Introduction

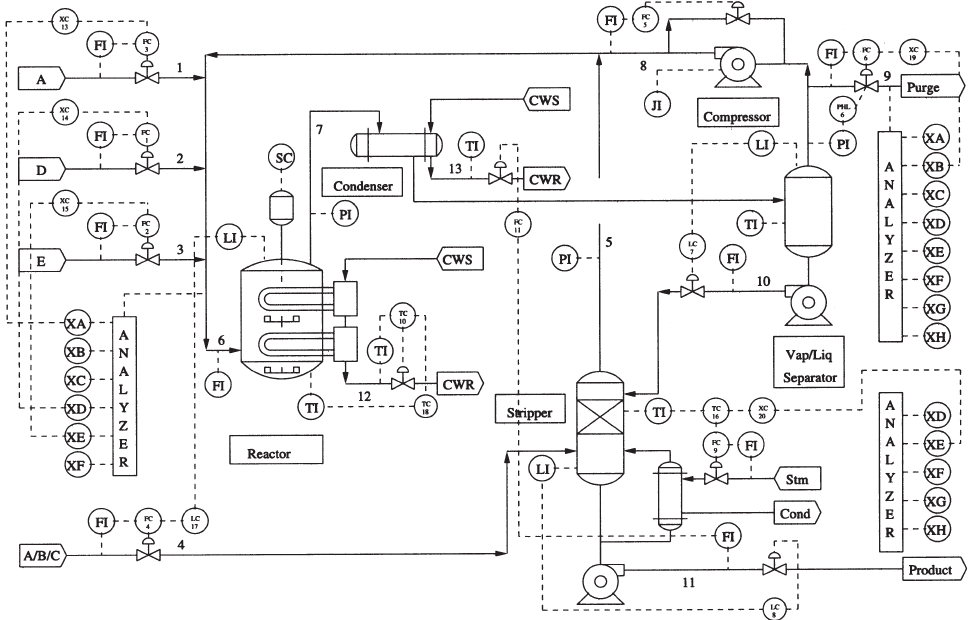
Process Monitoring and Fault diagnostic (PM-FD) represent a very active research field. Various PM-FD methods have been proposed in literature but generally they can be classified in three categories: Knowledge based, Model based and data-driven models [8]. Compared to the other models the data-driven model is gaining more attraction as it only depends on the measured process variables. In this work we want to compare different data driven PM-FD methods using an industrial benchmark process simulation named Tennessee Eastman Process (TEP).

The research question

The objective is to compare different data driven PM-FD models’ performance using training/test datasets generated from the simulated TEP and applying different comparison methods like Fault Detection Rate and False Alarm Rate to identify to best performing model. We will also apply Cross-project prediction to evaluate the possibility to use models build for easily detected faults to predict the hardly detected faults

Description of the data

The simulated process is shown below



The dataset contains simulated data generated at sampling periods of 3min. The simulated data contains normal operation, i.e., d00 as well as simulation for the 21 faulty conditions (d01 to d21). For each case, two sets of data are produced: training datasets and testing datasets. The training datasets are used to construct statistical predictive models. The testing datasets are used to estimate the accuracy of the built classifier.

The training and testing datasets include all the manipulated and measured variables except XMV12, for a total of 52 variables (or features). A feature vector at a particular time is given by :

X=[XMEAS(1),…, XMEAS(41),XMV(1),…,XMV(11)]T

The table below summarizes the datasets

|  |  |  |
| --- | --- | --- |
|  | **No Fault run (d00)** | **Fault Run (d01 - d21)** |
| **# samples training** | 500 | 480 |
| **# samples testing** | 960 | 960 |
| **Sampling time** | 3 min | 3 min |
| **# features(Variables)** | 52 | 52 |

Data Exploration

1. **Data preparation**:

We will develop a perdition model for the “Step” type of Faults for within project prediction

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fault number** | **Process variable** | **Type** | **+ Gaussain Noise** | **comments** |
| IDV(0) | Normal |  |  | Normal operation |
| IDV(1) | A/C feed ratio, B composition constant | Step | Yes |  |
| IDV(2) | B composition, A/C ration constant | Step | Yes |  |
| IDV(3) | D feed temperature | Step | Yes | Check with Mutual informtion |
| IDV(4) | Reactor cooling water inlet temperature | Step | Yes |  |
| IDV(5) | Condenser cooling water inlet temperature | Step | Yes | Check using mutual information |
| IDV(6) | A feed loss | Step | Yes |  |
| IDV(7) | C header pressure loss-reduced availability | Step | Yes |  |

Then apply it for the “Random” type of Faults, i.e. Cross project prediction

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fault number** | **Process variable** | **Type** | **+ Gaussain Noise** | **comments** |
| IDV(8) | A, B, and C feed composition | Random variable | Yes |  |
| IDV(9) | D feed temperature | Random variable | Yes | Hard detection. No observable changes in mean, variance or peak time [7]. Will be ignored initially. |
| IDV(10) | C feed temperature | Random variable | Yes |  |
| IDV(11) | Reactor cooling water inlet temperature | Random variable | Yes |  |
| IDV(12) | Condenser cooling water inlet temperature | Random variable | Yes |  |

1. **Evaluation Metrics**:
   1. Fault detection Rate / False Positive Rate

|  |  |  |
| --- | --- | --- |
|  | Predict- # of samples in the ith class | Predict- number of samples in the other classes |
| Actual- # of samples in the ith class | p | b |
| Actual- # of samples in the other classes | q | d |

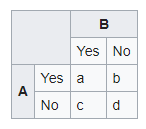
Fault Detection Rate (FDR) = p / (p+b) (True Positive from confusion Matrix)

False Positive Rate (FPR) = q/(q+d) (False Positive from confusion matrix)

* 1. KAPPA coefficient

Kappa measures the reliability of a model by measuring the inter-rater agreement for qualitative items. It is an indicator of the agreement between two observers taking into account the possibility of agreement per chance. Therefore, it is a more robust measure than the percentage.

Below is the method of calculating Kappa coefficient



|  |  |
| --- | --- |
|  | |
|  |  |

*Source:* [*https://en.wikipedia.org/wiki/Cohen's\_kappa*](https://en.wikipedia.org/wiki/Cohen's_kappa)

Interpreting the Kappa coefficient depends on the studied problem. In the literature there is a set of guidelines developed by Landis&Koch [9] to interpret Kappa values. This is provided in the table below

|  |  |
| --- | --- |
| **Kappa** | **Agreement level** |
| <0 | No agreement |
| 0–0.20 | Slight agreement |
| 0.21–0.40 | Fair agreement |
| 0.41–0.60 | Moderate agreement |
| 0.61–0.80 | Substantial agreement |
| 0.81–1 | Almost perfect agreement |

This convention is used in report to interpret Kappa values.

