

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

5,300

Open access books available

130,000

International authors and editors

155M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Machine Learning Models for Industrial Applications

*Enislay Ramentol, Tomas Olsson and Shaibal Barua*

## Abstract

More and more industries are aspiring to achieve a successful production using the known artificial intelligence. Machine learning (ML) stands as a powerful tool for making very accurate predictions, concept classification, intelligent control, maintenance predictions, and even fault and anomaly detection in real time. The use of machine learning models in industry means an increase in efficiency: energy savings, human resources efficiency, increase in product quality, decrease in environmental pollution, and many other advantages. In this chapter, we will present two industrial applications of machine learning. In all cases we achieve interesting results that in practice can be translated as an increase in production efficiency. The solutions described cover areas such as prediction of production quality in an oil and gas refinery and predictive maintenance for micro gas turbines. The results of the experiments carried out show the viability of the solutions.

**Keywords:** machine learning, prediction, regression methods, maintenance, degradation prediction

## 1. Introduction

The amount of data accumulated by man's activity is uncountable. Millions of tuples are registered daily in the databases, each of them constitutes an observation, an experience to learn from, and a situation that could reoccur in the future in a similar way. Learning from experience is something that humans do naturally and constantly, but what happens if the number of experiences exceeds our ability to process it? What happens if a fact is repeated millions of times and never happens again in exactly the same way?

Machine learning (ML) is the area of artificial intelligence, which deals with learning from the experience, that is, to extract automatically implicit knowledge in the information (stored in the form of data) [1].

In this paper we will describe two real-world industry problems that have been solved using ML. The first of these consists in predicting the quality of the final products of an oil and gas refinery, described in Section 2. The second consists of a model for estimating the degradation of a fleet of micro gas turbines, described in Section 3.

In the next section we offer the theoretical elements necessary for the development of our solutions. Any interested reader can find in Section 1.1 the description of the ML methods we have used. We also describe the general working scheme of the ML applications.

## 1.1 Machine learning

There are countless examples of complex real problems solved through ML, such as [2–6]. In ML, one important subarea is inductive learning; this type of method assumes that a set of examples or instances is known [7]. Formally, **learning** is defined as:

**Theorem 1.1.** Let a set of examples  $(x_i, y_i)$ ,  $x_i \in D$  be a domain space state  $D$  and  $y_i \in S$  be a solution domain space state  $S$ , or, let  $(x_i)$ ,  $i = 1, 2, \dots, n$ , be a domain space state  $D$  where the solutions are not defined, that is,  $S$  is undefined. The task of creating a system that can learn the input-output pairs  $\{(x, y)\}$  or learn the characteristics inherent to  $\{x\}$  is defined as learning.

The first case refers to **supervised learning**, where there is a solution  $y_i$  (the class label) for each input vector  $x_i$ , these examples are known as “classified” or “labeled” [8]. The second case refers to **unsupervised learning**, in which a system learns characteristics, traits, groups, and concepts from unlabeled data.

The supervised learning is a technique to deduce a function from training data. One component of the pair is the input data and the other, the desired results. The output of the function can be a numerical value (as in the regression problems) or a class label (as in the classification ones).

Formally **supervised learning** is defined as:

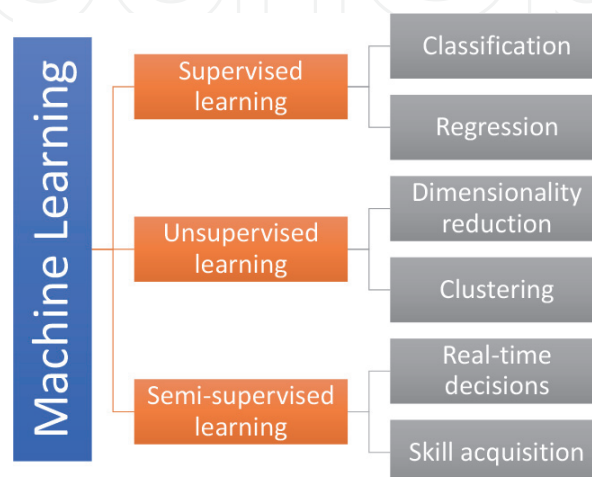
**Theorem 1.2.** Let  $T$  a training set, which is formed by pairs  $(x_i, y_i)$ ,  $i = 1 \dots n$ , where  $n$  represents the number of features,  $x_i$  is defined as input vector,  $y_i$  is the output value (the target). If,  $y_i$  is numeric then it is a regression task, and if,  $y_i$  is discrete then it is a classification task.

The need for supervised learning arises from the requirements of having an automated procedure that is much faster than a human supervisor and that, at the same time, can avoid biases and prejudices adopted by an expert [9].

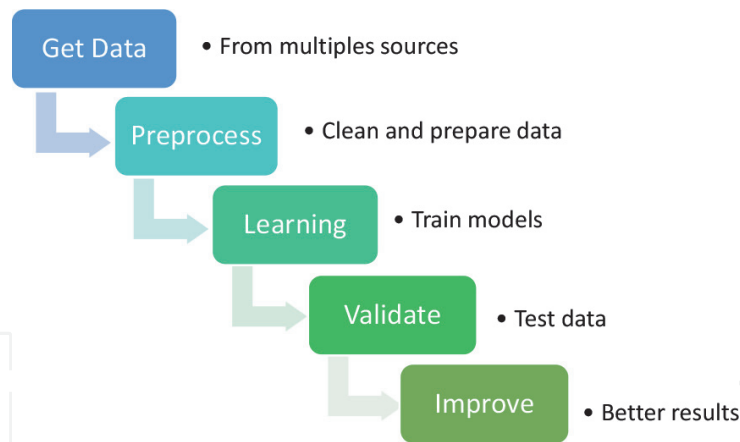
There is also another area in ML known as semi-supervised learning (SSL) [10, 11]. SSL uses both labeled and unlabeled data for training. Reinforcement learning (RL) is an example of SSL. In RL, the model learns how to act in changing environments. It is about taking suitable action to maximize reward in a particular situation. It has been widely used in games, autonomous driving, and many industrial applications. **Figure 1** summarizes the previous definitions.

