Determination of power engineering equipment's defects in predictive analytic system using machine learning algorithms

I A Shcherbatov¹ and G N Turikov¹,²,³

¹ National Research University "MPEI", Russia, 111250 Moscow, Krasnokazarmennaya, 14
² National Research University "MPEI", Russia, 111250 Moscow, Krasnokazarmennaya, 14, Grant № 313ГУЦЭС8-D3/56383

TsurikovGN@mpei.ru

Abstract. An approach to forecasting of power equipment's defects and failures is considered in this paper. The main operations of predictive analytic system for forecasting turbine's regulation system's defects is represented. Special attention on machine learning models tuning for explored forecast problem is paid. The following machine learning algorithms: Logistic Regression, Random Forest, Extreme Gradient Boosting, ensembles of these algorithms are explored. According to the results of optimal machine learning models determined, their comparison is done and the conclusion on the appropriateness of the use in predictive analytic system is made.

1. Introduction

Organization of repair operations of power equipment according to its monitored condition is actual problem in modern power industry. While power equipment repairs planning, technical evaluation as part of this approach is required [1]. Moreover, forecasting of power equipment's technical condition is also actual problem. Making an accurate forecast of possible equipment's defect or its transition is nonoperable state will allow to make timely response and to plan the appropriate activities and resources (financial, human, materials and machines) for this equipment normal conditioning or for its timely reconditioning [2].

Predictive analytic systems (PAS) allows to rearrange and timely host events for normal operable condition of power equipment ensuring. These systems work with Big Data and use machine learning (ML) algorithms for making an accurate forecasts. Working theory of PAS used for equipment's defects, distresses and failures predictions is as follows: unstructured information about technological parameters, time events, signalizations and lockouts goes to the datastore from the SCADA systems. At the next step data preprocessing, determination of outliers and anomalies, gaps filling is made - and the information becomes structurized. Then preprocessed data enters to the forecast model's input where classification problem using ML algorithms is being solved. Forecast model's output is not a discrete result (1 - "equipment's defect" / 0 - "no defects"), but the probability that this technical condition of the equipment belongs to a defect state (class "1"). Further, after optimal classification threshold determination, technical state that class "1" membership probability is more than optimal threshold value classifies as a state close to defect.
One more variant of PAS realization is that: technological parameters data goes directly to the special PAS module (FactoryTalk Analytics LogixAI) [3] where optimal machine learning model automatically determined and the forecast to the process control systems operator is being represented.

In this paper main stages of PAS working and classification problem of HPP (hydro power plant) turbine control system's power equipment defects are represented using ML algorithms implemented in Python 3.5.

2. Problem definition

It is required to make a forecast of turbine control system's defect: "drainage obstruction". Current technical state of the explored system is described by 10 numerical features and 3 binary features (features name are represented in the tables below). Features observations for the period from 10.07.2017 to 17.08.2017 with the different interrogations rates of each parameter as an archive from HPP SCADA system are received.

Technical state of explored system is corresponded to 2 classes: "1" - "defect is found", 0 - "no defect found". In researched classification problem to make a prediction of probability that this technical condition of the equipment belongs to a defect state (class "1") is required, thus predicting of possible defect appearance. As a mathematical tools the following ML algorithms are used: Logistic Regression, Random Forest, Extreme Gradient Boosting and ensembles of represented algorithms.

As an objective function that minimal value on train sample is corresponded to optimal ML model is log loss function (cross-entropy). ML models comparison on test sample using precision and recall metrics is done. The maximal value of recall in order to prevent model's false prediction about possible defects is required. The maximal value of precision is also is desirable to receive.

The following actions sequence for explored problem solving is used: data preprocessing (data per minute averaging, data synchronization, gaps filling, features scaling); ML models training (data on train and test samples splitting, Logistic Regression, Random Forest, Extreme Gradient Boosting models training using cross-validation, models hyperparameters optimization); ML models quality determination (calculation of precision, recall, F1-score, confusion matrix determination, ROC-curve building, optimal threshold calculation).