ROC is also considered for evaluation

1. **Tools**

In the capstone the following tools are used

**R programing language**: using RWeka package in R to explore/invoke WEKA’s prebuilt machine learning packages

**Weka**: (through RWeka) used for its reach portfolio of data analytics algorithms as well as the available evaluation metrics like Kappa

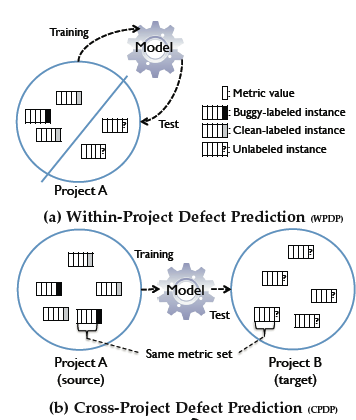
**Python**: mainly used for data preparations.

1. **Methodology followed**
   1. **Data standardization**

The data is standardized with zero mean and unit standard deviation. This is under the assumption that most of the variables are normally distributed – Distribution graphs provided in the next section. This is important as we have measurements with different units and also it helps most of machine learning algorithms to converge. No normalization is applied as we are interested in outliers in our case.

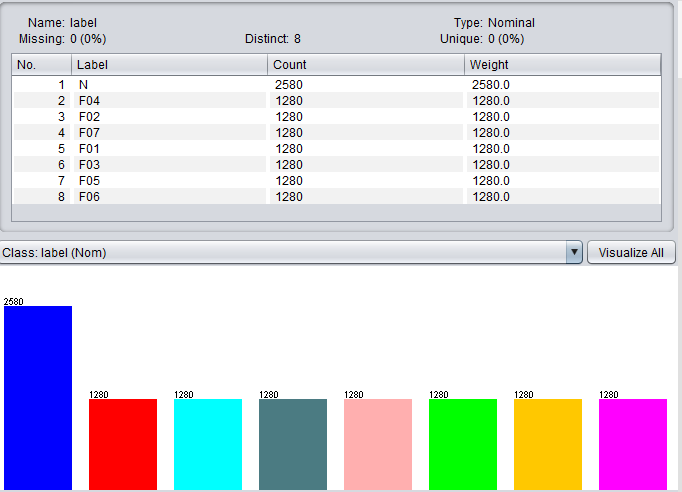
* 1. **Cross-Project prediction**

Cross-Project prediction is a new trend in defect/anomaly detection. The idea is to predict defects for new projects (or in new datasets) that lack enough defect data by using prediction models build by other projects or using other datasets [10]. The graph below summarizes the difference between within-project vs. Cross-project predictions.



* 1. **Steps followed**
* Construct data frames from the multiple CSV files provided
  + Two data frames created. One for 7 anomalies with the normal data (type Step), and another one for 5 anomalies + Normal of type Random.
* Data is divided into 70% training and 30% test sets.
* 10-folds cross validation is used on training data to select model and parameter
* Feature selection is done on the training data. New model is generated with the new features using steps above. The model is applied to the test set.
* Test the model against test data.

1. **Data Exploration results**
   * No missing values, NaN or null
   * All data was standardized. One dataframe that encompasses the 7 faults and the Normal state was created.
   * Fault classes are balanced. The Normal data could be considered unbalanced though if considered against the total number of faulty data

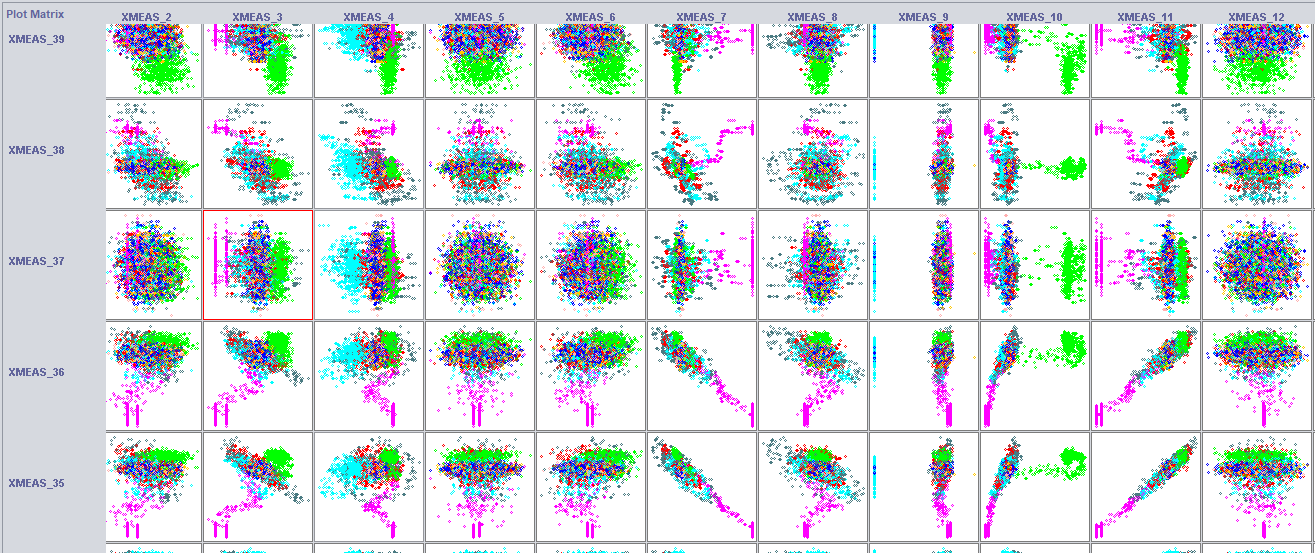


* + Data was standardized

Attributes Interactions

Below is a simple from the scatter plot matrix for all the attributes. The idea here is to quickly locate any good separation between the classes on the scatter plots.

