Politechnika Warszawska

WYDZIAŁ MECHANICZNY
ENERGETYKI I LOTNICTWA

Metody Komputerowe w Spalaniu

Correlation rate prediction using machine learning

Author:
Kamil Macura

Supervisor:
Dr inż. Mateusz Żbikowski

Warszawa, 2018

Contents

1 Introduction

The durability issues of materials in natural and artificial environments are extremely important in the design and use of structures and devices. Corrosion is one of the main sources of material losses. It also contributes to environmental pollution and poses a threat to human health. In the US alone, corrosion costs the oil and gas industry billions of dollars a year. Corrosion monitoring helps to improve the safety and the sustainability of assets. However, corrosion is a highly nonlinear problem influenced by complex characteristics and models for predicting the corrosion rate of steel currently lack a theoretical basis. Researchers have yet to reach a consensus on the best model to predict corrosion rate or pitting risk due to the lack of good understanding of factors that affect the corrosion process. One of the solutions of corrosion prediction may be the use of machine learning and artificial intelligence. Machine learning (ML) and artificial intelligence (AI) based approaches have attracted a great deal of scientific attention and have successfully used in nonlinear and optimization problems. In order to determine the corrosion rate, two machine learning libraries were used, Scikit-Learn and Tensorflow.

2 Datasets

The analysis of the corrosion rate was based on the article [1] in which the losses of material were monitored using ultrasounds, which was triggered by various types of acids. These acids are acetic and citric acid. Based on the measurements, changes in thickness with time have been plotted on the graphs from which the points used for analysis using machine learning were read.

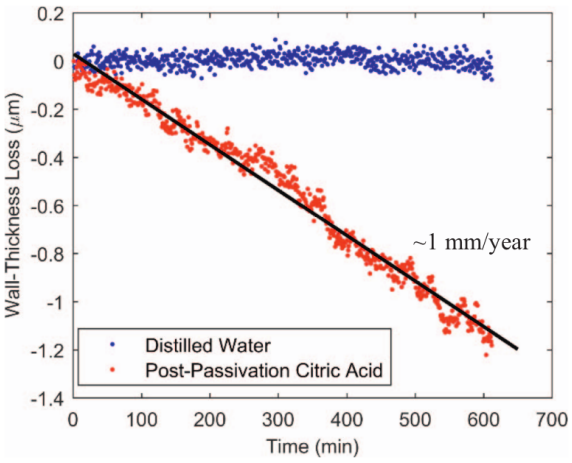

Figure 1: Loss in thickness under the influence of citric acid and distilled water as reference data.

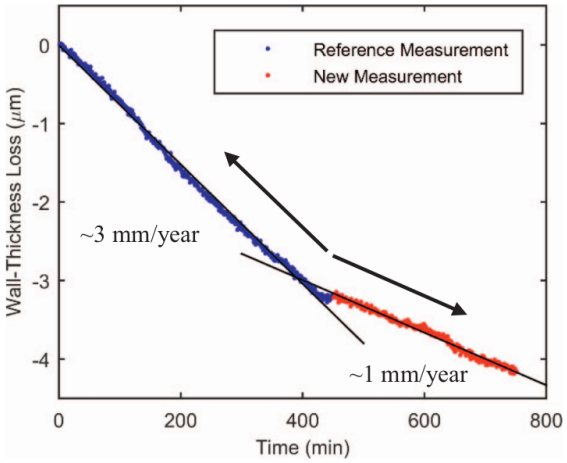

Figure 2: Loss in thickness under the influence of acetic acid.

From Fig. 1 using WebPlotDigitizer[2], the change in thickness for citric acid was read and from Fig. 2 for reference measurement, thus obtaining a base which in the first case was 982 points and in the second a little less 918 measuring points.

3 Machine learning model

As mentioned earlier, the machine learning algorithms were used to predict the corrosion rate, and to put it more accurately the Simple Linear and Polynomial Regression algorithm.

