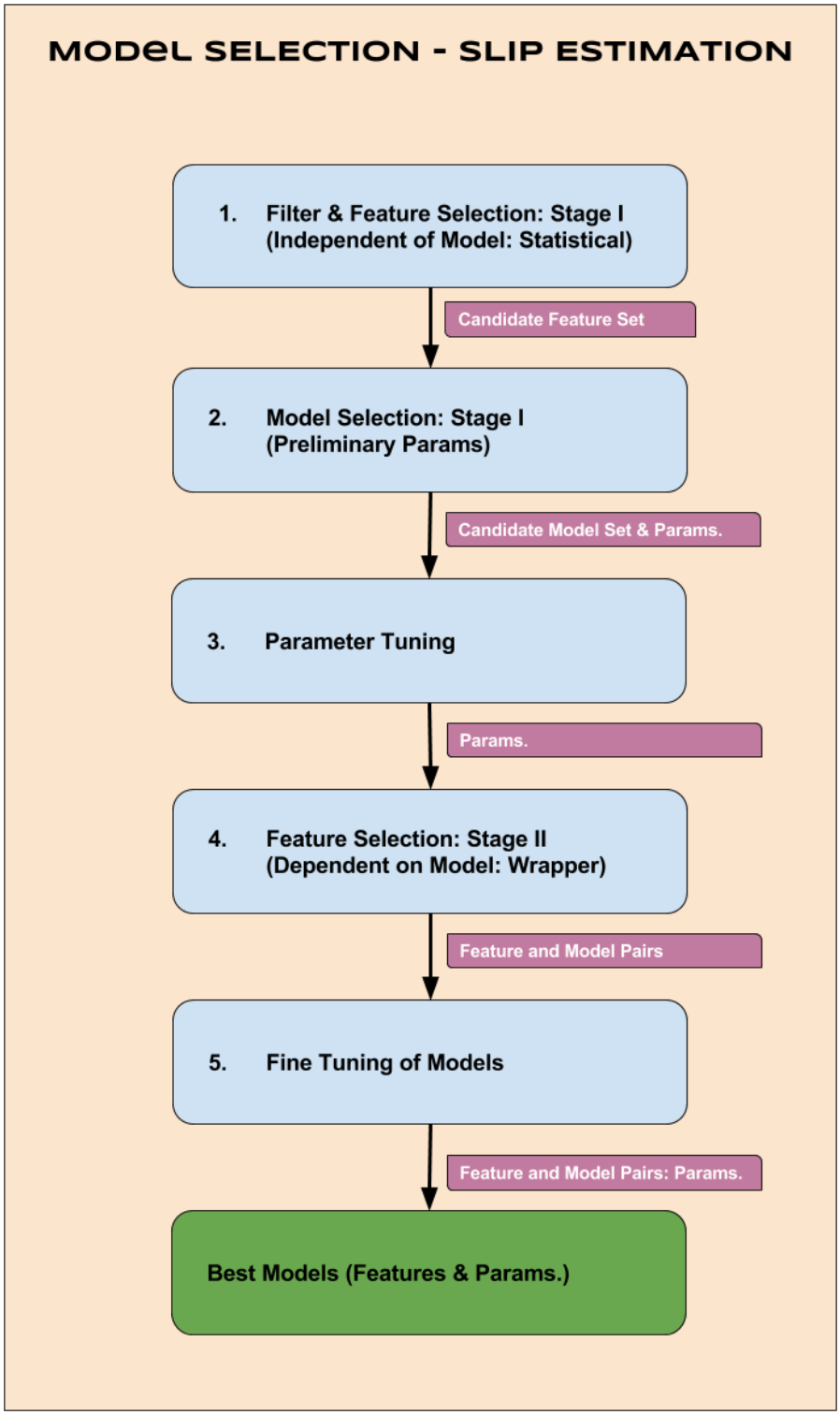

Figure 3: Model Selection.

2.1.1 Feature Selection (Statistical Based)

We perform namely 3 tests on the filtered data (will be referred to as data from now on throughout the remained of the report) and do feature ranking to determine which features are most useful which will aid us in the second stage of Feature Selection i.e. the Wrapper Based Methods. In this part we utilize 2 test based on Information Gain with respect to the labeled training data and 1 test based on the Mutual information and Statistical Similarity, these methods are elaborated on later.

2.2 Model Selection-Preliminary

Using Exploratory Data Analysis and some dry test runs on small data samples drawn from the dataset representations we eliminate the poorest performing model. This is the stage where we ruled out Gaussian Nave Bayes as a potential candidate with the help from our intuition of our data from exploratory data analysis.

2.3 Parameter Tuning

In this stage as coarse parameter tuning is performed on the Machine Learning models using human judgement and observation. This is to ensure that when the wrapper method is being tested, the models are tuned enough to show promising results or comparable to other ones to say the least. We adjust tree depth appropriately in Tree based learning models and also adjust the number of estimators in the Ensemble Methods. A non-linear kernel like a Radial Basis Function is used in the SVM classifier, appropriate hidden layers and learning rates are adjusted in the MLP and the neighbor likelihood is adjusted in the KNN algorithm.

2.4 Feature Selection (Wrapper Method)

In this stage we utilize wrapper based methods like Recursive Feature Elimination to prune out features by iteratively running the models trained and tested over a set features in each iteration. We then pick the features which help the models perform the best. In this stage we obtain model-feature pairs which are then tuned in the final stage to obtain the best suited model-feature pair.

2.5 Fine Tuning Models

This is the final stage of the model selection, in this stage all the best model-feature pairs are tuned using a exhaustive or randomized hyper-parameter tuning workflow over a set of user defined parameters related to the model.

3 Exploratory Data Analysis

It is tempting to use powerful machine learning and statistical models to find a solution to a problem, but before applying any learning techniques to solve a problem statement it is very important to understand and summarize a dataset without making any explicit

assumptions about its contents. It is a crucial step to take before diving into machine learning or statistical modeling because it provides the context needed to develop an appropriate model for the problem at hand and to correctly interpret its results. To aid our understanding about the dataset it is important to use quantitative and visual inspection methods vis-a-vis do Exploratory Data Analysis (EDA). We need this stage to understand the data we are handling and what type of models will be appropriate for this type of classification problem. In our EDA we have used histograms to better see the distribution and nature of the data.

3.1 MIT Dataset

We observe the MIT dataset as the baseline because prior research regarding model based Slip Estimation has already been performed. In Figure 4. we can see that except Motor Torque all the other features in the MIT dataset has a decaying exponential distribution, where-as Motor Torque has Gaussian distribution. This strongly tells us that a Gaussian based learner wouldn't be able to effectively learn the representations in the dataset.