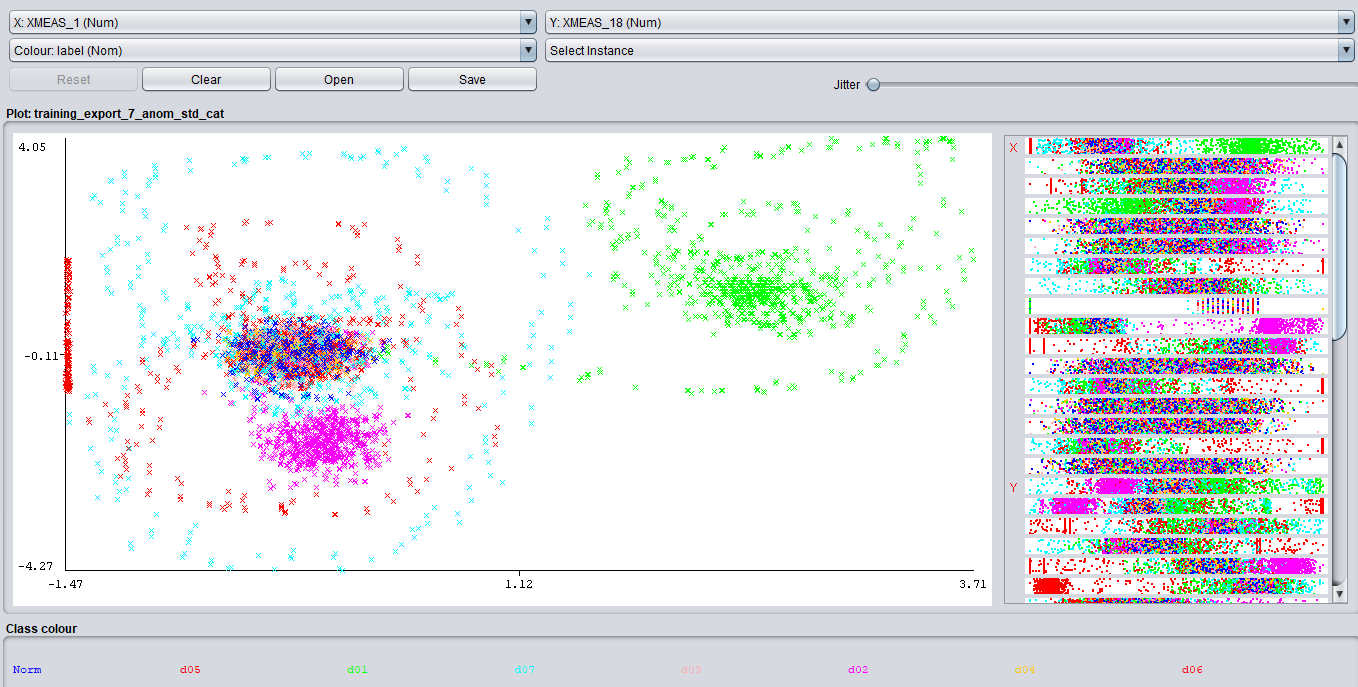
***Attributes Correlations***



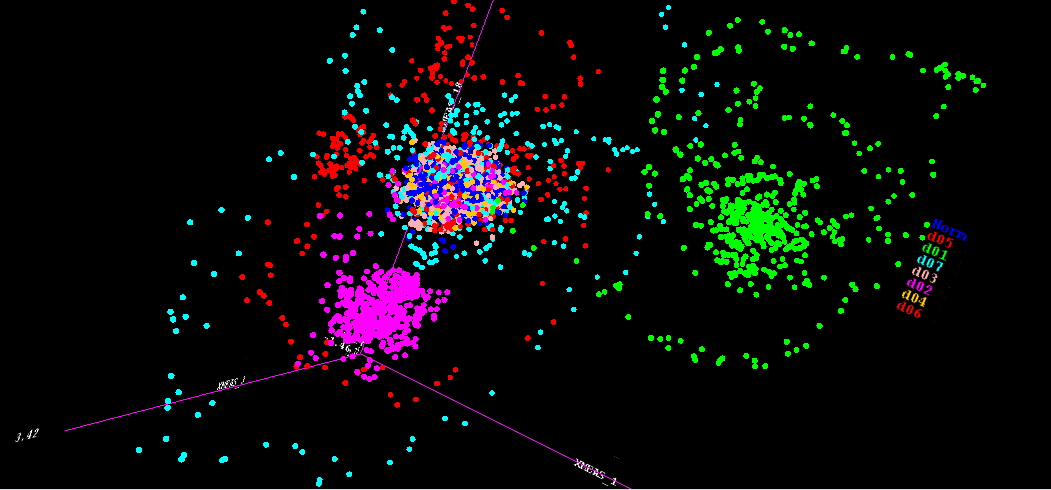
As the number of attributes is big we will take only one of them, i.e. XMEAS\_1 and check its scatter plot against another attribute XMEAS\_18 as shown below.

We can see that we can linearly separate data corresponding to class d01 and d06. It is also possible to separate class d02 by using a non-linear kernel instead of linear, see the 3-D graph. The other classes are not easily separable.

***Scatter Graph***



***3-D Graph***



D06

D02

D01

Table below summarize how easy is the separation for different classes when checking XMEAS\_1 scatter graphs. We see that d01 and d06 are always separable. Norm and d03 almost never been able to be separated which means that the data collected for fault d03 is not different than the data collected for normal operations. Therefore d03 will be hard to classify correctly.