3.1 Linear Regression

3.1.1 Simple Linear Regression

In statistics, linear regression is a linear approach to modelling the relationship between a scalar response and one or more explanatory variables. The case of one explanatory variable is called simple linear regression. Simple linear regression is given by the following equation:

$$\hat{y}(w, x) = w_0 + w_1 x$$

Across the module, we designate the vector $w = w_1$ as coefficient and w_0 as intercept.
Linear regression fits a linear model with coefficients to minimize the residual sum of squares between the observed responses in the dataset and the responses predicted by the linear approximation.

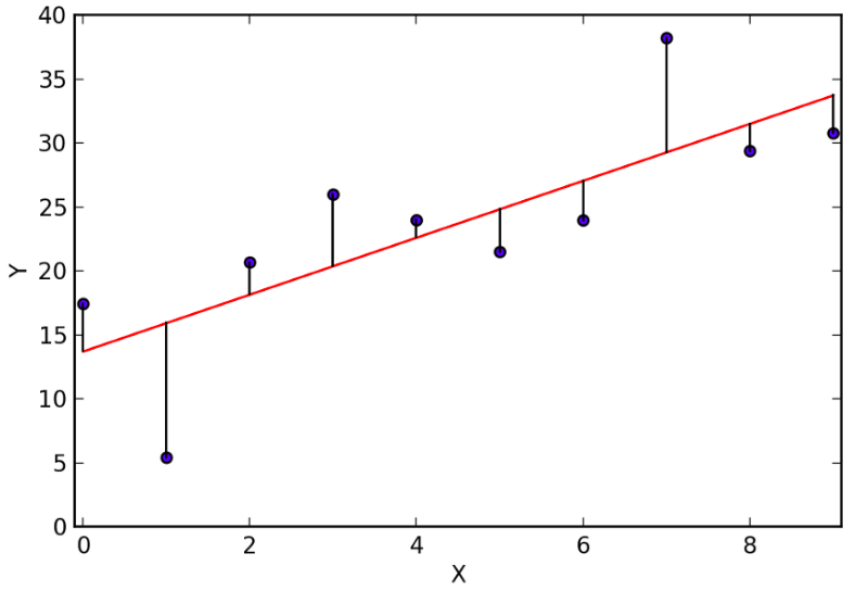

Figure 3: Simple linear regresion with marked residual

The goal was to minimize the black lines (Fig.3). It is related to minimizing the mean squared error (MSE).

3.1.2 Polynomial Regression

Polynomial regression is a form of linear regression where higher order powers (2nd, 3rd or higher) of an independent variable are included. Polynomial regression is given by the following equation:

$$\hat{y}(w, x) = w_0 + w_1 x + \cdots + w_p x_p$$

As previously we designate the vector $x = (w_1, \cdots, w_p)$ as coefficient and w_0 as intercept.