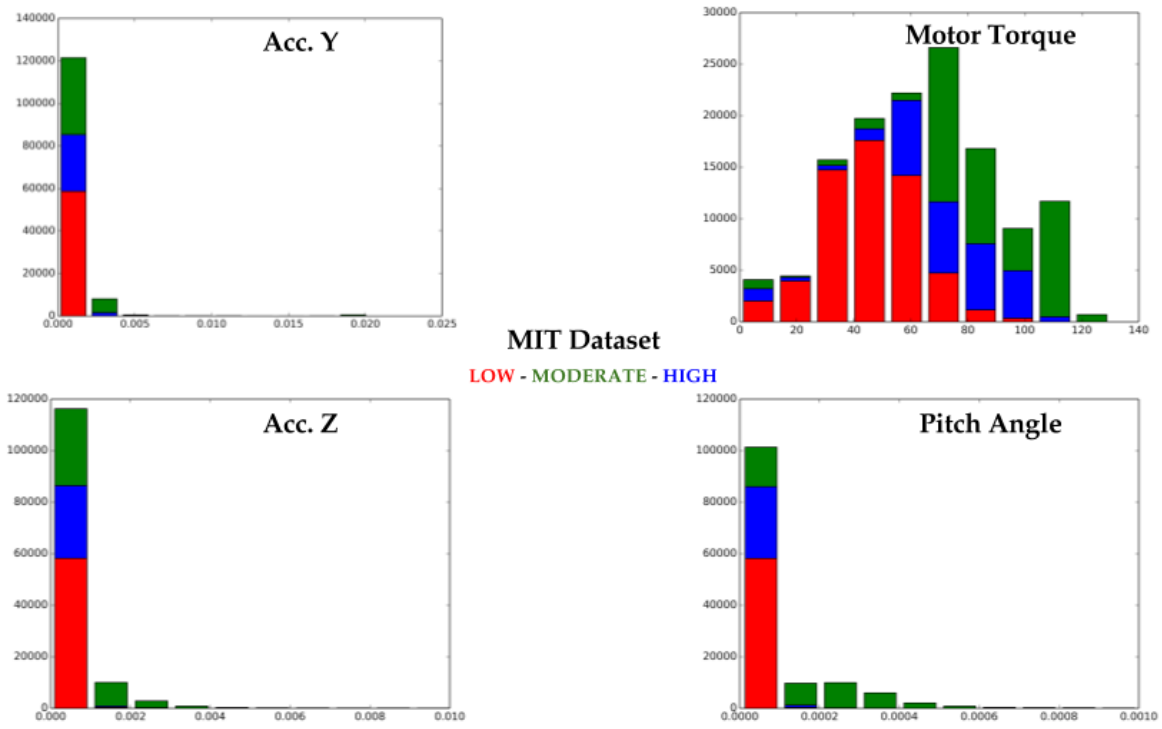

Figure 4: Histograms of MIT dataset.

3.2 LATUV Dataset

Moving onto a more appropriate dataset for LATUV, we visualize the features of from the LATUV testing done at GASCOLA on August 9th, 2017. It is important to note that the data-visualizations are only for the Left-Front wheel or the 'LF' wheel of LATUV (this is to simplify the visualizations as the other 3 wheels have very similar data profiles and it would

be overwhelming to have 7 figures for all 3 wheels in this report). We can observe from the visualizations of the 7 features related to the LF wheel in Figure 5 and Figure 6. The data representations are not at all Gaussian and we can also observe major class imbalances, that is the majority of slip cases are Low Slip events. This leads us to two conclusions.

1. Gaussian based classifier wouldn't be effective in our application.

2. This might give us false sense of good performance, hence when we test the models and compare performances we utilize Cohen's Kappa Statistic paired with Accuracy to determine a true prediction rate of the classifiers. Cohen's Kappa statistic takes into account how much agreement can be expected by chances. We use this instead of ROC or F1 score as ROC is only good when we have to check probability that a random positive example will be ranked above a random negative example and F1 score is more suited for test-mining related classifiers where we have high-dimensional data.

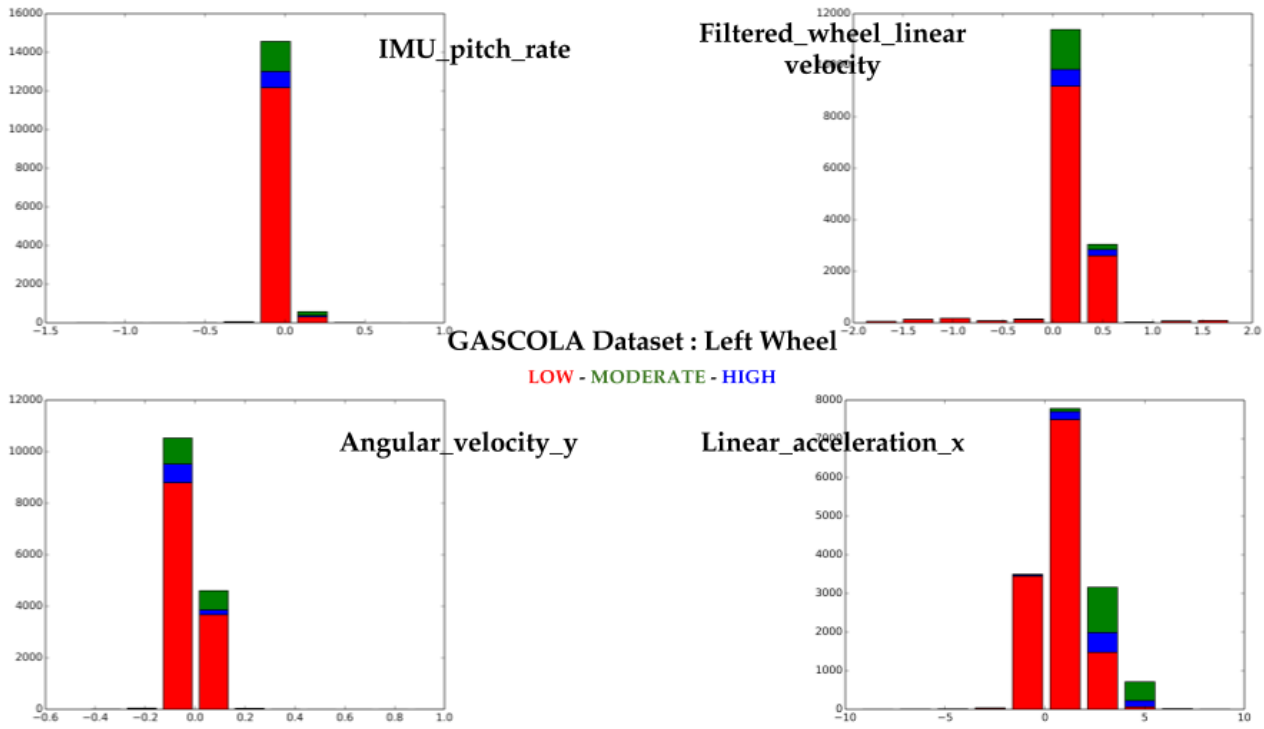

Figure 5: Histograms of LATUV dataset.