However, to get to the learning process, it is necessary to go through some preparing phases first. **Figure 2** shows these phases.

The first phase is the data collection; the data can come from multiple sources, be in different formats, etc. The second phase is the data preprocessing; in this



**Figure 1.**  
*Machine learning tasks.*



**Figure 2.**  
*Learning phases.*

phase numerous tasks are performed in order to prepare the data for the learning stage. These tasks can remove noise; normalize, discretize (is needed for the learning phase), and remove/replace missing values; select features; and select instances. When the data is ready, the learning phase can start. The data is partitioned in:

**Training set:** is the set of examples used to build the model to train the ML model.

**Testing set:** is the set of examples used to test the models. The ML model will assign an output to each example in the test set. In classification, if the output value assigned by the ML model matches the label that has the example in the test set, then it is true classified. In regression, an error is computed using the difference between the real value and the predicted. **Figure 2** shows the phases described above.

Given the relevance of preprocessing to our study in the following subsection, we will describe in detail some of the preprocessing techniques.

## 1.2 Preprocessing steps

In real-world applications, especially the industrial ones, data is rarely clean and homogeneous. Most often we find data that tends to be incomplete, redundant, noisy, or inconsistent. The area of ML that deals with the above problems is known as preprocessing. The preprocessing task consists of the set of techniques that are carried out before the learning process. Its objective is to obtain a higher-quality dataset. These techniques can be divided in the following groups:

1. Handling missing values: missing values occur for various reasons: human errors, errors in sensor measurements, data is merged from various sources, etc. Some learning methods can deal with missing values internally, while others do not. The most common techniques to deal with missing values are:
  - a. Remove the variables or remove the examples with missing values. This technique can reduce the data sample and cause loss of information.
  - b. Replace with an “estimated value.” There are several methods to estimate a missing value, such as the mean value of the variable, the median, the most frequent value, and so on.
2. Handling noise: a noisy value is a value that is not the correct one. It is also known as corrupt data. The noisy value may be very close to the true signal.

3. Handling outliers: an outlier is a value that is much different than the other values. Most of the time, the outliers are noise, but sometimes a data point that is true signal can be an outlier.
4. Instance selection (IS): not all instances are equally important. IS consists of the selection of the most appropriate examples for the learning stage. It is also known as dataset reduction or dataset condensation. During the IS process, you can select those most representative instances, free of noise, outliers, or missing values. Some of the used FS algorithms are those based on rough set theory and fuzzy rough set theory.
5. Feature selection (FS): not all features are equally important. FS consists of the selection of the most representative variables or features for the learning stage. Selecting the right subset of features to be used for the learning phase has proven an improvement in the performance of supervised and unsupervised learning.

### 1.3 Learning algorithms background

In this section, we will describe the most significant learning algorithms from the state of the art, emphasizing those that were used in the present research. First, we will describe some classifiers and then some regressors.

#### 1.3.1 Classification task

As we defined in previous sections, classification is the learning task where each input vector corresponds to a discrete output value, known as a class. Next we will describe the most representative classifiers in the state of the art.

- Decision tree C4.5 [12]: In 2008, it was ranked as #1 in the Top 10 Algorithms in Data Mining pre-eminent paper published by Springer LNCS. C4.5 builds decision trees (DT) from a training set, using the concept of information entropy. At each node of the tree, C4.5 chooses the attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. The splitting criterion is the normalized information gain (difference in entropy). The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then recurses on the partitioned sublists.<sup>1</sup> Decision trees are predictive models that use a set of binary rules to calculate a target value.
- k nearest neighbors' classifier (kNN) [13]: It is a non-parametric algorithm. Its purpose is to use a dataset in which the instances are separated into several classes to predict the classification of a new instance. This method, for a new example to be classified, finds its k nearest neighbor using Euclidean distance, and then the example is classified by a majority vote of its neighbors. In a similar way, this method is used in a regression task. The numeric output is mean of the nearest neighbors.
- Random forest (RF) [14]: It is an ensemble method formed by decision trees. During the training phase, the method builds  $n$  decision trees from randomly

---

<sup>1</sup> [https://en.wikipedia.org/wiki/C4.5\\_algorithm](https://en.wikipedia.org/wiki/C4.5_algorithm).



sampled datasets with replacement and randomly selected subset of features, where  $n$  is an input parameter. In the testing phase, each individual tree spits out a class prediction and the class with the most votes is then predicted. RF avoids the overfitting of the traditional decision trees.

- Multilayer perceptron (MLP) [15]: It is one of the most used artificial neural networks. It consists in a set of layers (minimum 3): one input layer, one or more hidden layer, and one output layer. The input layer has as many neurons as features in the training set. The number of hidden layers and the number of neurons in these layers are parameters defined by the user. The number of neurons in the output layer corresponds to the number of classes in training set. MLP used backpropagation for the learning process. MLP can be used for classification and regression task.
- Support vector machine (SVM) [16]: It is a discriminative classifier defined by a separating hyperplane. This algorithm performs as follows: given a labeled training set, it outputs an optimal hyperplane which categorizes new examples.

### 1.3.2 Regression task

Regression is a widely used task in the world of industrial applications. It learns from the data and then when facing a new entry, is able to predict an output value. The most used regression algorithms are:

- Linear regression (LR) [17]: is a linear method that models the relationship between a group of dependent variables and one or more independent variables. In LR the relationships are modeled using linear predictor functions.
- Partial least square (PLS) [18]: is also similar to linear regression but that at the same time projects the data into a lower dimensional space, so that less variables are used in reality in the prediction model.
- Decision tree regressor: is regression method that works in the same way as the DT as a classifier; it was introduced in [19]. A decision tree arrives to an estimation by asking a series of questions to the data. Every node of the tree represents a binary question to be answered. Each question is further restricting our possible values until the model has enough confidence to make a single prediction. In this way, it is possible to build very accurate rules about the data.
- Ridge [20]: is a method of regularization also known as Tikhonov regularization that puts weighted  $l_2$  norm penalty on the regression coefficients. This method has shown very good results in regression problems, specifically in those of linear regression with the problem of multicollinearity. Multicollinearity, correlated independent variables, is very common in problems with a large number of features.
- LASSO [21]: is another regularization method that puts weighted  $l_1$  norm penalty on the regression coefficients. The least absolute shrinkage and selection operator, known as LASSO, is a method that performs variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model it generates.