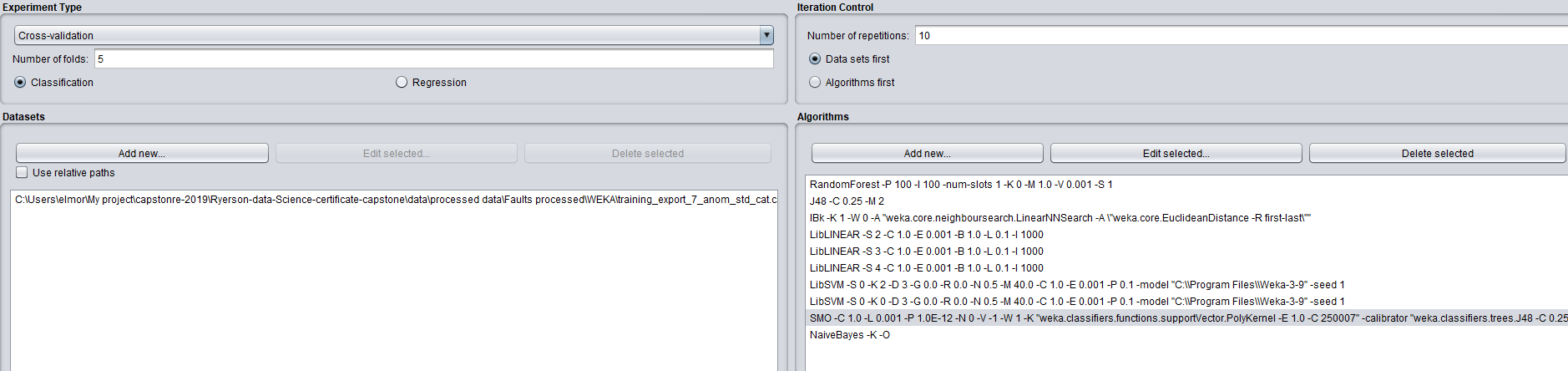
***Scatter plots summary***

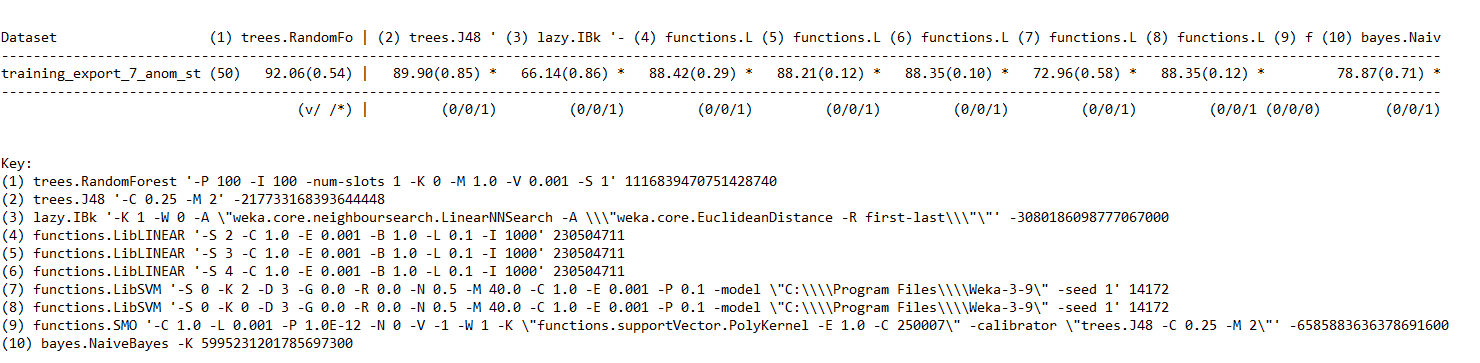
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Correlation/ Class** | **Norm** | **Fault-d01** | **Fault-d02** | **Fault-d03** | **Fault-d04** | **Fault-d05** | **Fault-d06** | **Fault-d07** |
| XMEAS\_1-XMEAS\_2 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_3 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_4 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_5 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_6 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_7 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_8 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_9 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_10 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_11 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_12 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_13 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_14 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_15 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_16 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_17 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_18 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_19 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_20 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_21 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_22 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_23 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_24 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_25 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_26 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_27 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_28 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_29 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_30 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_31 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_32 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_33 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_34 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_35 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_36 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_37 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_38 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_39 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_40 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMEAS\_41 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMV\_1 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMV\_2 |  |  |  |  |  |  |  |  |
| **XMEAS\_1-XMV\_3** | **Correlated** | | | | | | | |
| XMEAS\_1-XMV\_4 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMV\_5 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMV\_6 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMV\_7 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMV\_8 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMV\_9 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMV\_10 |  |  |  |  |  |  |  |  |
| XMEAS\_1-XMV\_11 |  |  |  |  |  |  |  |  |

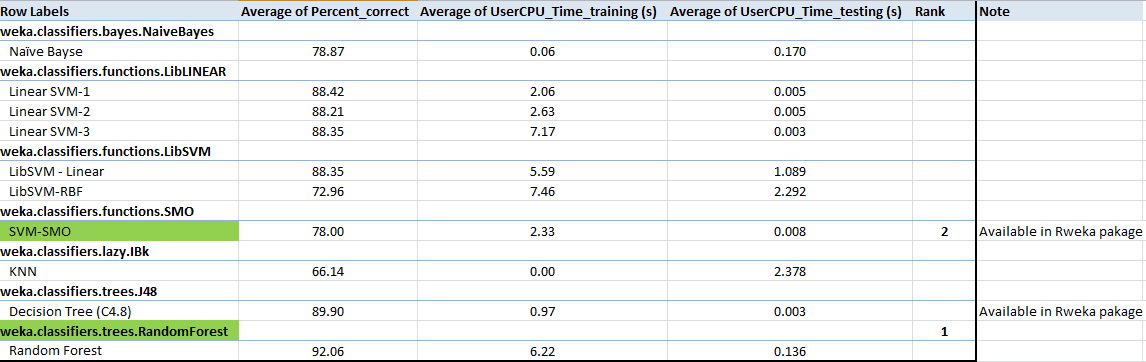
Algorithms evaluation

At this stage a small experiment is designed in Weka using the experimenter to evaluate and pick up the best 3 algorithms. Experiment setup was as following:

* Algorithms default setting is used
* The data set used is a subset (70%) of the original data
* 10 algorithms were picked.
* 10 repetitions with 5 K-folds cross validation are applied
* Main evaluation criteria are:
  + Average model accuracy and its standard deviation.
  + Cost of model evaluation, i.e., CPU time.

Snapshot below shows the configuration

Evaluation results are shown below.

Summary of the main comparison fields and the 2 chosen algorithms is provided in the table below. SVM-SMO is chosen over the other algorithm as it is available in the R package Rweka

Model Selection

## Random Forest

* + Building the model (Intra-Project validation Multiclass)
* Cross validation
* Break ties randomly is selected.
* Multi-classes
* === Summary ===
* Correctly Classified Instances 7482 92.6219 %
* Incorrectly Classified Instances 596 7.3781 %
* Kappa statistic 0.9142
* Mean absolute error 0.0483
* Root mean squared error 0.1286
* Relative absolute error 22.3605 %
* Root relative squared error 39.1209 %
* Total Number of Instances 8078
* === Detailed Accuracy By Class ===
* TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
* 0.993 0.000 1.000 0.993 0.997 0.996 1.000 1.000 d01
* 0.984 0.000 1.000 0.984 0.992 0.991 1.000 1.000 d02
* 0.546 0.014 0.835 0.546 0.661 0.644 0.969 0.807 d03
* 1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 d04
* 0.941 0.005 0.958 0.941 0.949 0.943 0.999 0.993 d05
* 1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 d06
* 0.999 0.000 1.000 0.999 0.999 0.999 1.000 1.000 d07
* 0.942 0.073 0.786 0.942 0.857 0.817 0.980 0.924 Norm
* Weighted Avg. 0.926 0.018 0.929 0.926 0.923 0.911 0.992 0.960
* === Confusion Matrix ===
* a b c d e f g h <-- classified as
* 889 0 0 0 0 0 0 6 | a = d01
* 0 860 0 0 0 0 0 14 | b = d02
* 0 0 502 0 20 0 0 397 | c = d03
* 0 0 0 912 0 0 0 0 | d = d04
* 0 0 13 0 858 0 0 41 | e = d05
* 0 0 0 0 0 892 0 0 | f = d06
* 0 0 0 0 0 0 888 1 | g = d07
* 0 0 86 0 18 0 0 1681 | h = Norm
  + Building the model (Intra-Project validation Binary)

1. === Summary ===
2. Correctly Classified Instances 7408 93.3585 %
3. Incorrectly Classified Instances 527 6.6415 %
4. Kappa statistic 0.8094
5. Mean absolute error 0.1488
6. Root mean squared error 0.2386
7. Relative absolute error 42.1731 %
8. Root relative squared error 56.8074 %
9. Total Number of Instances 7935
10. === Detailed Accuracy By Class ===
11. TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
12. 0.962 0.163 0.952 0.962 0.957 0.810 0.981 0.994 Fault
13. 0.837 0.038 0.868 0.837 0.852 0.810 0.981 0.932 Norm
14. Weighted Avg. 0.934 0.134 0.933 0.934 0.933 0.810 0.981 0.980
15. === Confusion Matrix ===
16. a b <-- classified as
17. 5889 232 | a = Fault
18. 295 1519 | b = Norm
    * Building the model (Cross-Project binary)

For Cross –Project, the same model that is trained for intra-Project Binary classification case will be used.