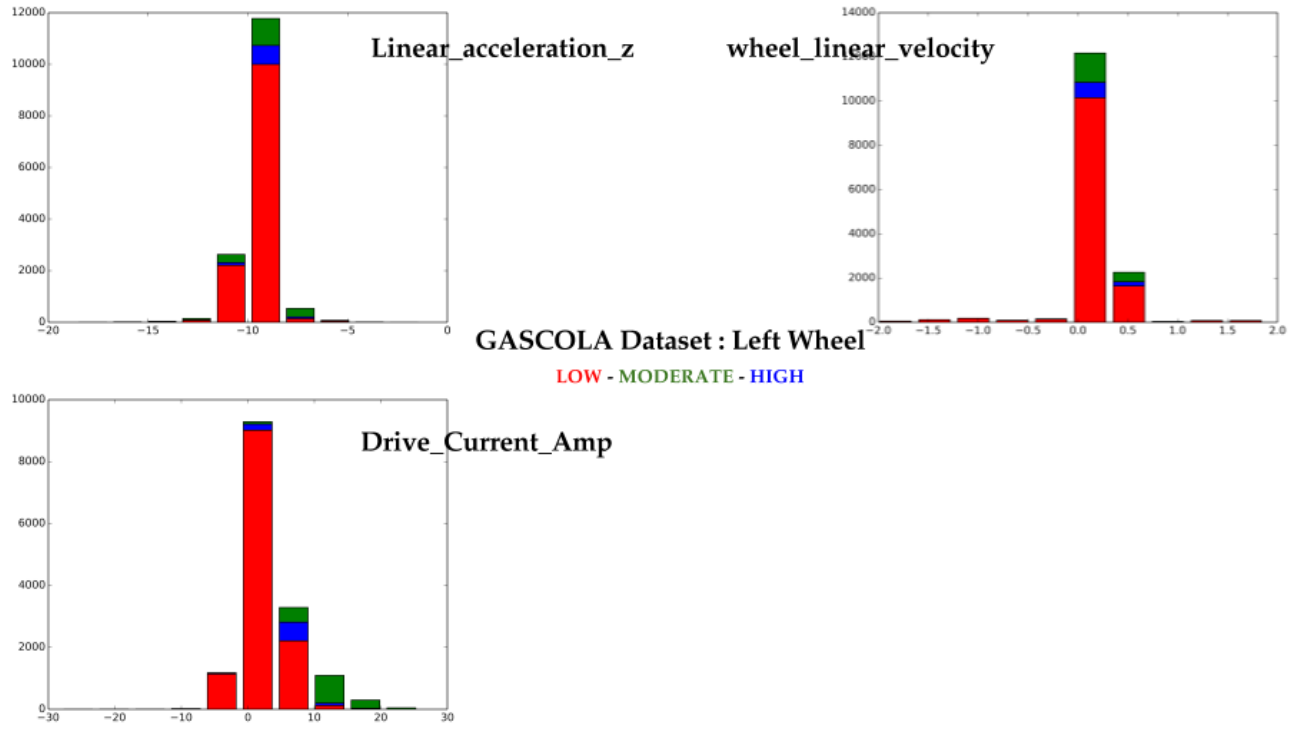

Figure 6: Histograms of LATUV dataset.

NOTE: The data-visualizations of the tests run on 43rd Street Concrete Facility is very much similar to the GASCOLA data in terms of the nature of distribution, hence it is gives us a sanity check that there is a repeating pattern in the data during slip events and that an effective rule can be learned to predict or classify slip.

4 Offline Machine Learning Training and Testing pipeline.

The offline architecture visualized in Figure 7. captures the software implementation of the Model Selection pipeline discussed before. The software architecture captures the information flow between different stages of the model selection process and maps the dependencies between each stage. The custom Python package developed at PI has modules which are embedded in the different stages to show their usage. The legend for the figure is at the top left corner and can be referred to see the state of progress and package dependency.