- Gaussian process regression (GPR) [22]: is a non-parametric, Bayesian method for regression that infers all possible values over a probability distribution. In recent years, GPR has gained popularity among the machine learning researchers because of its robustness and performance in terms of classification and prediction accuracy. The advantage of using GPR is that it can be utilized in the exploration and exploitation scenario [22].

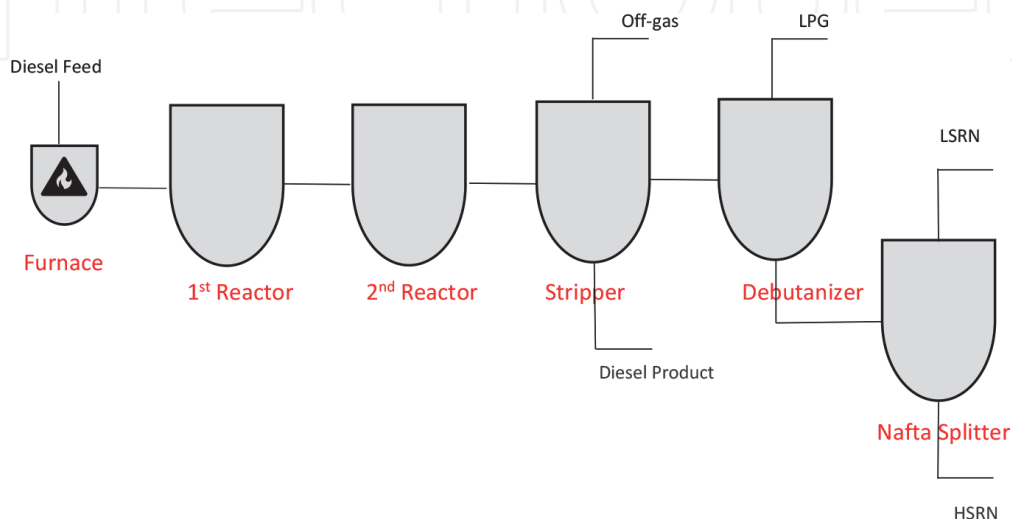
## 2. Use case 1: predicting output quality in Tüpras

Obtaining high-quality products is a fundamental objective of the Turkish Refinery Tüpras. Its four main products (diesel 95%, diesel sulfur, HSRN 95%, and LSRN 95%) must meet certain quality parameters dictated by the customer. In practice, achieving the quality required by the client is not a simple task, since during the distillation process the oil is subjected to many physical and chemical processes. However, taking into account that (a) in each of the phases of distillation of the crude oil, many variables are measured (in the laboratory or using sensors), (b) the initial chemical properties of the crude oil are known, and (c) the company have historical data on the final quality of the products, in this investigation, we will use machine learning techniques to predict the final quality values of Tüpras products.

### 2.1 Problem description

The main task of the Tüpras refinery is to convert crude oil into usable final products, satisfying the specifications established by consumers. To achieve the quality specifications, it is necessary to take many decisions, which means in our context change the manipulated parameters in the distillation process. **Figure 3** shows how the crude oil goes through a distillation process.

As can be seen in **Figure 3**, crude oil goes through several processes before becoming a final product. When we analyzed the historical data we have, we observed that in only 7 of the 254 days of which we have information, the three products were in the desired range. This gives us the measure of how hard it can be to achieve the desirable quality. Predictions based on historical values, using ML, can help achieve the desired quality. Knowing in advance the quality value, it will be possible to take decisions and make changes in the distillation process that allows to reach the desired value.



**Figure 3.**  
Tüpras refinery process scheme.

### 2.1.1 Variables and the frequency in measuring

The complete cycle, starting with crude oil until transforming into a high-quality diesel, lasts approximately 240 min (4 hours). Having a large amount of data that comes from different sources, measured with different frequency, our first task is to create a dataset that logically and consistently unifies the complete distillation cycle.

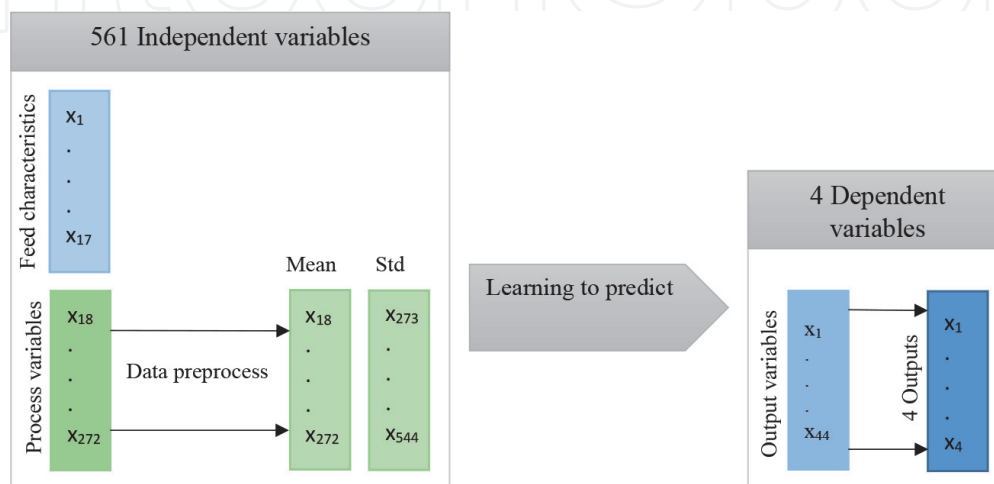
Data was collected from 260 days, but after removing missing data, we have 254 days left. The data consists of:

- 17 raw input feed characteristics measured once a day where the timepoint was not specified.
- 272 process-related parameters measured every minute each day, in total 1440 measurements each day.
- 44 output feed characteristic variables where we only predict four of them (diesel 95%, diesel sulfur, HSRN 95%, and LSRN 95). The output variables were measured at 8 am every day and are valid for process measurements from 4 am to 8 am.