3. Data preprocessing

As mentioned, 10 numerical and 2 binary features in the explored classification problem are being investigated. Number of each parameter's observations depends on interrogation rate of these parameters. Because of different interrogations rates of each parameter (e.g. interrogation rate for water level on turbine's cover is 53-54 times per minute, but for oil pressure in LP oil-pressure unit is 1 time per 2-3 minutes), the problem of data synchronization by time is actual. For solving this problem, firstly, every feature's observation was averaged per minute. The feature with the maximal number of averaged observations was determined as a gage feature on the next step. Finally, every feature's observation was aligned by time with the reference observation.

Thus, on the next step the problem of data gaps become actual. Filling the gaps using linear interpolation and extrapolation was chosen as an approach for solving this problem. As a result the shape of complete features matrix was 12 features × 37439 observations. The complete formation of this matrix allowed to build synchronized trends of explored parameters. One of them, named water's level on turbine's cover is illustrated on the figure 1. Analyzing this trend and the fact that defect: "drainage obstruction" was discovered on August 10th - allowed to make a visual conclusion that water's level on turbine's cover influences on explored defect's registration.

One more important step for receiving models with a good quality is features scaling. The main reason by which features scaling required - is necessity in the equal features order for correct functioning of
machine learning algorithms (especially for linear models where feature weights are being used). In the explored classification problem features normalization as a method of scaling is used.

![Figure 1](image.png)

**Figure 1.** Trend of the parameter: water's level on turbine's cover.

Also, main aspect before ML models training is correlation coefficients calculation. The correlation coefficients in explored classification problem between features and target data (defects) using Pearson method are calculated and their values in table 1 are represented.

**Table 1.** Correlation coefficients between features and target data

| Feature's name                               | Correlation coefficient | Feature's name                               | Correlation coefficient |
|----------------------------------------------|-------------------------|----------------------------------------------|-------------------------|
| Water's level on turbine's cover             | 0.873381                | Oil level in the drain tank of LP OPU         | -0.173015               |
| Water's pressure                             | 0.253708                | Oil level in the leakage tank                 | -0.009348               |
| Oil temperature in the drain tank of HP OPU  | -0.085357               | HP OPU pump №1 ON                            | -0.091616               |
| Oil pressure in the HP OPU boiler (sensor 1) | -0.292796               | HP OPU pump №2 ON                            | -0.091590               |
| Oil pressure in the HP OPU boiler (sensor 2) | -0.293886               | Emergency low pressure in LP OPU accumulator  | 0.002289                |
|                                             |                         | signalization                                |                         |
| Oil level in the drain tank of HP OPU        | 0.309959                | Oil level in the LP OPU boiler                | 0.397182                |
| Oil pressure in the LP OPU boiler            | -0.257557               |                                              |                         |

As analyzed, water's level on turbine's cover makes the most significant influence in defects diagnosis. All binary and some numerical features don't make any contribution in that diagnosis - these features after using Ridge and Lasso regularization will be deleted.

### 4. Machine learning models training

#### 4.1. Splitting data on train and test samples

As noted, the number of observation in preprocessed dataset is 37439. Among these observations only 2448 to class "1" are matched - that means 6.5% regarding the whole number of observations. Splitting
data to train and test samples in 70% to 30% ratio was made using stratification because of necessity for the same classes percentage ratio on train and test samples. Thus, the number of observations on the train sample is 26207, with 1704 belonging to class "1", and 11232 on the test sample, with 734 observations belonging to positive class.

As seen, the problem of class disbalance is actual and data oversampling is required. For solving this problem no data changes were made, but while model training error was calculated, the class balancing weights, \( w_j \), as in the formula for log loss calculation below are added.

\[
Q(w, X) = \sum_{i=1}^{I} \ln(1 + \exp(w_j \times w^T x_i))
\]  

**4.2. Logistic Regression model**

As noted, the required result after prediction's making is the class "1" (defects) membership probability, that can be converted to a binary answer after optimal threshold foundation. I.e. the result that we firstly need is

\[
\pi(x) = P(y = 1 \mid x)
\]  

In Logistic Regression [4] model this probability using sigmoid function is calculated:

\[
\pi(x) = \frac{\exp(<w, x>)}{1 + \exp(<w, x>)},
\]  

where \(< w, x >\) - scalar product of features weights, \( w_j \), and features, \( x^{(j)} \).