* + Testing the model (Intra-Project validation)

3642 data measurements were holdout from the original data and set as testing dataset for the models.

What we observe is:

* The accuracy of the model has improved by about **2%** on the testing data
* The true positive, i.e., Fault detection for Fault d03 has improved by about **9.5%**

Based on the above the Random Forest model was able to generalize very well.

=== Summary ===

Correctly Classified Instances 3261 94.1941 %

Incorrectly Classified Instances 201 5.8059 %

Kappa statistic 0.9323

Mean absolute error 0.0446

Root mean squared error 0.1211

Relative absolute error 20.7048 %

Root relative squared error 36.8709 %

Total Number of Instances 3462

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 d01

0.980 0.000 1.000 0.980 0.990 0.989 1.000 1.000 d02

0.562 0.006 0.910 0.562 0.695 0.692 0.976 0.849 d03

1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 d04

0.970 0.005 0.957 0.970 0.964 0.959 0.999 0.994 d05

1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 d06

1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 d07

0.970 0.062 0.824 0.970 0.891 0.860 0.986 0.948 Norm

Weighted Avg. 0.942 0.015 0.946 0.942 0.938 0.930 0.994 0.972

=== Confusion Matrix ===

a b c d e f g h <-- classified as

385 0 0 0 0 0 0 0 | a = d01

0 398 0 0 0 0 0 8 | b = d02

0 0 203 0 9 0 0 149 | c = d03

0 0 0 368 0 0 0 0 | d = d04

0 0 3 0 357 0 0 8 | e = d05

0 0 0 0 0 388 0 0 | f = d06

0 0 0 0 0 0 391 0 | g = d07

0 0 17 0 7 0 0 771 | h = Norm

* + Testing the model (intra-Project Binary)
* === Summary ===
* Correctly Classified Instances 2462 93.0813 %
* Incorrectly Classified Instances 183 6.9187 %
* Kappa statistic 0.8036
* Mean absolute error 0.1496
* Root mean squared error 0.24
* Relative absolute error 42.3417 %
* Root relative squared error 57.1032 %
* Total Number of Instances 2645
* === Detailed Accuracy By Class ===
* TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
* 0.956 0.155 0.954 0.956 0.955 0.804 0.980 0.994 Fault
* 0.845 0.044 0.852 0.845 0.848 0.804 0.980 0.930 Norm
* Weighted Avg. 0.931 0.130 0.931 0.931 0.931 0.804 0.980 0.979
* === Confusion Matrix ===
* a b <-- classified as
* 1950 89 | a = Fault
* 94 512 | b = Norm
  + Testing the model (Cross-Project validation)

the model selected above (Random Forest) will be applied to a new data set that is completely different but has got the same attributes.

* + The new dataset is built from fault data d08 to d12 corresponding to random variables applied to the system.
  + The model is trained on the old dataset but tested on the new dataset. Cross-validation is applied during the training stage.
  + The case with balanced and unbalanced dataset was tested. Both bad prediction of Normal cases. All instances were classified as Fault
  + Accuracy was high but Kappa is Zero which means no agreement between the current and the prediction.

=== Summary ===

Correctly Classified Instances 5600 89.7436 %

Incorrectly Classified Instances 640 10.2564 %

Kappa statistic 0

Mean absolute error 0.1238

Root mean squared error 0.3124

Relative absolute error 67.198 %

Root relative squared error 102.9834 %

Total Number of Instances 6240

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

1.000 1.000 0.897 1.000 0.946 ? 0.586 0.922 Fault

0.000 0.000 ? 0.000 ? ? 0.586 0.119 Norm

Weighted Avg. 0.897 0.897 ? 0.897 ? ? 0.586 0.839

=== Confusion Matrix ===

a b <-- classified as

5600 0 | a = Fault

640 0 | b = Norm

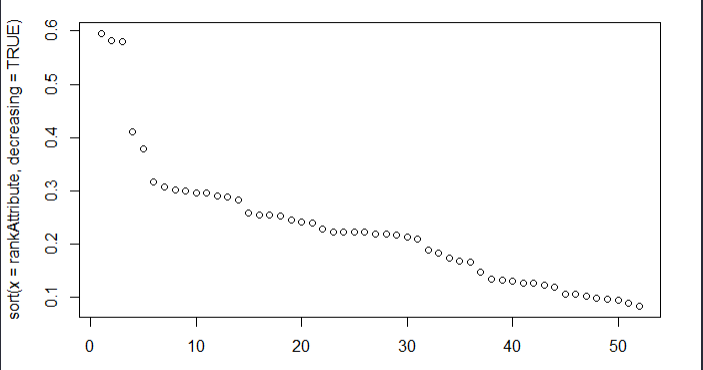
## Support Vector Machine

* Same dataset is used as the Random Forest.
* Using linear Kernel and Decision tree (C4.8) as calibrator
* The algorithm could not detect fault d03.
* Building the model
* === 10 Fold Cross Validation ===
* === Summary ===
* Correctly Classified Instances 7139 88.3758 %
* Incorrectly Classified Instances 939 11.6242 %
* Kappa statistic 0.8636
* Mean absolute error 0.1886
* Root mean squared error 0.293
* Relative absolute error 87.2609 %
* Root relative squared error 89.119 %
* Total Number of Instances 8078
* === Detailed Accuracy By Class ===
* TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
* 0.998 0.000 0.998 0.998 0.998 0.997 1.000 0.997 d01
* 0.983 0.000 1.000 0.983 0.991 0.990 0.995 0.990 d02
* 0.000 0.000 ? 0.000 ? ? 0.882 0.352 d03
* 1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 d04
* 0.997 0.000 1.000 0.997 0.998 0.998 1.000 0.999 d05
* 1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 d06
* 1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 d07
* 1.000 0.149 0.656 1.000 0.792 0.747 0.926 0.656 Norm
* Weighted Avg. 0.884 0.033 ? 0.884 ? ? 0.969 0.849
* === Confusion Matrix ===
* a b c d e f g h <-- classified as
* 893 0 0 0 0 0 0 2 | a = d01
* 0 859 0 0 0 0 0 15 | b = d02
* 0 0 0 0 0 0 0 919 | c = d03
* 0 0 0 912 0 0 0 0 | d = d04
* 2 0 0 0 909 0 0 1 | e = d05
* 0 0 0 0 0 892 0 0 | f = d06
* 0 0 0 0 0 0 889 0 | g = d07
* 0 0 0 0 0 0 0 1785 | h = Norm
  + Testing the model (Intra-Project validation)
* The same Testing dataset, as random Forest, is used
* The accuracy of the model has improved by about 1.**2%** on the testing data