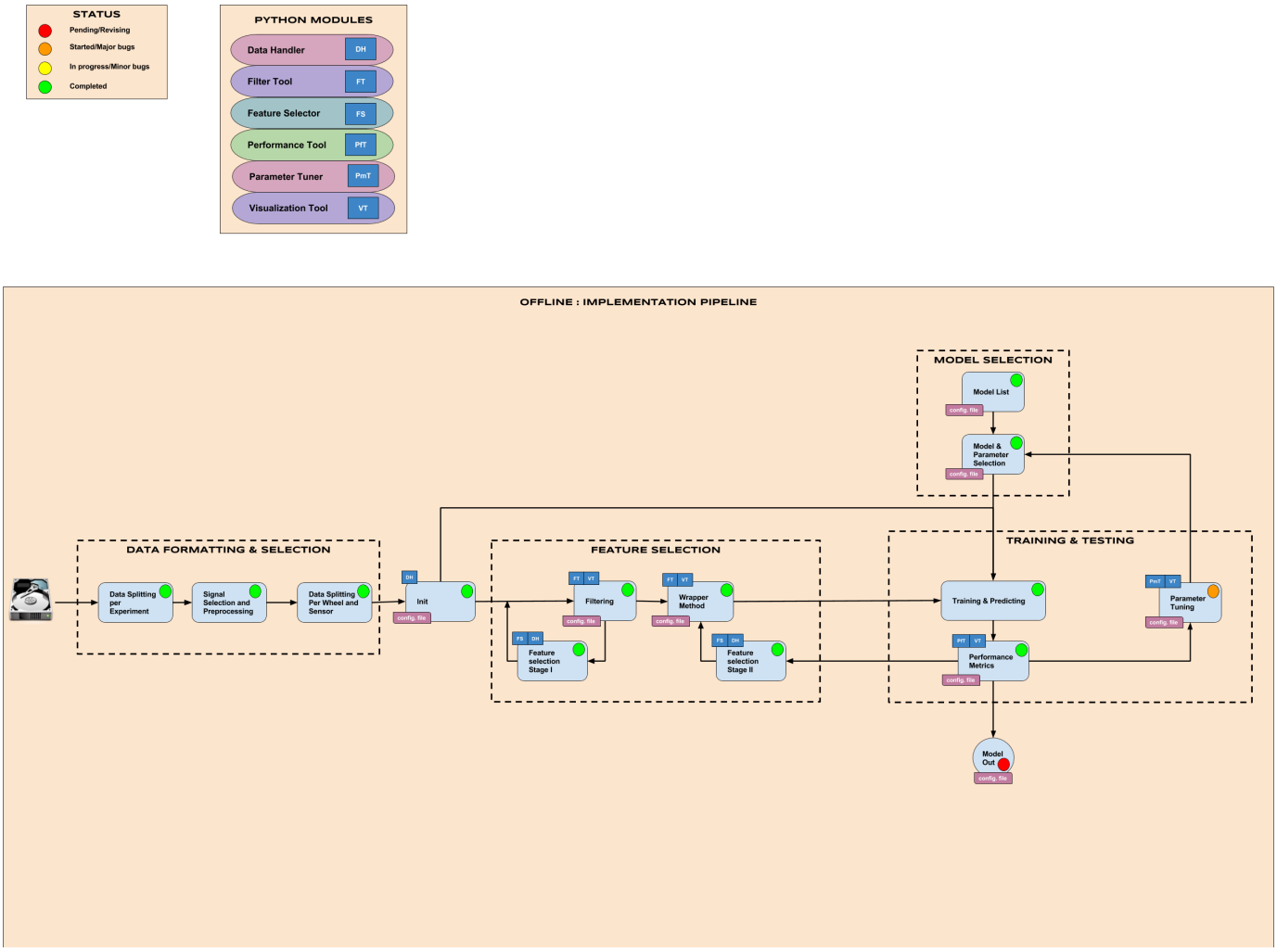

Figure 7: Architecture of Offline Machine Learning Training and Testing pipeline.

5 Feature Selection

This section outlines the details of the feature selection stages and methodologies.

5.1 Feature Selection: Statistical Methods

In this section the 3 Statistical Feature Selection metric are described and the appropriate test results from running the Feature Selection module are visualized in Figures labeled 8,9 and 10, all the seven features from the GASCOLA field-test for the LF wheel are ranked. The other wheels and field-tests results are not included for the sake of compactness of the report and also one example is enough to demonstrate the feature selection method.

5.1.1 Fisher Score

It is a similarity based scoring metric based. It is based off the Fisher Information and Invariantized Entropy computation, it basically captures the similarity in the probability distribution between different features and target labels, it gives a high score to feature with high similarity with target labels and low similarity with other features. The formulation for the scoring is beyond the scope of this report. Further reading can be found at: http://stat.columbia.edu/~jakulin/Int/jakulin05phd.pdf

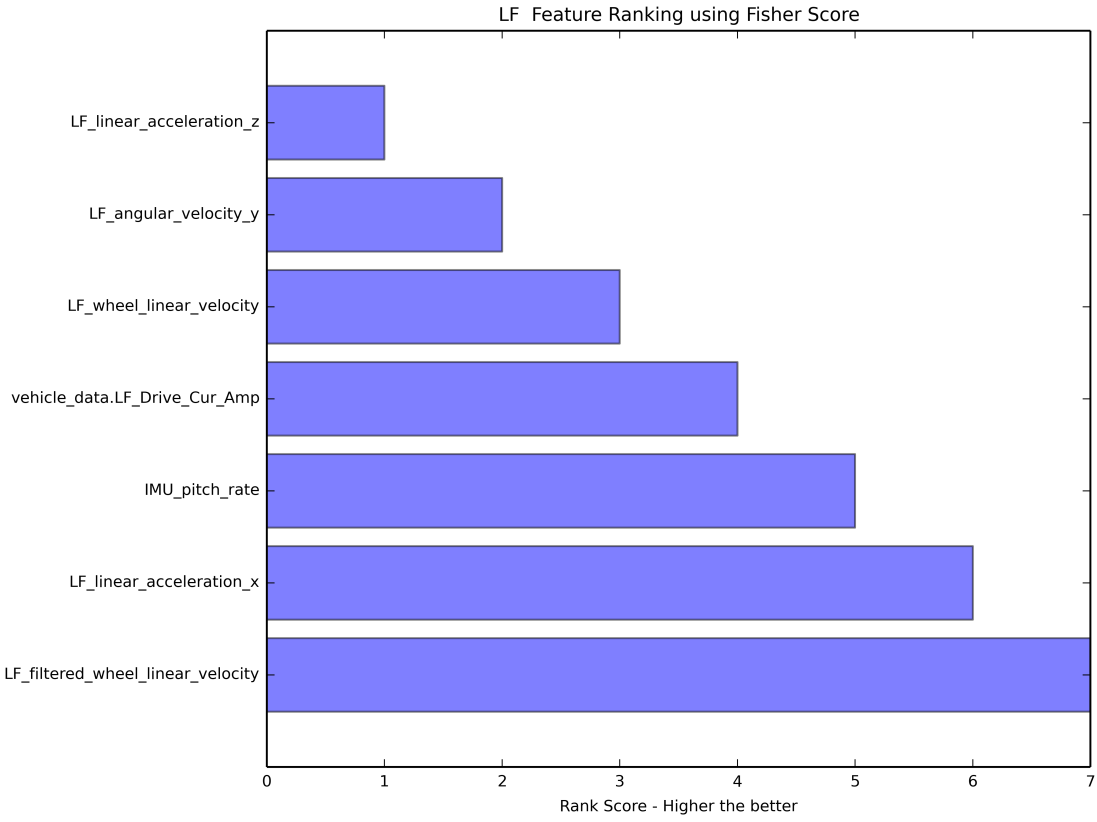

Figure 8: Feature Ranking: Fisher Score

5.1.2 ICAP

It is a mutual information and information gain based powerful feature ranking algorithm. The higher the score, the better the feature. It is called Interaction Capping. The formulation can be found here: http://www.jmlr.org/papers/volume13/brown12a/brown12a.pdf