The tuning of this model on the train sample by maximal likelihood method is done. The following modifications (with L1-, L2- regularization) of log loss function (2) that minimal value while optimal feature's j weights, \( w_j \), it is necessary to find to receive the best model, are used.

\[
Q(w, X) = \frac{1}{2} w^T w + C \sum_{i=1}^{I} \ln(1 + \exp(-y_j \times w^T x_i)) \rightarrow \min_w ,
\]  

or with L1-regularization:

\[
Q(w, X) = \sum_{j=1}^{J} |w_j| + C \sum_{i=1}^{I} \ln(1 + \exp(-y_j \times w^T x_i)) \rightarrow \min_w ,
\]  

where - model hyperparameter that shows, how much we allow the model to be attuned to data.

Thus, to make an optimal Logistic Regression model - determination of optimal regularization method, hyperparameter, \( C \), and features weight, \( w_j \) is required.

In this exploration optimization problems (5) and (6) were solved using grid search and stratified K-fold cross-validation strategy for finding optimal regularization method and model hyperparameter, \( C \). Parameter \( C \) was changed between 0.01 and 100, Lasso and Ridge [5] regularization methods were used. As a result the optimal Logistic Regression model with Ridge regularization method and hyperparameter \( C = 100 \) was received using limited memory Broyden-Fletcher-Goldfarb-Shanno algorithm (LBFGS) [6] that is the part of quasi-Newton family algorithms.

Motion of log loss function's value according to hyperparameter \( C \) (while model tuning) is shown at the figure 2. Represent Logistic Regression algorithm's quality on test sample: building ROC-curve (while algorithm is ideal - curve passes through the (0,1) coordinate) - figure 3, PR-curve (while algorithm is ideal - curve passes through the (1,1) coordinate) - figure 4, actual-predicted plot (figure 5). Algorithm's quality metrics including log loss function value, precision, recall, number of FN (False Negative) and FP (False Positive) mistakes are represented at the table 2.
Figure 2. Log loss motion according to parameter C (LFBGS-method).

Figure 3. ROC-curves on test and train samples.

Figure 4. PR-curves on test and train samples.

Table 2. Algorithm's quality metrics.

| Metrics                                           | Value         | Metrics                                           | Value         |
|---------------------------------------------------|---------------|---------------------------------------------------|---------------|
| Log loss on train sample                          | 0.0006270288  | Optimal threshold                                 | 0.613599      |
| Log loss on test sample                           | 0.0011196107  | FN (after optimal threshold determination) on test sample | 0             |
| Recall (after optimal threshold determination) on test sample | 1.0           | FP (after optimal threshold determination) on test sample | 5             |
| Precision (after optimal threshold determination) on test sample | 0.999523719  |                                                  |               |
The optimal threshold was calculated using following considerations: while bounds on FN is that FN = 0 (because, as mentioned before, maximal recall is required), the minimal value of FP should be found (because of maximal precision is desirable to receive).

One more important Logistic Regression model's result is features after regularization weight's values. It is worth noting that optimal regularization method for solving explored classification problem is Ridge regularization. While using this regularization method, features weights, $w_j$, are being decreased if features don't make any contribution in defects diagnosis, but are not being deleted as in the Lasso regularization method. Thus, after optimal feature's after Ridge regularization weights calculation, weights that values are smaller than 0.005 were deleted from the dataset.

4.3. Random Forest model

Analyzing actual-predicted plot built by Logistic Regression model, we can conclude that this ML model in explored classification problem avoids FN mistakes is not risky. Therefore, it is necessary to try more risky algorithm - ensemble of decision trees - Random Forest [7].

For explored classification problem the best Random Forest model using log loss function (4) (instead $< w ,x >$ - model's answers $a(x) = \frac{1}{N} \sum_{n=1}^{N} b_n(x)$ used, where $b_n(x)$ - base algorithms ) was found. To receive the best model, following optimal model's parameters using grid search were determined: optimal tree's number, optimal tree's maximal depth, minimal number of samples for splitting each node, minimal number of samples in every decision tree's leaf. To receive the optimal value of each parameter separate iterations of grid search were produced. After each iteration completing the number of base algorithms required for minimum log loss function's value finding were re-calibrated. Log loss function's motion depending on trees number while cross-validation of the train sample - at the figure 6 is shown. ROC-, PR-curves have the absolutely same behaviors on train and test samples as when Logistic Regression model explored and therefore, it is not necessary for these curves demonstration. Actual-predicted plot with optimal threshold founded at the figure 7 is shown. Algorithm's quality metrics at the table 3 are represented.
Figure 6. Log loss function depending on trees number.