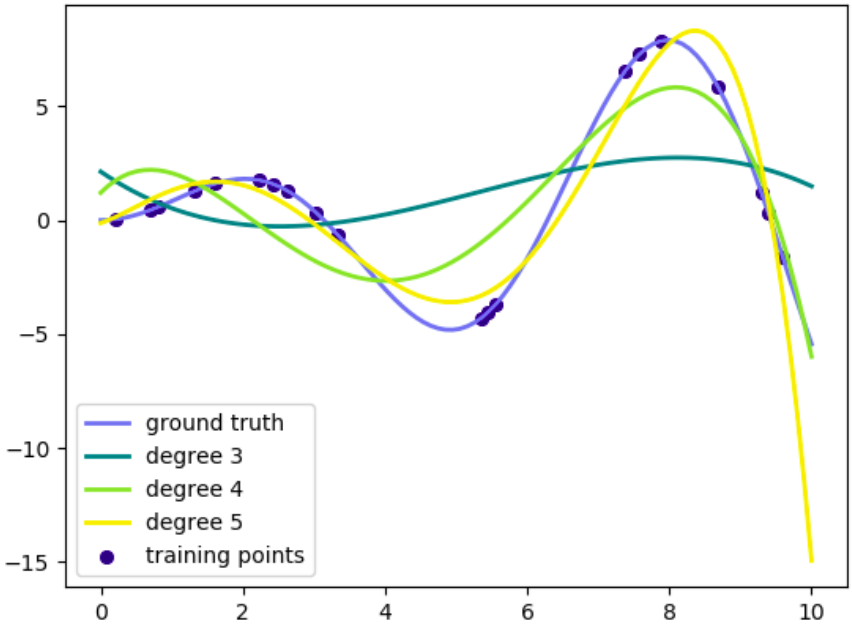

Figure 4: Polynomial regression with different order power

Choosing the right curve describing the input data in the best possible way is often subjective. One example is underfitting where the describing curve does not describe in a sufficiently accurate way, however, we can get overfitting which is due to the fact that the curve passes through many points causing that the nature of the data is not reflected. The coefficient R^2 may influence the selection of the appropriate curve

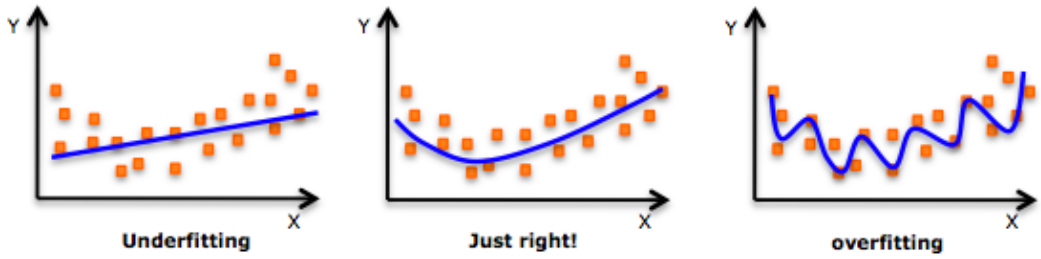

Figure 5: Linear regression with different order power

To compare the prediction models obtained MSE and R^2 were used:
MSE - The mean squared error (MSE) of an estimator measures the average of the squares of the errors—that is, the average squared difference between the estimated values and what is estimated:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

R^2 - R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression. The definition of R-squared is fairly straight-forward; it is the percentage of the response variable variation that is explained by a linear model:

$$R^2 = \frac{\sum_{i=1}^{n}(\hat{Y}_i - \bar{Y}_i)^2}{\sum_{i=1}^{n}(Y_i - \bar{Y}_i)^2} = 1 - \frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{\sum_{i=1}^{n}(Y_i - \bar{Y}_i)^2}$$

3.2 Scikit-Learn

Using the Scikit-Learn library the definition of linear regression is relatively simple, linear regression is built into the library and it is enough to call it using *from sklearn.linear_model import LinearRegression* and fit data to it. In the case of polynomial regression, it should be called using *from sklearn.preprocessing import PolynomialFeatures* and then fit and transform data using *fit_transform*.

3.2.1 Citric Acid

For citric acid using the Scikit-Learn model the following model was obtained:

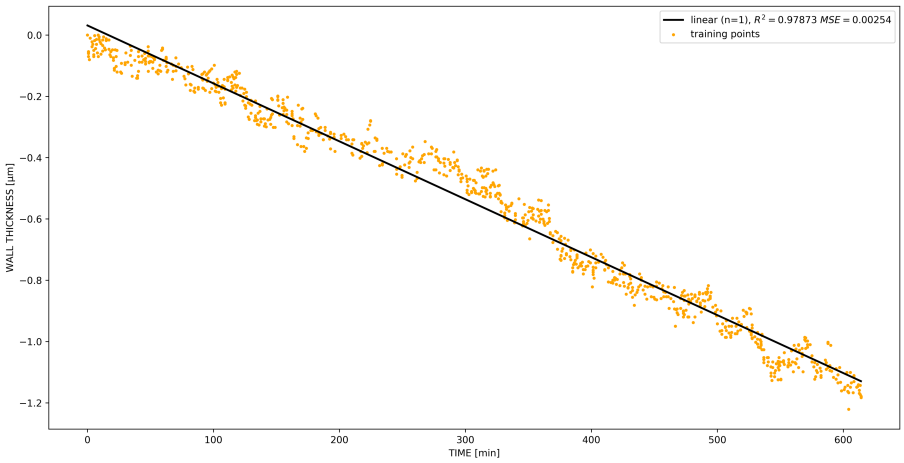

Figure 6: Linear regression

For Simple Linear regression:

$$Coefficient : -0.00189$$
$$Intercept : 0.03121$$

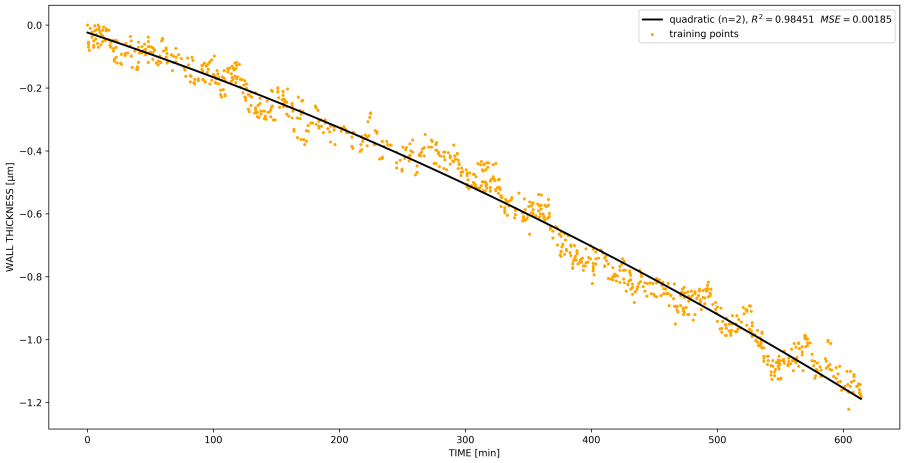

Figure 7: Quadratic regression

For Quadratic regression:

$$Coefficient : \left[-1.33282e^{-3}, 9.258679e^{-7}\right]$$
$$Intercept : -0.02378$$

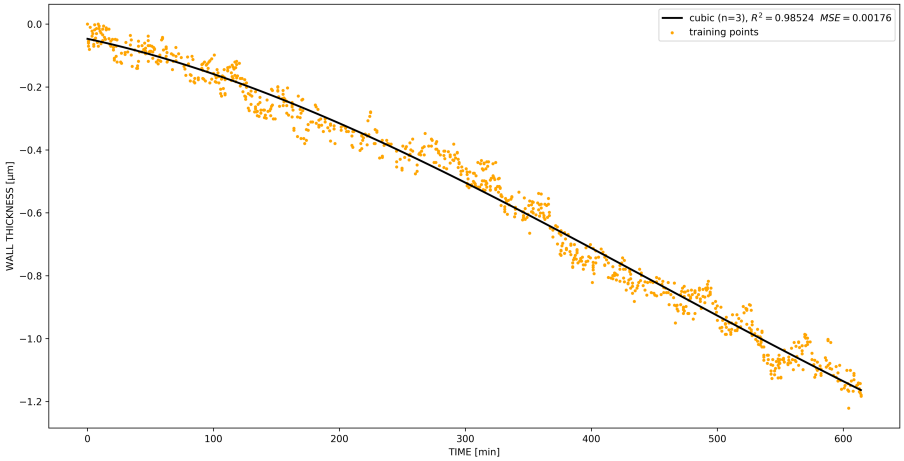

Figure 8: Cubic regression

For Cubic regression:

$$Coefficient : \left[-8.54838e^{-4}, -2.86525e^{-6}, 2.1092e^{-9}\right]$$
$$Intercept : -0.0468$$