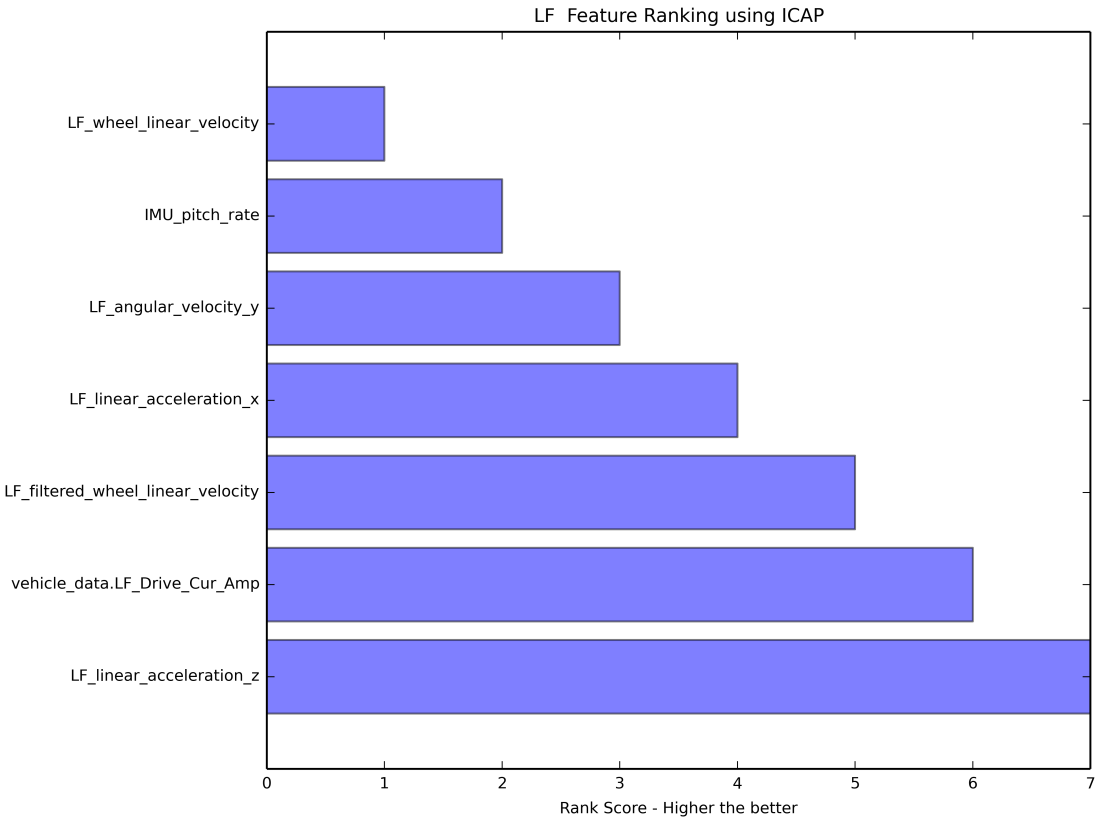

Figure 9: Feature Ranking: ICAP

5.1.3 JMI

It is called Joint Mutual Information, this metric for ranking utilizes the information theory framework to extract mutual information and interdependency between distributions. The higher the score, the better the feature. The formulation can be found here: http://www.jmlr.org/papers/volume13/brown12a/brown12a.pdf

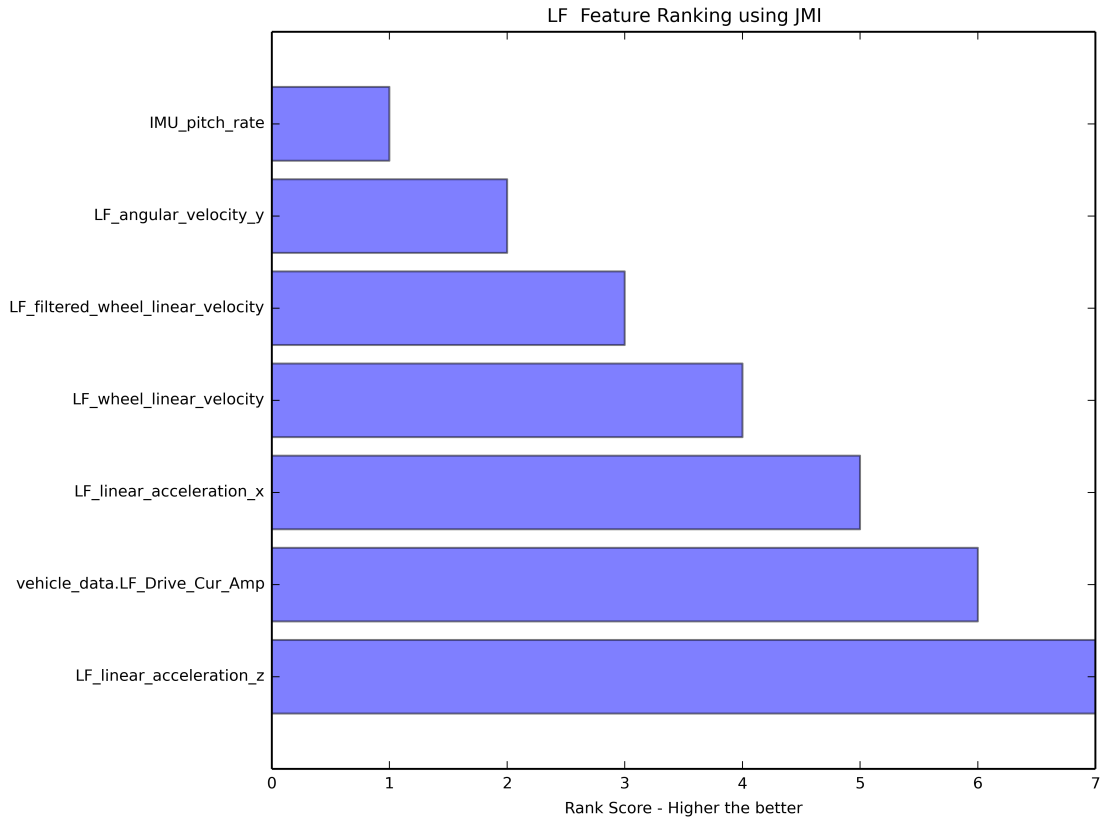

Figure 10: Feature Ranking: JMI

As observed from Fisher Score, ICAP and JMI feature ranking, there is no seeming pattern between importance of features. We didn't a very informative insight into what features are crucial, but as we only have 7 features to consider, these rankings by scores do tell us one thing that none of the features are bad, i.e. none of them scored 0, that means we can keep these features and see how they pair up with the models.

5.2 Feature Selection: Wrapper Methods

This is a model specific feature selection method, whereby the model and features are both are paired and tested for best performance. In this module we utilize Recursive Feature Elimination to prune out features by iteratively running the models trained and tested over a set features in each iteration. Our goal is to obtain model-feature pairs that perform the best. In the Figures labeled 11,12,13,14,15 and 16, the Wrapper based feature selection is run on all the 6 Classifiers. The Prediction Rate is a weighted score of Kappa Statistic and Prediction Accuracy and it is calculated using K-fold cross-validation to avoid misrepresentation of the algorithms performance. The figures are self-explanatory and show how model performance is affected when different feature combinations are picked. The relation between the Statistical Based Feature Selection scoring and the results from the Wrapper Method Feature Selection will be discussed upon in the future report.