For the creation of the dataset, we consider:

- The dataset was created in the form  $x_1, x_2, \dots, x_n \rightarrow y$ , where  $n$  is the number of independent variables and each  $x_i$  represent a variable measured during the distillation process. These variables can be sensor measurements, manipulated variables, control variables, and the 17 raw input feed characteristics. The output variable or the dependent variable ( $y$ ) is the final quality. In this way we have a decision system ready for learning task.
- We take into account the time delay of the process.

Thus, in total we have 279 (17 feed +272 process) independent variables that are used to predict four dependent variables. However, since the output variables are only valid for 4 hours, that is, 240 minutes in total, there are  $240 \times 272$  measurements plus 17 input variables for each output variable sample. Thus, there are many more independent variables than dependent variables, and therefore, we use the mean and standard deviation of each process parameter over each 4-hour period as



**Figure 4.**  
 Dependent and independent variables in the learning process.



features. Consequently, we have  $17 + 272 \times 2 = 561$  independent variables for each 4-hour period. A graphic description can be found in **Figure 4**.

Notice also that only the above 4 hours of each day have labels—that is, have valid output variables (data is labeled)—while the remaining 20 hours lack valid labels (data is unlabeled).

Now we have the data ready for the learning process; in the next subsection, we will describe learning process.

## 2.2 A first approach: predicting final quality

In this subsection, we will describe the different experiments we used to evaluate the prediction performance for the output variables. First, we will report results from learning only from the labeled data. Next, we will present an analysis that uses learning curves to understand the learning problem, whether more data or more features would help improve the performance. Finally, we will describe results from applying semi-supervised learning where also the unlabeled data was used.

### 2.2.1 Experiment 1: prediction with only labeled data

In the first experiment, we use four regression methods described in previous section. We use LOO<sup>2</sup> cross-validation to investigate which method works best for predicting the four output variables when only trained on the labeled data. In LOO, we use  $N - 1$  (where  $N = 254$  days) data points for training the machine learning method that is then tested on the remaining data points. This is repeated  $N$  times resulting in  $N$  different predictions. For evaluating the prediction performance, we use root mean squared error (RMSE) that takes the square root of the mean of the square of the difference between the predictions and the true values.

In **Table 1**, the result is shown, and as can be seen, ridge regression has the overall best result with the smallest error (RMSE) for three of the output variables, while random forest has the smallest error for two of the output variables. We also notice that the errors of the two first output variables are not improved much by any of the methods compared to PMEAN, while the two last are improved quite a lot. Thus, in the following section, we will try to improve the performance for ridge regression.

### 2.2.2 Experiment 2: learning curves

In order to investigate whether we can learn some more from collecting more data or whether more features are needed, we can plot learning curves. Learning

| Methods            | Output variables |               |          |          |
|--------------------|------------------|---------------|----------|----------|
|                    | Diesel 95%       | Diesel sulfur | HSRN 95% | LSRN 95% |
| PMEAN <sup>a</sup> | 2.50             | 1.00          | 8.22     | 5.31     |
| Ridge              | 2.36             | 0.79          | 3.69     | 3.68     |
| PLS                | 2.44             | 0.79          | 4.53     | 4.34     |
| RF                 | 2.36             | 0.73          | 4.05     | 3.96     |

<sup>a</sup>As baseline, PMEAN is a simple algorithm used as a base of comparison. PMEAN uses the mean of the training data as prediction.

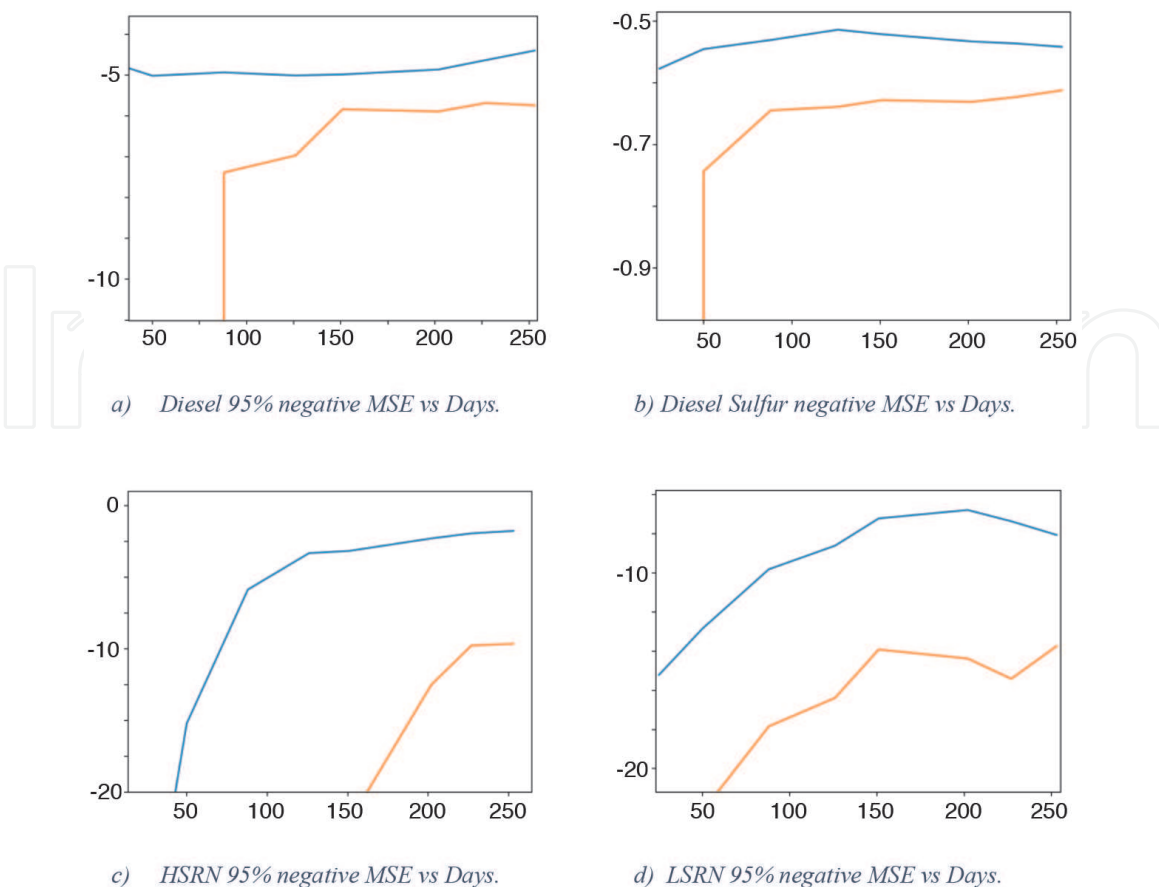
**Table 1.**  
RMSE of the LOO cross-validation result for labeled data.