Figure 7. Actual-predicted plot with optimal threshold founded.

Table 3. Algorithm’s quality metrics.

| Metrics                                 | Value         | Metrics                               | Value         |
|-----------------------------------------|---------------|---------------------------------------|---------------|
| Log loss on train sample                | $7.816 \times 10^{-5}$ | Optimal threshold                     | 0.0665        |
| Log loss on test sample                 | 0.000364      | FN (after optimal threshold determination) on test sample | 0             |
| Recall (after optimal threshold determination) on test sample | 1.0            | FP (after optimal threshold determination) on test sample | 5             |
| Precision (after optimal threshold determination) on test sample | 0.999523719   |                                            |               |

As seen, Random Forest optimal model's log loss function is smaller than log loss of Logistic Regression model, but the main for explored problem's metrics on test sample: recall and precision are the same, the number of FN and FP errors is the same too. Wherein, as from actual-predicted plot
(figure 7) seen, Random Forest model has much more risky behavior while class "1" membership probabilities predicting - this is the reason of too small value of the optimal threshold founded.

### 4.4. Extreme gradient boosting (XGBoost) model

One more explored model for solving the problem in hand - is wide-used in ML: XGBoost [8] model. It is also an ensemble of decision trees algorithms, wherein the building of the base algorithms is not parallel as in Random Forest model, but consistent. The every next base algorithm, $b(x_i)$, to correct errors of already existing composition, $a_{N-1}(x)$, is being built.

For explored classification problem the best XGBoost model using log loss function (4) was found. To receive the best model, following optimal model's parameters using grid search were determined: optimal number of base algorithms (decision trees), optimal maximal tree's depth, optimal minimal number of observations in tree's leaf (min_child_weight), optimal minimum loss reduction required to make a split (gamma), optimal number of observations randomly taken for tree studying (subsample), optimal number of features randomly taken for tree studying (colsample_bytree), regularization parameters (alpha and lambda). To receive the optimal value of each parameter separate iterations of grid search were produced. After each iteration completing the number of base algorithms required for minimum log loss function's value finding were re-calibrated. As the result, the next optimal parameters (table 4) were received. The relationship log loss - number of decision trees is at the figure 8 is represented.

**Table 4. Optimal parameters values of XGBoost model.**

| Optimal parameters                  | Value |
|------------------------------------|-------|
| Number of decision trees           | 327   |
| Maximal tree depth                 | 4     |
| min_child_weight                   | 1     |
| gamma                              | 0     |
| subsample                          | 1.0   |
| subsample                          | 0.16  |
| Regularization parameter (lambda)  | 2.0   |
| Regularization parameter (alpha)   | 0.13  |

**Figure 8.** Log loss function depending on trees number.
Demonstrate the result of XGBoost model's working on the test sample. ROC-, PR-curves have the absolutely same behaviors on train and test samples as when Logistic Regression and Random Forest model explored and therefore, it is not necessary for these curves demonstration. Actual-predicted plot with optimal threshold founded at the figure 9 is shown. General metrics at the table 5 are represented.

**Table 5. Algorithm's quality metrics.**

| Metrics                                      | Value           | Metrics                                      | Value           |
|----------------------------------------------|-----------------|----------------------------------------------|-----------------|
| Log loss on train sample                    | $7.594 \times 10^{-5}$ | Optimal threshold                           | 0.0295          |
| Log loss on test sample                     | 0.000404        | FN (after optimal threshold determination) on test sample | 0               |
| Recall (after optimal threshold determination) on test sample | 1.0             | FP (after optimal threshold determination) on test sample | 3               |
| Precision (after optimal threshold determination) on test sample | 0.999714        |                                              |                 |

![Figure 9. Actual-predicted plot with optimal threshold founded.](image)

As analyzed, the XGBoost model results on the test sample is a little bit worse than Random Forest model's results. This conclusion is only for log loss on test sample, but not for precision - the number of FP errors became less when using XGBoost model.