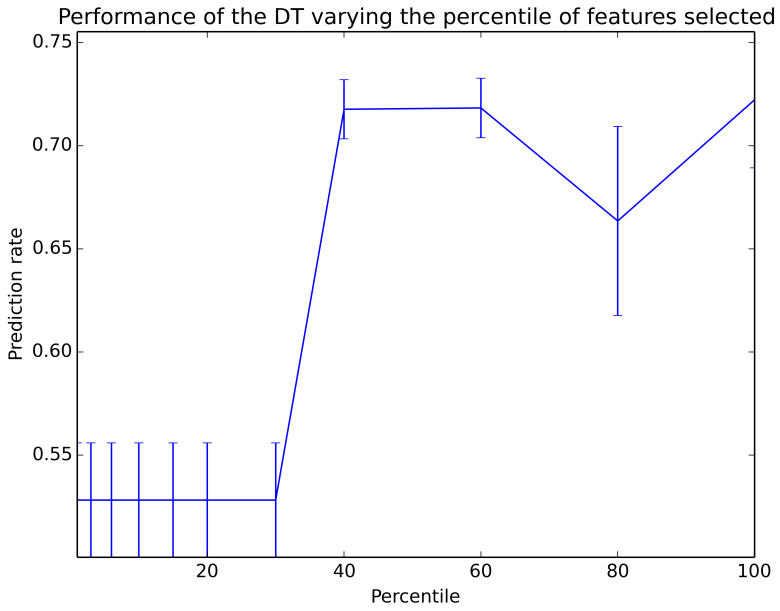

Figure 11: Feature Selection using Wrapper Method for DT

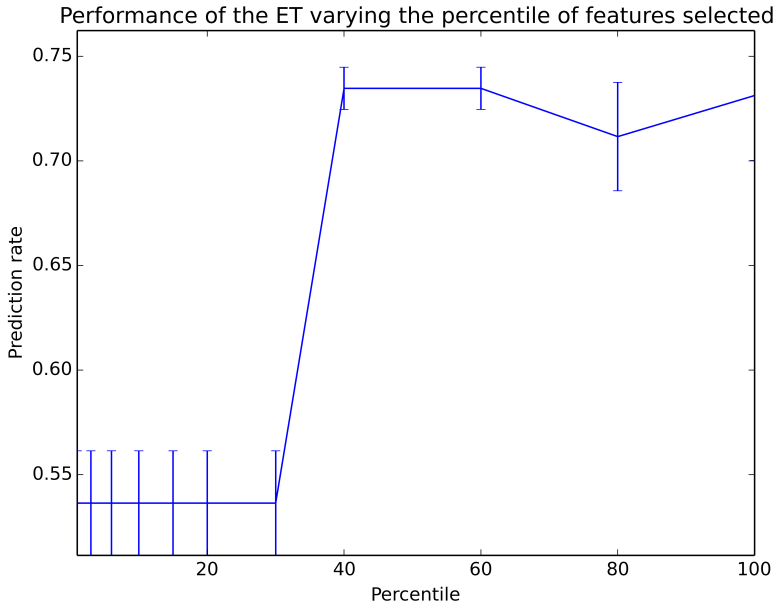

Figure 12: Feature Selection using Wrapper Method for ET

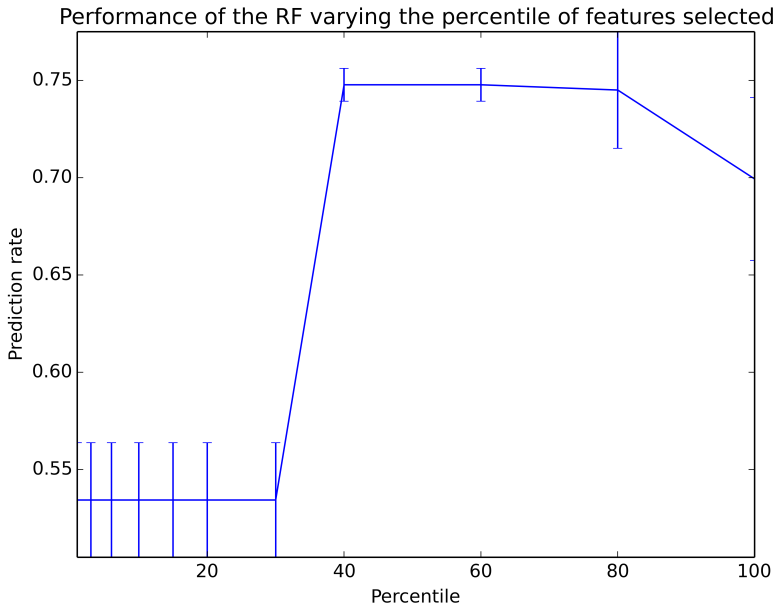

Figure 13: Feature Selection using Wrapper Method for RF

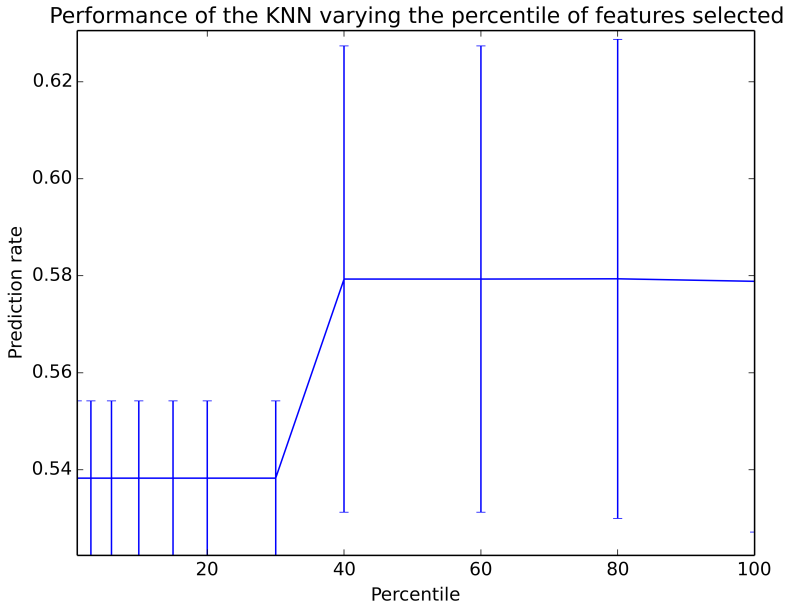

Figure 14: Feature Selection using Wrapper Method for KNN

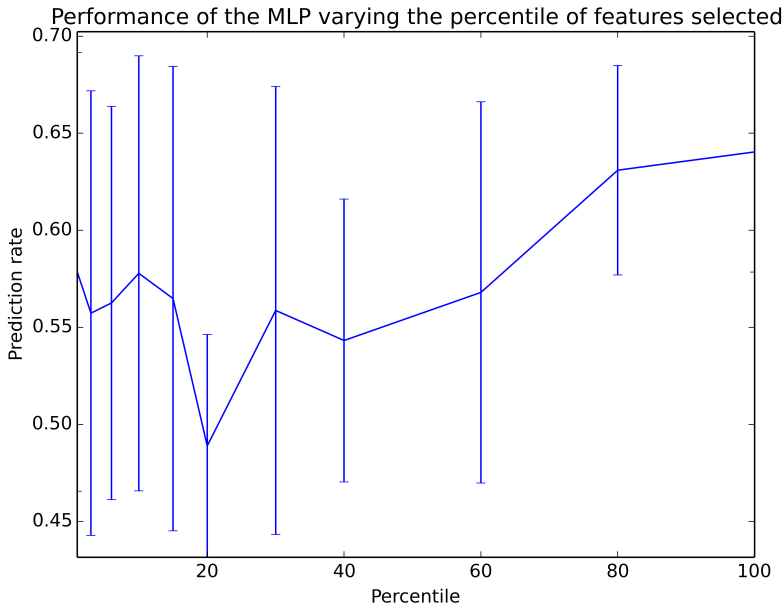

Figure 15: Feature Selection using Wrapper Method for MLP

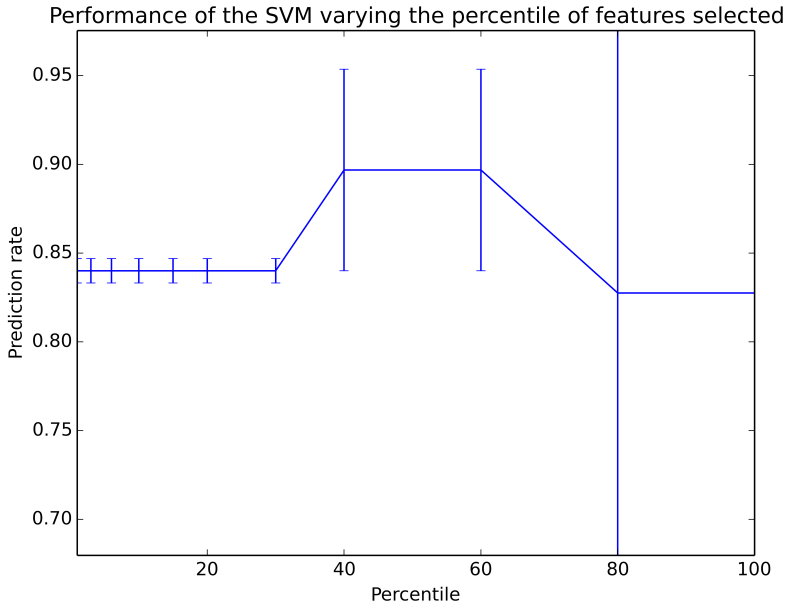

Figure 16: Feature Selection using Wrapper Method for SVM

It can be observed that Trees outperform other methods, which is a good news as we can visualize the trees and it being a rule based model we can actual extract the rules learned by the Tree based learning models which can then be used to derive parametric equations from the learning rules.

6 Performance Tuning

Model-feature pairs obtained from Wrapper Method will be tuned using a exhaustive or randomized hyper-parameter tuning workflow over a set of user defined parameters related to the model. This module is still under development and will be completed in the next phase of the STTR.

7 Online Slip Prediction Pipeline: ROS Integration

The Online Slip Estimation architecture has been visualized in Figure 17. The software architecture captures the information flow between different stages of the online slip estimation process and maps the dependencies between each stage. The custom Python package developed at PI has modules which are embedded in the different stages to show their usage. The legend for the figure is at the top left corner and can be referred to see the state of progress and package dependency.