<sup>2</sup> Leave-One-Out

curves show the number training examples on the x-axis and the accuracy on the y-axis for both training data and test data that was not used for training. As accuracy we use the negative mean squared error (negative MSE), that is, the negative square of the RMSE. The learning curves for the output variables using ridge regression are shown in **Figure 5**.

The upper blue curve shows accuracy for training data, and the lower orange curve shows the accuracy for test data. Higher value means better performance, and as can be seen, the accuracy is better for the training curve than for the test curve, which is natural since the test curve should indicate the generalization performance of the algorithm. By extrapolating the curves, we can draw some conclusions from them.

The number of training examples is quite limited so what can be learned from the curves is also quite limited. However, we notice that the learning curves for the two upper output variables—**Figure 5a** and **b**—are quite similar, while the same can be said for the two lower learning curves, **Figure 5c** and **d**. We also observe that for the two upper learning curves, the test curves reach a plateau around  $-6$  and  $-0.65$ , respectively, after which no more improvement is seen. Neither do we see much of an improvement for the training curves. This indicates that more training examples will not likely improve the performance, but instead we need more features or a more complex algorithm. For the lower left learning curve (c), we do not see the plateau that clearly, while the lower right curve (d) shows increasing performance with more data. So, for the lower curves, more training examples might improve the performance. In the next experiment, we will investigate this by using a semi-supervised approach that also uses the unlabeled data for training.



**Figure 5.**  
The learning curves for the four different output variables.

### 2.2.3 Experiment 3: prediction with a semi-supervised approach

A semi-supervised approach uses both labeled and unlabeled data in the training phase [11]. In essence, we achieve this by creating more training examples and by using the ML algorithm to label them. We create unlabeled data by moving a sliding window of length 4 hours over each day with time step of 1 hour from 4:00 am to 0:00 pm. This created 20 periods of 4-hour data with 1 labeled and 19 unlabeled time periods. This increased the number of training examples from 254 to almost 5000 ( $\approx 20 \times 254$ ). The algorithm we use to train on both labeled and unlabeled data has the following steps:

1. Train the learning method using only labeled data.
2. Predict the labels (output variables) of the unlabeled data.
3. Train the learning method using both labeled data and the data with predicted labels.
4. Repeat step 2 and step 3 until the difference between the old and new predicted labels becomes small.

The algorithm uses the maximum likelihood principle in that it converges toward the values with maximum likelihood, similar to how the expectation-maximization (EM) algorithm works [23].

For evaluation, we use also LOO cross-validation. That is, we used only labeled data for evaluation but used all unlabeled data in the training phase as described above and tested the trained method on the left-out labeled data. The result is shown in **Table 2**. The overall best approach is clearly ridge regression with semi-supervised learning. This confirms the observation from the learning curve analysis that the first and second output variable would not improve with more training examples, while the two last output variables we can indeed see improved performance with more data.

### 2.3 A second approach

After concluding a first stage in which we carried out the study shown in the previous section, we obtained new data from Tüpras. With the new data with a total 269 samples, we designed three new experiments. The objective of the following three experiments is to find with which dependent variables the best predictions of the variables are achieved.

| Methods      | Output variables |               |          |          |
|--------------|------------------|---------------|----------|----------|
|              | Diesel 95%       | Diesel sulfur | HSRN 95% | LSRN 95% |
| PMEAN        | 2.50             | 1.00          | 8.22     | 5.31     |
| Ridge        | 2.36             | 0.79          | 3.69     | 3.68     |
| Ridge (SEMI) | 2.34             | 0.82          | 2.54     | 3.31     |

**Table 2.**  
RMSE of the LOO cross-validation with semi-supervised learning.

### 2.3.1 Experiment 1: not using the controlled variables

In our first experiment, we will predict the quality of the output variables without using the controlled variables. As in previous section, we will use LOO validation. **Table 3** shows the results. As we can observe, best results are obtained in all cases for LASSO, while ridge performs much worse for diesel 95% than in the first approach.

### 2.3.2 Experiment 2: using controlled variables

In our second experiment, we will predict the quality of the output variables including the controlled variables as independent variables. **Table 4** shows the results. Again, LASSO gets the best results in all cases, while ridge is the second best.