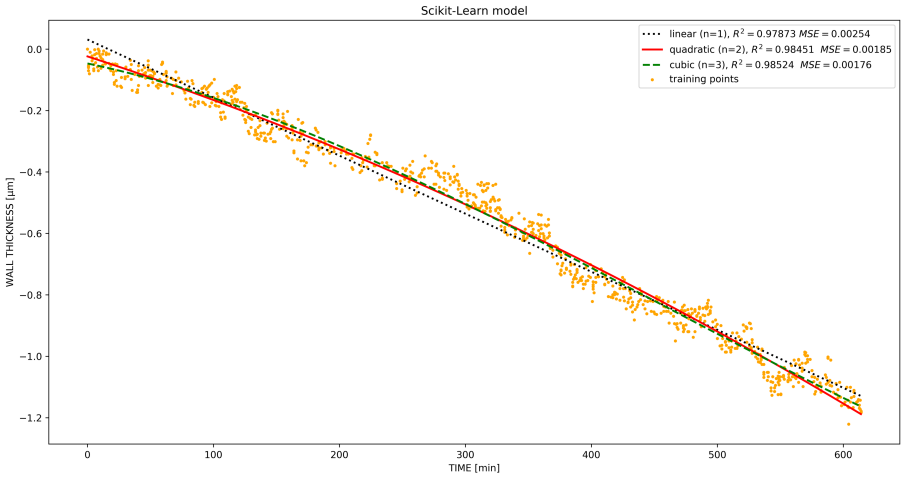

Figure 9: All linear regression curves

	Linear	Quadratic	Cubic
MSE	0.002536	0.001846	0.001759
R^2	0.978727	0.984512	0.985244

Table 1: Error of corrosion rate for citric acid using Scikit-Learn

3.2.2 Acetic acid

For acetic acid using the Scikit-Learn model the following model was obtained:

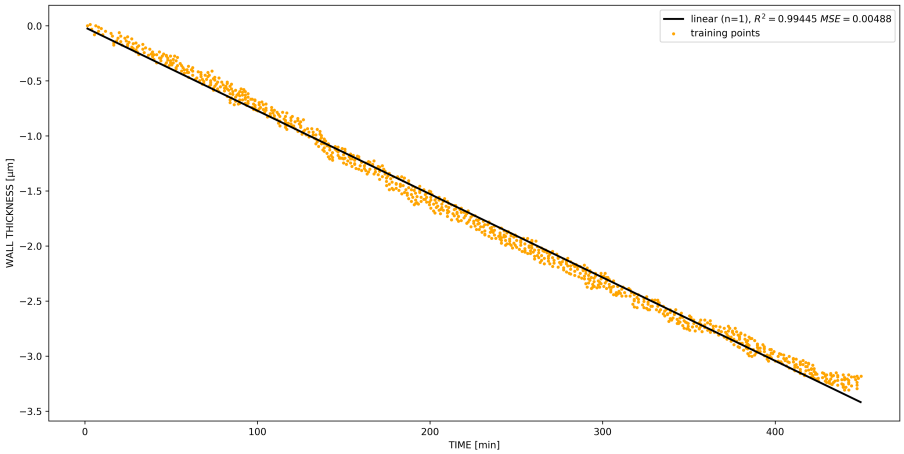

Figure 10: Linear regression

For Linear regression:

$$Coefficient : -0.00757$$
$$Intercept : -0.01379$$

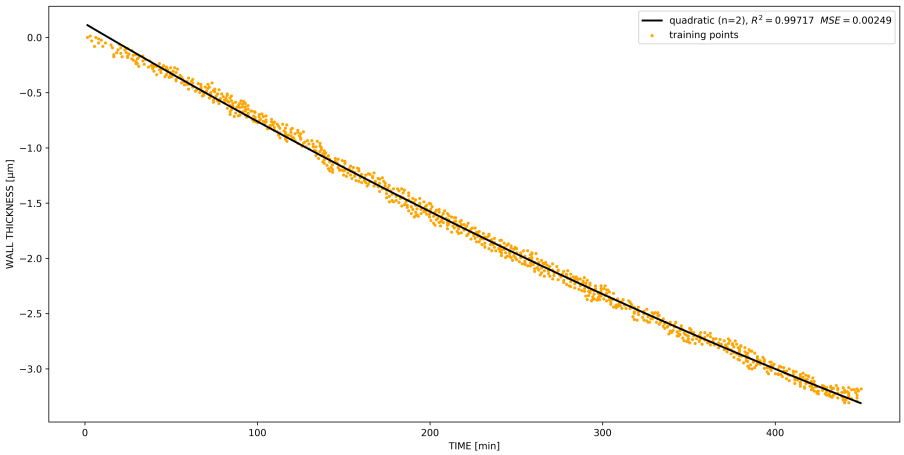

Figure 11: Quadratic regression

For Quadratic regression:

$$Coefficient : \left[-9.2011e^{-3}, -3.46923e^{-6}\right]$$
$$Intercept : 0.124684$$