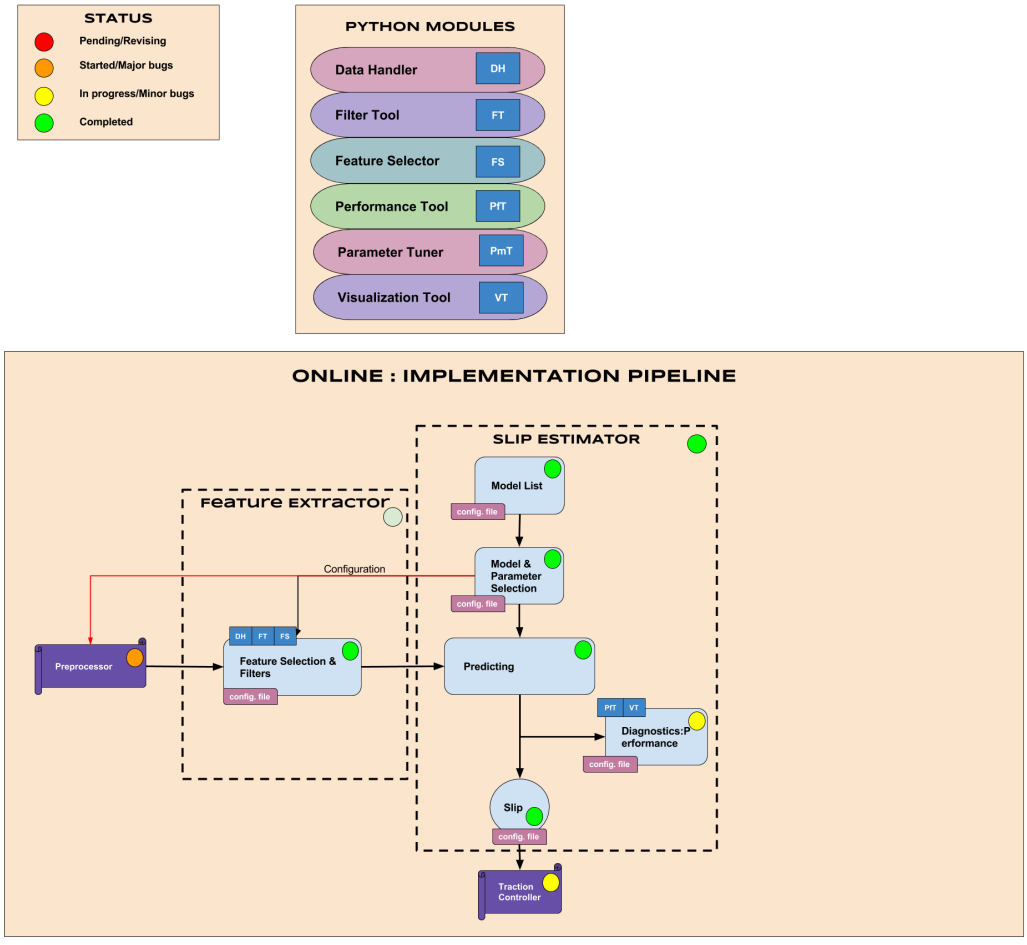

Figure 17: Online Slip Predictor Software Architecture

8 Online Slip Estimator Module: Tests

From the results of the Wrapper Method, we observe that Tress are consistently outperforming other classifiers. As we don't have deployed the hyper-parameter tuner module yet, we pick Decision Trees out of all the Tree based classifiers because of it's simplicity and model size and also known tuning parameters from trial and error. We emulated the Online Slip Estimator Module in ROS on the LATUV computer and ROS-played the BAG file of the GASCOLA field test to check the real-time performance of the Slip Estimator. We had trained the Decision Tree based model on 70% of All Data(all data from particular field-test and terrain) from the LF wheel and then loaded the model into the online slip estimator module. When we ran the tests, we observed that the Slip Predictor had an accuracy of over 97% on the whole of All Data, and it also performed surprisingly well (accuracy of over 91%) when it encountered unseen data from another terrain! The Figures 18 and 19 visualize the performance of parameter tuned Decision Tress on emulated stream of data. The slip class are numbered as such:

- 0: Low Slip
- 1: Moderate Slip
- 2: High Slip

(it is important to note that all available 7 features were used instead of the model feature pair as we wanted to the effect of overfitting of Decision Tress.)

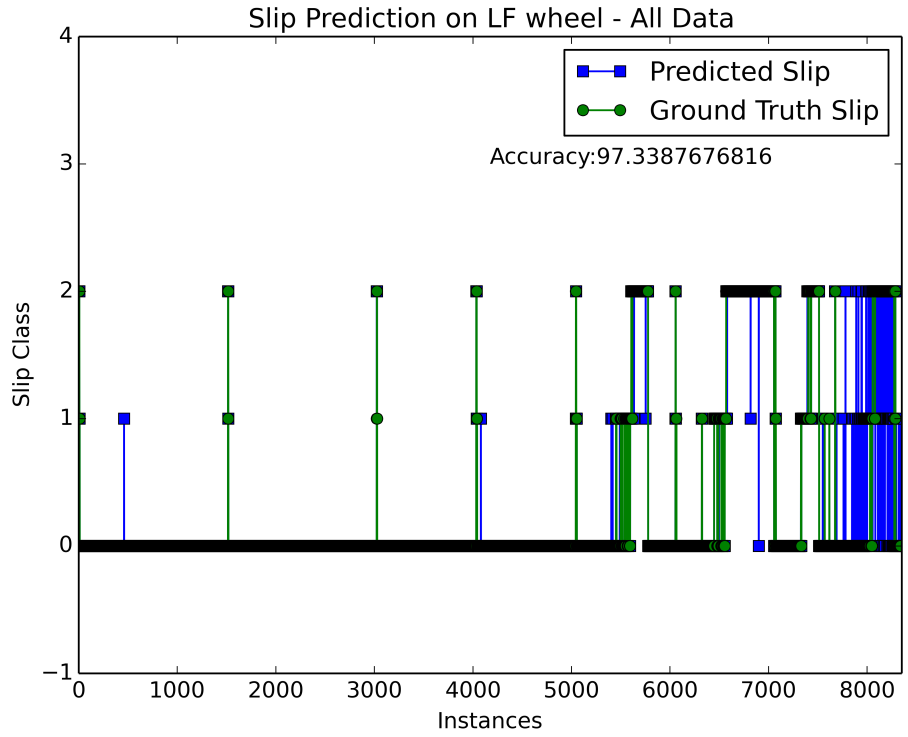

Figure 18: Online Slip Prediction: All Data from one terrain.

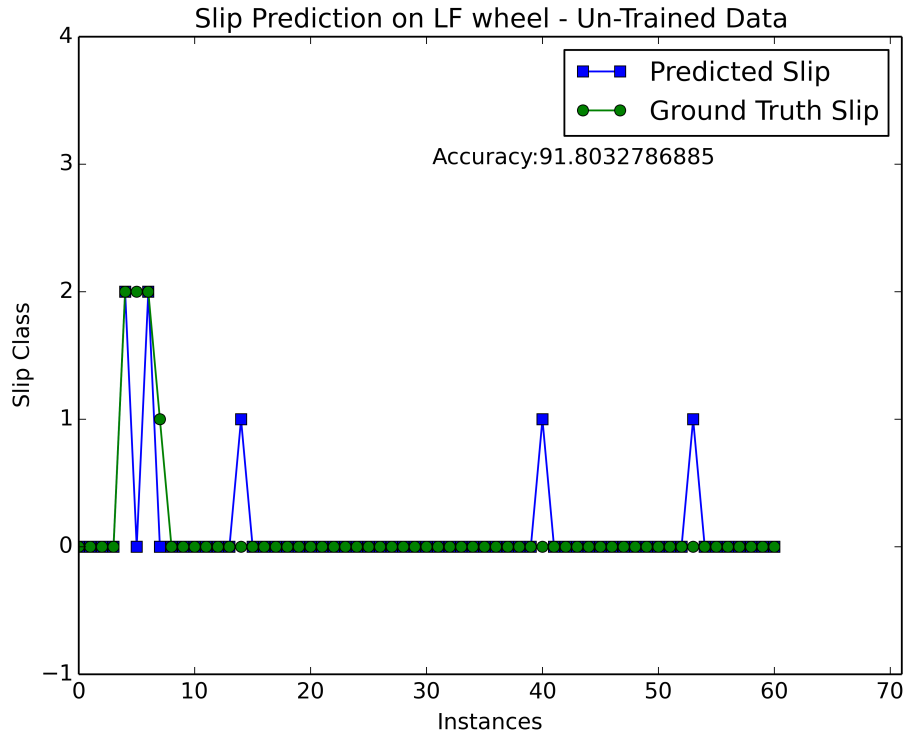

Figure 19: Online Slip Prediction: unseen data from different terrain.

9 Future Work and Investigations

Once the hyper-parameter tuner is completed, and the final model selection pipeline will be ready, we will be able to rank models by different criteria, like Prediction Accuracy, Kappa Statistic, Recall, Precision, Memory Size and Computational Efficiency. This will give us a sure shot way of knowing that the model that we are picking is best suited for our application in Slip Estimation.

Some investigations related to the slip class boundaries, i.e. how the different classes of slip are split and what would be the optimum boundaries for differentiating different slip zones from a data perspective is a worthy endeavor. Also the use of online learning techniques like Reinforcement Learning and Inverse Reinforcement Learning instead of Supervised Learning based Classifiers is a desirable research direction in the field of traction control as these online models can learn new terrains efficiently in absence of prior supervised and label training data.