### 2.3.3 Experiment 3: using only controlled variables and diesel feed

In our third experiment, we will predict the quality of the output variables using only controlled variables and diesel feed characteristics. **Table 5** shows the results. LASSO gets the best results in all cases.

| Methods | Output variables |               |          |          |
|---------|------------------|---------------|----------|----------|
|         | Diesel 95%       | Diesel sulfur | HSRN 95% | LSRN 95% |
| PMEAN   | 2.47             | 0.99          | 8.21     | 5.31     |
| Ridge   | 2.58             | 0.85          | 2.64     | 3.49     |
| PLS     | 2.50             | 0.82          | 3.32     | 3.91     |
| LASSO   | 2.26             | 0.78          | 2.21     | 2.68     |

**Table 3.**  
 RMSE of the LOO cross-validation for experiment 1.

| Methods | Output variables |               |          |          |
|---------|------------------|---------------|----------|----------|
|         | Diesel 95%       | Diesel sulfur | HSRN 95% | LSRN 95% |
| PMEAN   | 2.45             | 1.00          | 8.24     | 5.34     |
| Ridge   | 2.41             | 0.88          | 2.79     | 3.56     |
| PLS     | 2.72             | 0.88          | 3.30     | 3.92     |
| LASSO   | 2.29             | 0.79          | 2.26     | 2.81     |

**Table 4.**  
 RMSE of the LOO cross-validation for experiment 2.

| Methods | Output variables |               |          |          |
|---------|------------------|---------------|----------|----------|
|         | Diesel 95%       | Diesel sulfur | HSRN 95% | LSRN 95% |
| PMEAN   | 2.46             | 1.00          | 8.23     | 5.32     |
| Ridge   | 2.18             | 0.85          | 5.68     | 4.42     |
| LASSO   | 2.07             | 0.82          | 5.64     | 4.43     |

**Table 5.**  
 RMSE of the LOO cross-validation for experiment 3.

## 2.4 Partial conclusions

In this section we have described the use of regression methods to predict the four output variables of the Tüpras refinery.

In our first approach, we have described the evaluation of using ridge regression, partial least squares, and random forest to the problem of predicting the four output variables, where the ridge regression had the best performance. We have also shown that using a semi-supervised approach, we could improve the performance for two of the variables, which also indicates that more data collected from the process would most likely further improve the performance. However, for two of the variables, the learning methods did not improve the performance much compared to the baseline using the mean value of the training data, and neither did semi-supervised learning. For those two variables, we need to consider other relevant features than the mean or standard deviation.

When using more data (second approach), we constantly get the best results using LASSO regressor for the prediction of the four output variables. From our results we conclude that:

- For the prediction of diesel 95 it is better to use only the controlled variables and the diesel feed characteristics.

In contrast, ridge regression shows varying performance in the experiments, while being many times the second best. Thus, ridge seems to be less stable than LASSO for this problem.

## 3. Use case 2: predictive maintenance model from micro gas turbine

The need to predict maintenance intervals is a problem that currently has great relevance in the field of ML applied to the industry. Predicting in advance if a device needs maintenance can result in significant savings in time and money. With predictive maintenance, important failures and breakdowns in production time can be avoided [5, 23]. It is a fact that the maintenance intervals recommended by the manufacturers almost never correspond in practice with reality. This is largely due to the fact that local conditions vary a lot from one environment to another and manufacturers operate with generic measurements that do not take into account specific conditions.

In this section we will describe a proposal to estimate the performance degradation of a fleet of micro gas turbines. An important issue to consider is that there is no explicit degradation measure, which therefore must be estimated.

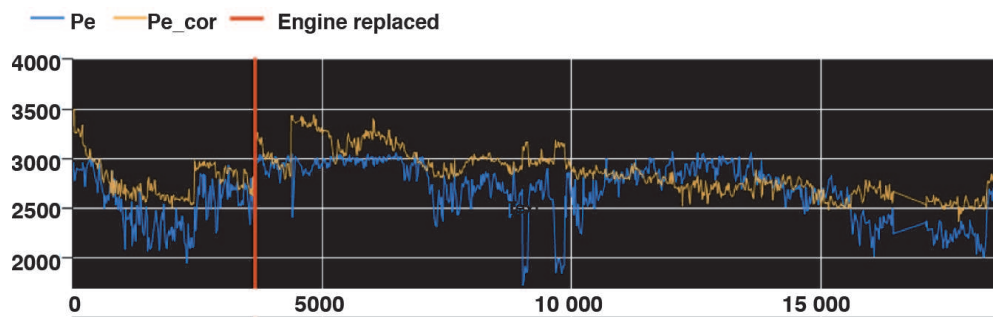
### 3.1 Problem description

The existing method for estimating degradation uses a linear model fitted to data from a reference system which then is used to correct the generated power from an installed system. Thus, the values are corrected and normalized so that they can be compared. In **Figure 6**, we can observe an example of the current approach. The yellow curve is the corrected power which shows the current approach to measuring degradation.

In addition to that, there are some conditions that make the problem unusual. These conditions are as follows:

- The systems are small-scale and low-cost installations, so there is only a small number of sensors available.





**Figure 6.** *Pe is measured and Pe\_cor is corrected power; engine replacement indicates start of currently installed engine life.*

- At the time of development of our method, there were not many systems installed and not many failures recorded, and thus, a traditional supervised ML approach could not be used.
- Each system is always running at full capacity, where the ideal power is the maximum power generated when there is no loss due to degradation and no effect of ambient conditions.
- Finally, there are recordings of maintenance actions, but the effect of an action on remaining degradation is not known.

Given the above list of circumstances, the design goal of the proposed method is to measure degradation:

1. Using only data from real systems and removing the need for a reference system
2. More smoothly than the existing method
3. Relative to the ideal power generation

### 3.1.1 Data collection and preprocessing

Data was collected from five different micro gas turbines with system ID 24, 27, 28, 29, and 30. The data was sampled every minute, but we used only samples from every second hour, since it was deemed to be sufficient for long-term degradation modeling. We use data from the parameters shown in **Table 6**.

## 3.2 Approach: power degradation model

The proposed method uses a regression model where physical properties are taken into account. As we said before, we are not facing a classic problem of supervised learning, since degradation cannot be measured. Thus, instead we let both the degradation and ideal power be properties of the model, and the model is trained to predict the measured electric power.