4.5. Ensemble of explored algorithms

Analyzing explored models results, it should be noted that all the results are good enough, but they can be improved using the compositions (ensembles) of these models. I.e. it is necessary to combine two models: the model that avoids FN and is not risky (Logistic Regression) with risky model (Random Forest or XGBoost). The ensemble model in explored problem works using vote [9], i.e. to each model weight is being assigned and the resulting probability predictions with considering of the model weights is being made.

To receive the best ensemble model in explored problem - the following optimization problem was solved: while bound on FN errors is that FN = 0 (because, as mentioned before, maximal recall was required), the optimal models voting weights that corresponded to minimal value of FP (because of
maximal precision was desirable to receive) were found. As a result, it was determined, that the best model combination (with FN = 0 and FP = 1) is ensemble of Logistic Regression and Random Forest models with the weights: 2.5 and 8 respectively. Actual-predicted plot with optimal threshold founded at the figure 10 is shown.

![Actual-predicted plot with optimal threshold found.](image)

5. Models quality comparison

Represent a summary table (table 6) of explored models quality, where the model quality's parameters are: the number of FN and FP errors on the test sample after optimal threshold determination.

| Optimal model (weights respectively) | FN | FP  | Optimal threshold |
|--------------------------------------|----|-----|-------------------|
| Logistic Regression                  | 0  | 5   | 0.6136            |
| Random Forest                        | 0  | 5   | 0.0665            |
| XGBoost                              | 0  | 3   | 0.0295            |
| Logistic Regression & Random Forest (2.5, 8) | 0  | 1   | 0.2               |
| Logistic Regression & XGBoost (1, 6)  | 0  | 2   | 0.1190            |

As seen, the most suitable model for the problem of class "1" (defect: "drainage obstruction") membership probability predictions solution is the ensemble of Logistic Regression and Random Forest models. Using this model it become possible to make a forecast of power equipment defect: "drainage obstruction" for technical state of this equipment - with the high level of recall (1.0) and precision (0.9999) metrics.

6. Conclusion

The full list of working predictive analytic system's operations for predicting turbine's regulation system's defect: "drainage obstruction" is represented. The tuning of optimal machine learning models for making defect's predictions is shown. As analyzed, the optimal ML model (in explored problem: ensemble of Logistic Regression and Random Forest) allows to make very accurate defect's forecast, that can be used for making operating reaction and power equipment's timely reconditioning.
References

[1] Jashura A I 2006 Power Equipment Maintenance and Repair System (Moscow: National Center ENAS) 504 p

[2] Arakelian E K, Pashchenko A F, Shcherbatov I A, Tsurikov G N, Titov F M 2019 Creation of Predictive Analytics System for Power Energy Objects Proc. of 2018 11th Int. Conf. MLSD 2019 5 p (Moscow: National Center ENAS) 504 p

[3] Tropin A 2020 LogixAI - Analytic Solution on Rockwell Automation ControlLogix platform [Electronic resource] Access mode: https://www.klinkmann.ru/about/events/rockwell-automation-webinars-2020/

[4] Murphy K 2012 Machine Learning A Probabilistic Perspective (Massachusetts: Massachusetts Institute of Technology) 1098 p

[5] Tibshirani R 1996 Regression Shrinkage and Selection via LASSO J. of the Royal Statistical Society V 58 I 1 pp 267–288

[6] Raghu Bollapragada, Jorge Nocedal, Dheevatsa Mudigere, Hao-Jun Shi, Ping Tak Peter Tang A progressive batching l-bfgs method for machine learning Proc. of the 35th Int. Conference on Machine Learning, 2018 pp 620-629

[7] Hastie T, Tibshirani R, Friedman J 2009 The elements of statistical learning. Second Edition (Springer Series in Statistics) 745 p

[8] Tianqi Chen, Carlos Guestrin XGBoost: A Scalable Tree Boosting System Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, 2016 pp 785-794

[9] Aishwarya Singh 2018 A Comprehensive Guide to Ensemble Learning (with Python codes) [Electronic resource] Access mode: https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/