|  |
| --- |
| === Summary ===  Correctly Classified Instances 3090 89.2548 %  Incorrectly Classified Instances 372 10.7452 %  Kappa statistic 0.8734  Mean absolute error 0.1885  Root mean squared error 0.2928  Relative absolute error 87.4378 %  Root relative squared error 89.1838 %  Total Number of Instances 3462  === Detailed Accuracy By Class ===  TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class  0.995 0.000 0.997 0.995 0.996 0.996 0.998 0.996 d01  0.980 0.000 1.000 0.980 0.990 0.989 0.998 0.993 d02  0.000 0.000 ? 0.000 ? ? 0.879 0.325 d03  1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 d04  0.997 0.000 1.000 0.997 0.999 0.998 1.000 0.998 d05  1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 d06  1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 d07  1.000 0.139 0.682 1.000 0.811 0.766 0.930 0.682 Norm  Weighted Avg. 0.893 0.032 ? 0.893 ? ? 0.971 0.855  === Confusion Matrix ===  a b c d e f g h <-- classified as  383 0 0 0 0 0 0 2 | a = d01  0 398 0 0 0 0 0 8 | b = d02  0 0 0 0 0 0 0 361 | c = d03  0 0 0 368 0 0 0 0 | d = d04  1 0 0 0 367 0 0 0 | e = d05  0 0 0 0 0 388 0 0 | f = d06  0 0 0 0 0 0 391 0 | g = d07  0 0 0 0 0 0 0 795 | h = Norm |
|  |
| |  | | --- | | > | |

Feature Selection

* + Attribute selection applied on Training data only
  + Using Rweka’ weka.attributeSelection.ClassifierAttributeEval to evaluate the worth of an attribute, towards the class.
    - Decision tree is used as a classification
    - Cross validation is used to reduce variance (5 folds)
  + Running the code in R we are able to get list of ranked attributes. 15 out of 52 were selected (above the redline) and a new model fitted with the reduced data.



The Model with reduce number of attributes provided much better Accuracy and Kappa score than the model with Full list of features.

## Random Forest

* Building the model (intra-project multiclass)

=== Summary ===

Correctly Classified Instances 7609 94.1941 %

Incorrectly Classified Instances 469 5.8059 %

Kappa statistic 0.9325

Mean absolute error 0.0558

Root mean squared error 0.1376

Relative absolute error 25.8084 %

Root relative squared error 41.8576 %

Total Number of Instances 8078

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.994 0.000 0.999 0.994 0.997 0.996 1.000 1.000 d01

0.986 0.000 0.999 0.986 0.993 0.992 1.000 1.000 d02

0.860 0.008 0.933 0.860 0.895 0.883 0.992 0.955 d03

0.870 0.004 0.961 0.870 0.913 0.904 0.996 0.975 d04

0.830 0.003 0.968 0.830 0.894 0.885 0.993 0.962 d05

1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 d06

0.998 0.000 1.000 0.998 0.999 0.999 1.000 1.000 d07

0.974 0.056 0.831 0.974 0.897 0.869 0.992 0.974 Norm

Weighted Avg. 0.942 0.014 0.947 0.942 0.942 0.932 0.996 0.982

=== Confusion Matrix ===

a b c d e f g h <-- classified as

890 0 0 0 0 0 0 5 | a = d01

0 862 0 0 0 0 0 12 | b = d02

0 0 790 2 7 0 0 120 | c = d03

0 0 9 793 6 0 0 104 | d = d04

0 0 21 22 757 0 0 112 | e = d05

0 0 0 0 0 892 0 0 | f = d06

1 0 0 0 1 0 887 0 | g = d07

0 1 27 8 11 0 0 1738 | h = Norm

* Building the model (intra-project binary Class)
* === Summary ===
* Correctly Classified Instances 7528 94.8708 %
* Incorrectly Classified Instances 407 5.1292 %
* Kappa statistic 0.8498
* Mean absolute error 0.1522
* Root mean squared error 0.2318
* Relative absolute error 43.1538 %
* Root relative squared error 55.1902 %
* Total Number of Instances 7935
* === Detailed Accuracy By Class ===
* TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
* 0.980 0.158 0.955 0.980 0.967 0.851 0.989 0.996 Fault
* 0.842 0.020 0.927 0.842 0.882 0.851 0.989 0.963 Norm
* Weighted Avg. 0.949 0.126 0.948 0.949 0.948 0.851 0.989 0.989
* === Confusion Matrix ===
* a b <-- classified as
* 6000 121 | a = Fault
* 286 1528 | b = Norm
* Testing the model (intra-project multiclass)

=== Summary ===

Correctly Classified Instances 3332 96.2449 %

Incorrectly Classified Instances 130 3.7551 %

Kappa statistic 0.9563

Mean absolute error 0.0495

Root mean squared error 0.1243

Relative absolute error 22.9372 %

Root relative squared error 37.8704 %

Total Number of Instances 3462

=== Detailed Accuracy By Class ===

TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class

0.995 0.000 1.000 0.995 0.997 0.997 1.000 1.000 d01

0.993 0.000 1.000 0.993 0.996 0.996 1.000 1.000 d02

0.873 0.004 0.963 0.873 0.916 0.908 0.996 0.969 d03

0.910 0.004 0.965 0.910 0.937 0.930 0.998 0.986 d04

0.899 0.003 0.971 0.899 0.934 0.927 0.999 0.988 d05

1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 d06

1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 d07

0.989 0.036 0.891 0.989 0.937 0.920 0.997 0.989 Norm

Weighted Avg. 0.962 0.009 0.964 0.962 0.962 0.956 0.998 0.992

=== Confusion Matrix ===

a b c d e f g h <-- classified as

383 0 0 0 0 0 0 2 | a = d01

0 403 0 0 0 0 0 3 | b = d02

0 0 315 3 3 0 0 40 | c = d03

0 0 4 335 2 0 0 27 | d = d04

0 0 6 7 331 0 0 24 | e = d05

0 0 0 0 0 388 0 0 | f = d06

0 0 0 0 0 0 391 0 | g = d07

0 0 2 2 5 0 0 786 | h = Norm

* Testing the model (intra-project Binary class)
* === Summary ===
* Correctly Classified Instances 2564 96.9376 %
* Incorrectly Classified Instances 81 3.0624 %
* Kappa statistic 0.9118
* Mean absolute error 0.1424
* Root mean squared error 0.2157
* Relative absolute error 40.3024 %
* Root relative squared error 51.3172 %
* Total Number of Instances 2645
* === Detailed Accuracy By Class ===
* TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
* 0.987 0.091 0.973 0.987 0.980 0.912 0.995 0.998 Fault
* 0.909 0.013 0.955 0.909 0.932 0.912 0.995 0.983 Norm
* Weighted Avg. 0.969 0.073 0.969 0.969 0.969 0.912 0.995 0.995
* === Confusion Matrix ===
* a b <-- classified as
* 2013 26 | a = Fault
* 55 551 | b = Norm
* Testing the model (Cross-project Binary class)

A very important observation here is by reducing the number of attributes in the new model was able to detect few Normal measure this reduce the false alarm. This in contrast to the case in the paragraph before in which no attribute selection was performed.