Now we introduce our model: let  $y$  be the measured power,  $\vec{x}$  be a column vector with the ambient parameters like weather, pressure, etc. Then, let  $\vec{t}$  be a column vector with time-dependent variables and  $n$  and  $m$  be the number of systems and number of maintenance periods, respectively. We use  $1 \leq i \leq n$  to denote a specific

| Variables                           | Parameters  | Unit      |
|-------------------------------------|---|-----------|
| Predicted variable (y)              | • Net electric output power   | Watts     |
| Ambient (contextual) variables (x)  | • Measured return water temperature                                     | Kelvin    |
|                                     | • Inlet air temperature   | Kelvin    |
|                                     | • Ambient pressure  | Bar       |
|                                     | • Ambient pressure at stand still                                       | Bar       |
|                                     | • Measured turbine speed  | Rpm       |
|                                     | • Set point requests based on heat demand                               | —         |
|                                     | • The internal set point for desired speed and turbine exit temperature | —         |
| Ambient pressure variable*          | • Ambient pressure is missing   | Dummy     |
| Time-dependent variables            | • Total number of running hours   | Hours     |
| Affecting the degradation trend (t) | • Total number of starts and stops                                      | Frequency |
| Maintenance actions (M)             | • Total number of running hours when action was taken                   | —         |
| The ideal power per system (k)**    | • Net electric output power during installation                         | Watts     |

\*To handle missing values of the ambient pressure variable, we add a dummy variable that is 1 when the variable is missing and 0 when it is present.

\*\*The ideal power was measured when a system was installed and corresponds to the power that would be generated without disturbances from ambient variables and degradation due to wear.

**Table 6.**  
Parameters used in the prediction model.

system and  $1 \leq j \leq m$  for a specific maintenance period. Now, we define the generic model of degradation as follows:

$$y = k_i + g(\vec{x}) + e(\vec{t}; i, j) \quad (1)$$

where  $k_i$  is the known ideal power generation for system  $i$ , function  $g$  is the effect of ambient variables, and function  $e$  is the degradation over time due to wear.

In the above equation, the signal is divided into two parts: a variation caused by ambient conditions and a degradation trend given by the time-dependent variables. It is also assumed that the variation due to ambient variables is the same for different systems, while the degradation is dependent on both the individual system and the maintenance period.

Let us assume a linear model for both functions  $g$  and  $e$  as follows:

$$\begin{aligned} g(\vec{x}) &= \vec{c}^T \vec{x} \\ e(\vec{t}; i, j) &= a_i + b_j + \vec{e}_0^T \vec{t} + \vec{e}_i^T \vec{t} \end{aligned} \quad (2)$$

where  $\vec{c}$  is a column vector to model ambient conditions and  $a_i$  and  $b_j$  model remaining degradation at start or after a maintenance action for system  $i$  and maintenance period  $j$ , respectively, and  $\vec{e}_0$  and  $\vec{e}_i$  are column vectors modeling the degradation common for all systems and for each individual system, respectively.

| System | r    | RMSE  | MAPE |
|--------|------|-------|------|
| 24     | 0.91 | 81.80 | 2.35 |
| 27     | 0.95 | 72.13 | 2.20 |
| 28     | 0.82 | 71.31 | 1.95 |
| 29     | 0.92 | 62.85 | 1.83 |
| 30     | 0.95 | 52.49 | 1.50 |
| All    | 0.92 | 70.59 | 2.01 |

**Table 7.**  
*Estimation model results over five systems.*

In order to get a feasible solution, we put a  $l_1$  regularization on the remaining degradation coefficients in  $\vec{a}$  and  $\vec{b}$  so that the coefficients are kept close to zero. We also assume that the degradation is monotonic so that  $\vec{e} \leq 0$ . The solution was implemented using a machine learning framework called Keras<sup>3</sup> together with TensorFlow<sup>4</sup> as backend.

### 3.3 Experimental results

In this experiment, we use the existing method that uses corrected power derived from a reference system to validate the new method. **Table 7** shows, for each of the five systems we have tested, the Pearson correlation coefficient (r) between the estimated negative degradation and the corrected power, the root mean squared error, and mean absolute percentage error (MAPE in %) for predicting the measured power.

As can be seen, the correlation coefficients are above 0.9 for all but one system (28), which indicates that the proposed method is indeed a good replacement for the corrected power. Also, the RMSE and MAPE are of reasonable sizes.

### 3.4 Partial conclusions

In this use case, we presented a machine learning approach that incorporates physical properties into the model in order to estimate the degradation of a fleet of gas systems. In addition, we show that it was a good replacement of the existing approach to measuring degradation that was based on data from a reference system.

## 4. Conclusion

In this chapter, we presented an overview of machine learning and presented example use cases where we applied machine learning. In the first use case, we predicted the diesel product quality using common regularized linear regression, while in the second use case, we used a more customized regularized regression to implement predictive maintenance.

<sup>3</sup> <https://keras.io>.

<sup>4</sup> <https://tensorflow.org>.

As general conclusions, we can summarize:

- LASSO and ridge regressor were very efficient methods in predicting diesel quality at UC 1.
- For the prediction of diesel 95, it is better to use only the controlled variables and the diesel feed characteristics.
- The incorporation of physical properties into the degradation model in use case 2 is very useful for the final maintenance prediction.

A summary of general approach to solving a problem with machine learning is to:

1. Start by defining the learning problem: what variable should be predicted? If there is no explicit variable, it might be an unsupervised problem, but as in use case 2, it can also be a variable that is not measured. Thus, the sought variable needs to be extracted from or part of the estimated model.
2. Next, chose a performance metric that measures the desired outcome. In use case 1, it was quite simple since the diesel quality was measured directly, while in use case 2, the desired outcome—the time when the degradation of a system is too bad—was not measured directly.
3. Then, start out with a simple model, like a linear regression model, which also can be used as a baseline for comparison of more complicated models used in the next step.
4. Plot and analyze the learning curves (see Section 2.2.2). If the curves indicate potential of using a more complex model, then try with a more complex model like a random forest or a neural network. However, the selection of model is also dependent on the size of the dataset. If there is only small dataset as in use case 1, it is not possible to use a too complex model since more model parameters need more data for training.
5. Finally, test the models on a dataset not used for training above. This is to ensure that the performance measures the generalization power of the model and to avoid overfitting to the training data.