In conclusion attribute/ feature selection is essential for a model to be able to generalize to support even a data it never learned

* === Summary ===
* Correctly Classified Instances 5602 89.7756 %
* Incorrectly Classified Instances 638 10.2244 %
* Kappa statistic 0.0081
* Mean absolute error 0.1293
* Root mean squared error 0.3088
* Relative absolute error 70.1936 %
* Root relative squared error 101.7922 %
* Total Number of Instances 6240
* === Detailed Accuracy By Class ===
* TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
* 1.000 0.995 0.898 1.000 0.946 0.054 0.626 0.931 Fault
* 0.005 0.000 0.750 0.005 0.009 0.054 0.626 0.149 Norm
* Weighted Avg. 0.898 0.893 0.883 0.898 0.850 0.054 0.626 0.851
* === Confusion Matrix ===
* a b <-- classified as
* 5599 1 | a = Fault
* 637 3 | b = Norm

## Support Vector Machine

* + Building the Model
* === Summary ===
* Correctly Classified Instances 5370 66.4769 %
* Incorrectly Classified Instances 2708 33.5231 %
* Kappa statistic 0.5953
* Mean absolute error 0.1932
* Root mean squared error 0.3008
* Relative absolute error 89.4191 %
* Root relative squared error 91.5135 %
* Total Number of Instances 8078
* === Detailed Accuracy By Class ===
* TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
* 0.981 0.000 1.000 0.981 0.990 0.989 0.998 0.989 d01
* 0.973 0.000 1.000 0.973 0.986 0.985 0.994 0.982 d02
* 0.000 0.000 0.000 0.000 0.000 -0.007 0.723 0.196 d03
* 0.000 0.000 0.000 0.000 0.000 -0.006 0.680 0.168 d04
* 0.083 0.000 1.000 0.083 0.154 0.273 0.665 0.246 d05
* 1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 d06
* 1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 d07
* 1.000 0.430 0.398 1.000 0.569 0.476 0.785 0.398 Norm
* Weighted Avg. 0.665 0.095 0.640 0.665 0.580 0.571 0.846 0.593
* === Confusion Matrix ===
* a b c d e f g h <-- classified as
* 878 0 0 0 0 0 0 17 | a = d01
* 0 850 0 0 0 0 0 24 | b = d02
* 0 0 0 0 0 0 0 919 | c = d03
* 0 0 0 0 0 0 0 912 | d = d04
* 0 0 3 2 76 0 0 831 | e = d05
* 0 0 0 0 0 892 0 0 | f = d06
* 0 0 0 0 0 0 889 0 | g = d07
* 0 0 0 0 0 0 0 1785 | h = Norm
  + Testing the Model

The SVM is not able to detect 2 faults.

Also the accuracy and the Kappa of the model with reduced attribute is very bad compared to before.

* === Summary ===
* Correctly Classified Instances 2377 68.6597 %
* Incorrectly Classified Instances 1085 31.3403 %
* Kappa statistic 0.6195
* Mean absolute error 0.193
* Root mean squared error 0.3004
* Relative absolute error 89.5169 %
* Root relative squared error 91.4957 %
* Total Number of Instances 3462
* === Detailed Accuracy By Class ===
* TP Rate FP Rate Precision Recall F-Measure MCC ROC Area PRC Area Class
* 0.990 0.000 1.000 0.990 0.995 0.994 0.998 0.993 d01
* 0.968 0.000 1.000 0.968 0.984 0.982 0.994 0.981 d02
* 0.000 0.001 0.000 0.000 0.000 -0.010 0.718 0.175 d03
* 0.000 0.000 ? 0.000 ? ? 0.677 0.163 d04
* 0.079 0.000 1.000 0.079 0.146 0.267 0.639 0.227 d05
* 1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 d06
* 1.000 0.000 1.000 1.000 1.000 1.000 1.000 1.000 d07
* 1.000 0.406 0.424 1.000 0.595 0.502 0.797 0.424 Norm
* Weighted Avg. 0.687 0.093 ? 0.687 ? ? 0.850 0.608
* === Confusion Matrix ===
* a b c d e f g h <-- classified as
* 381 0 0 0 0 0 0 4 | a = d01
* 0 393 0 0 0 0 0 13 | b = d02
* 0 0 0 0 0 0 0 361 | c = d03
* 0 0 0 0 0 0 0 368 | d = d04
* 0 0 3 0 29 0 0 336 | e = d05
* 0 0 0 0 0 388 0 0 | f = d06
* 0 0 0 0 0 0 391 0 | g = d07
* 0 0 0 0 0 0 0 795 | h = Norm

Summary Results & Conclusion

The table above summarises the results we were able to achieve applying two techniques and 2 supervised learning algorithm to the TEP anomaly data. In conclusion we have the following:

* Random-Forest generated the best performance at any level of comparison.
* Using Attribute selection the computing cost of Random Forest has improved by about 20% with almost no sacrifice on the performance metrics
* In binary classification RF performed better compared to multi-class.
* We were able to Cross-project prediction by applying feature selection.

One Important next step will be to study more the Cross-project Defect detection. Random Forest showed the potential to be applied in this domain. Cross-project defect detection presents a potential growth opportunity as there is the need to apply existing ML algorithm to accelerate the detection of new issues that have potentially no data for training.

Appendix

## Process variables

The process contains 41 measured variables and 11 manipulated variables. One variable is ignored (XMV(12)) as it is not manipulated. Table 01, 02 and 03 shows all the variables.

Table 01- Manipulated variables

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable** | **Description** | **Unit** | **Comment** |
| XMV(1) | D Feed Flow (Stream 2) | Kg/H |  |
| XMV(2) | E Feed Flow (Stream 3) | Kg/H |  |
| XMV(3) | A Feed Flow (Stream 1) | kscm/H |  |
| XMV(4) | Total Feed Flow (Stream 4) | kscm/H |  |
| XMV(5) | Compressor Recycle Valve | % |  |
| XMV(6) | Purge Valve (Stream 9) | % |  |
| XMV(7) | Separator Pot Liquid Flow (Stream 10) | m3/H |  |
| XMV(8) | Stripper Liquid Product Flow (Stream 11) | m3/H |  |
| XMV(9) | Stripper Steam Valve | % |  |
| XMV(10) | Reactor Cooling Water Flow | m3/H |  |
| XMV(11) | Condenser Cooling Water Flow | m3/H |  |
| XMV(12) | Agitator Speed |  | Will Not be included as it is not manipulated |