As an overall conclusion, we can see that we ended up with quite simple variants of linear models in both use cases, which is not uncommon given the authors experience from industrial problems. Another general comment is that in most cases each industrial problem is quite unique and there is no single solution that fits every problem. So, it is important to understand the problem domain and chose methods that fit that particular problem. Hence, machine learning is not a silver bullet that will solve all problems. If there is a good physical model, a machine learning model will probably not be a better choice. However, it might be a benefit to create a hybrid model combining the physical model with a data-driven machine learning model.

## **Acknowledgements**

The research of Dr. Enislay Ramentol has been funded by the European Research Consortium for Informatics and Mathematics (ERCIM) Alain Bensoussan

Fellowship Programme. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 723523.



IntechOpen

## Author details

Enislay Ramentol<sup>1†</sup>, Tomas Olsson<sup>2\*†</sup> and Shaibal Barua<sup>2,3†</sup>

1 Fraunhofer Institute for Industrial Mathematics, Kaiserslautern, Germany


2 Research Institute of Sweden, Västerås, Sweden

3 School of Innovation, Design and Engineering, Mälardalen University, Västerås, Sweden

\*Address all correspondence to: [tomas.olsson@ri.se](mailto:tomas.olsson@ri.se)

† The first two authors contributed equally; the third author contributed description of the state-of-the-art algorithms.

## IntechOpen

© 2020 The Author(s). Licensee IntechOpen. Distributed under the terms of the Creative Commons Attribution - NonCommercial 4.0 License (<https://creativecommons.org/licenses/by-nc/4.0/>), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited. 



## References

- [1] Mitchel T. Machine Learning. New York: McGraw Hill; 1997
- [2] Mazurowskia MA, Habasa PA, Zuradaa JM, Lob JY, Bakerb JA, Tourassib GD. Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural Networks*. 2008;**21**:427-436
- [3] Merschmann L, Plastino A. A lazy data mining approach for protein classification. *IEEE Transactions on Nanobioscience*. 2007;**6**:36-42
- [4] Milletari F, Navab N, Ahmadi S-A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. arXiv:1606.04797v1 [cs.CV]. 2016
- [5] Ramentol E, Gondres I, Lajes S, Bello R, Caballero Y, Cornelis C, et al. Fuzzy-rough imbalanced learning for the diagnosis of high voltage circuit breaker maintenance: The SMOTE-FRST-2T algorithm. *Engineering Applications of Artificial Intelligence*. 2016;**48**:134-139
- [6] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. arXiv:1505.04597v1 [cs.CV]. 2015
- [7] Kasabov NK. Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering. Cambridge, Massachusetts, London, England: Massachusetts Institute of Technology; 1998
- [8] Cherkassky V, Mulier F. Learning from data. In: Concepts, Theory, and Methods. 2nd ed. Hoboken, New Jersey: John Wiley & Sons, Inc.; 2007
- [9] Orriols-Puig A, Bernadó-Mansilla E, Sastry K, Goldberg DE. Substructural surrogates for learning decomposable classification problems: Implementation and first results. In: Conference Companion on Genetic and Evolutionary Computation: GECCO 07. 2007
- [10] Triguero FHI, García S. Self-labeled techniques for semi-supervised learning: Taxonomy, software and empirical study. *Knowledge and Information Systems*. 2015;**42**:245-284
- [11] Chapelle O, Schlkopf B, Zien A. Semi-Supervised Learning. 1st ed. Cambridge, Massachusetts, London, England: The MIT Press; 2010
- [12] Quinlan JR. C4.5 Programs for Machine Learning. CA: Morgan Kaufmann; 1988
- [13] Peterson LE. K-nearest neighbor. *Scholarpedia*. 2009;**4**(2):1883. Revision 137311
- [14] Ho TK. Random decision forests. In: Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1). Washington, DC, USA: IEEE Computer Society. 1995
- [15] Gardner MW, Dorling SR. Artificial neural networks (the multilayer perceptron)—A review of applications in the atmospheric sciences. *Atmospheric Environment*. 1998;**32**(14):2627-2636
- [16] Hearst MA. Support vector machines. *IEEE Intelligent Systems*. 1998;**13**(4):18-28
- [17] Seber GAF, Lee AJ. Linear Regression Analysis. 2nd ed. Hoboken, New Jersey: John Wiley & Sons, Inc.; 2003
- [18] Vinzi VE, Chin WW, Henseler J, Wang H. Handbook of Partial Least Squares: Concepts, Methods and

Applications. 1st ed. Berlin, Heidelberg:  
Springer Publishing Company,  
Incorporated; 2010

[19] Breiman L, Friedman J, Olshen R,  
Stone C. Classification and Regression  
Trees. Monterey, CA: Wadsworth and  
Brooks; 1984. New edition

[20] Hoerl AE, Kennard RW. Ridge  
regression: Biased estimation for  
nonorthogonal problems.  
Technometrics. 1970;12(1):55-67

[21] Tibshirani R. Regression shrinkage  
and selection via the Lasso. Journal of  
the Royal Statistical Society, Series B.  
1994;58:267-288

[22] Schulz E, Speekenbrink M,  
Krause A. A tutorial on Gaussian process  
regression: Modelling, exploring, and  
exploiting functions. Journal of  
Mathematical Psychology. 2018;85:1-16,  
ISSN: 0022-2496. DOI: 10.1016/j.  
jmp.2018.03.001

[23] Moon TK. The expectation-  
maximization algorithm. IEEE Signal  
Processing Magazine. 1996;13(6):47-60