Table 02- Process measurements variables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Block Name** | **Variable** | **Description** | **Units** | **Sampling Interval (min)** |
| Input Feed | XMEAS(1) | A Feed (Stream 1) | kscmh | 3 |
| XMEAS(2) | D Feed (Stream 2) | kg/hr | 3 |
| XMEAS(3) | E Feed (Stream 3) | kg/hr | 3 |
| XMEAS(4) | Total Feed (Stream 4) | kscmh | 3 |
| Reactor | XMEAS(6) | Reactor Feed Rate (Stream 6) | kscmh | 3 |
| XMEAS(7) | Reactor Pressure | kPa gauge | 3 |
| XMEAS(8) | Reactor Level | % | 3 |
| XMEAS(9) | Reactor Temperature | Deg C | 3 |
| Separator | XMEAS(11) | Product Sep Temp | Deg C | 3 |
| XMEAS(12) | Product Sep Level | % | 3 |
| XMEAS(13) | Prod Sep Pressure | kPa gauge | 3 |
| XMEAS(14) | Prod Sep Underflow (Stream 10) | m3/hr | 3 |
| Sripper | XMEAS(15) | Stripper Level | % | 3 |
| XMEAS(16) | Stripper Pressure | kPa gauge | 3 |
| XMEAS(17) | Stripper Underflow (Stream 11) | m3/hr | 3 |
| XMEAS(18) | Stripper Temperature | Deg C | 3 |
| XMEAS(19) | Stripper Steam Flow | kg/hr | 3 |
| Miscellanneous | XMEAS(5) | Recycle Flow (Stream 8) | kscmh | 3 |
| XMEAS(10) | Purge Rate (Stream 9) | kscmh | 3 |
| XMEAS(20) | Compressor Work | kW | 3 |
| XMEAS(21) | Reactor Cooling Water Outlet Temp | Deg C | 3 |
| XMEAS(22) | Separator Cooling Water Outlet Temp | Deg C | 3 |

Table 03- Composite measurements variables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Block Name** | **Variable** | **Description** | **Stream** | **Sampling Interval (min)** |
| Reactor Feed Analysis | XMEAS(23) | Component A | 6 | 6 |
| XMEAS(24) | Component B | 6 | 6 |
| XMEAS(25) | Component C | 6 | 6 |
| XMEAS(26) | Component D | 6 | 6 |
| XMEAS(27) | Component E | 6 | 6 |
| XMEAS(28) | Component F | 6 | 6 |
| Purge Gas Analysis | XMEAS(29) | Component A | 9 | 6 |
| XMEAS(30) | Component B | 9 | 6 |
| XMEAS(31) | Component C | 9 | 6 |
| XMEAS(32) | Component D | 9 | 6 |
| XMEAS(33) | Component E | 9 | 6 |
| XMEAS(34) | Component F | 9 | 6 |
| XMEAS(35) | Component G | 9 | 6 |
| XMEAS(36) | Component H | 9 | 6 |
| Product Analysis | XMEAS(37) | Component D | 11 | 15 |
| XMEAS(38) | Component E | 11 | 15 |
| XMEAS(39) | Component F | 11 | 15 |
| XMEAS(40) | Component G | 11 | 15 |
| XMEAS(41) | Component H | 11 | 15 |

## Process Faults

The TEP simulation contains 21 pre-programmed faults (table 04). Sixteen of these faults are known and five are unknown.

Table 04 – Process faults

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fault number** | **Process variable** | **Type** | **+ Gaussain Noise** | **comments** |
| IDV(0) | Normal |  |  | Normal operation |
| IDV(1) | A/C feed ratio, B composition constant | Step | Yes |  |
| IDV(2) | B composition, A/C ration constant | Step | Yes |  |
| IDV(3) | D feed temperature | Step | Yes | Hard detection. No observable changes in mean, variance or peak time [7]. Will be ignored initially. |
| IDV(4) | Reactor cooling water inlet temperature | Step | Yes |  |
| IDV(5) | Condenser cooling water inlet temperature | Step | Yes |  |
| IDV(6) | A feed loss | Step | Yes |  |
| IDV(7) | C header pressure loss-reduced availability | Step | Yes |  |
| IDV(8) | A, B, and C feed composition | Random variable | Yes |  |
| IDV(9) | D feed temperature | Random variable | Yes | Hard detection. No observable changes in mean, variance or peak time [7]. Will be ignored initially. |
| IDV(10) | C feed temperature | Random variable | Yes |  |
| IDV(11) | Reactor cooling water inlet temperature | Random variable | Yes |  |
| IDV(12) | Condenser cooling water inlet temperature | Random variable | Yes |  |
| IDV(13) | Reaction kinetics | Slow drift | Yes |  |
| IDV(14) | Reactor cooling water valve | Sticking | Yes |  |
| IDV(15) | Condenser cooling water valve | Sticking | Yes | Hard detection. No observable changes in mean, variance or peak time [7]. Will be ignored initially. |
| IDV(16) | Unknown | Unknown | Yes |  |
| IDV(17) | Unknown | Unknown | Yes |  |
| IDV(18) | Unknown | Unknown | Yes |  |
| IDV(19) | Unknown | Unknown | Yes |  |
| IDV(20) | Unknown | Unknown | Yes |  |
| IDV(21) | The valve from (4) fixed at steady state position | Constant Position | Yes | Hard detection. No observable changes in mean, variance or peak time [7]. Will be ignored initially. |

## References

[1] Shen Yin et al. A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process. Journal of Process Control 22 (2012) 1567– 1581

[2] <https://catalog.data.gov/dataset/985c920e-acd8-453d-a75f-b7cb8292371e>

[3] J. J. Downs and E. F. Vogel. A plant-wide industrial-process control problem. *Compo fj Chem. Eng.,* 17:245-255, 1993.

[4] Evan Russel, L. H. Chiang, R.D. Braatz. Data-Driven Methods for fault Detection and Diagnosis in Chemical Process. Springer-Verlag London 2000.

[5] WASHINGTON, U. <Http://depts.washington.edu/control/LARRY/TE/download.html>..

[6] dataset link : <http://web.mit.edu/braatzgroup/links.html>

[7] Y.W. Zhang, “Enhanced statistical analysis of nonlinear process using KPCA, KICA and SVM,” *Chemical Engineering Science*, vol. 64, pp. 800–801, 2009

[8] M. Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki, J. Schröder, Diagnosis and Fault-Tolerant Control, Springer-Verlag, Berlin, 2006.

[9]  Landis, J.R.; Koch, G.G. (1977). "The measurement of observer agreement for categorical data". Biometrics. **33** (1): 159–174. [*doi*](https://en.wikipedia.org/wiki/Digital_object_identifier):[*10.2307/2529310*](https://doi.org/10.2307%2F2529310). [*JSTOR*](https://en.wikipedia.org/wiki/JSTOR) [*2529310*](https://www.jstor.org/stable/2529310). [*PMID*](https://en.wikipedia.org/wiki/PubMed_Identifier) [*843571*](https://www.ncbi.nlm.nih.gov/pubmed/843571).

[10] MJaechang Nam, Wei Fu, Sunghun Kim, Heterpgeneous Defect Prediction. IEEE TRANS SE. SUBMITTED JAN‘16, REVISION#?, APR‘16