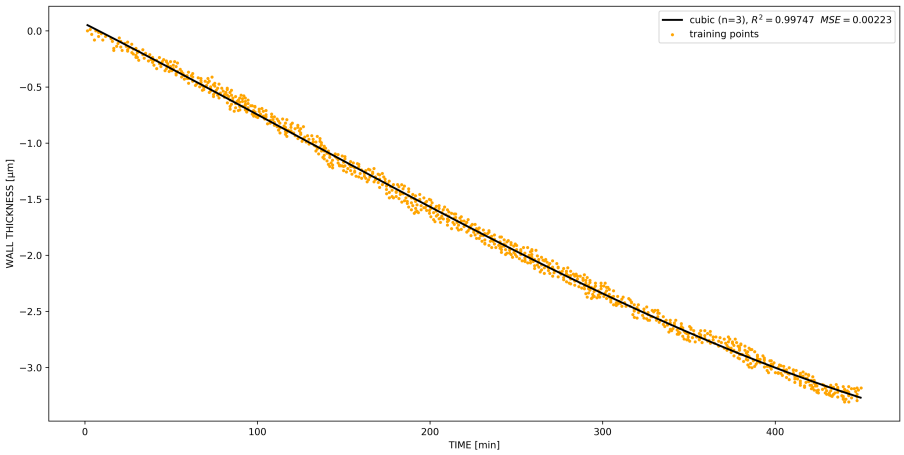

Figure 12: Cubic regression

For Cubic regression:

$$Coefficient : \left[-7.8287e^{-3}, -3.6929e^{-6}, 1.0279e^{-8}\right]$$
$$Intercept : 0.06316$$

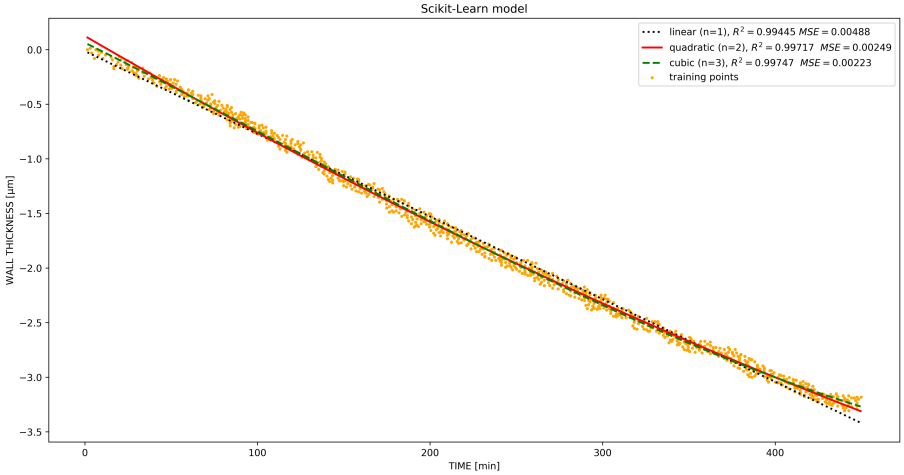

Figure 13: All linear regression curves

	Linear	Quadratic	Cubic
MSE	0.004881	0.002486	0.002225
R^2	0.994449	0.997172	0.997469

Table 2: Error of corrosion rate for acetic acid using Scikit-Learn

3.3 Tensorflow

As in the previous section, linear regression was used to analyze the corrosion coefficient, but in this case it was modeled in Tensorflow. Tensorflow is a low-level library compared to Scikit-Learn which is high-level library. So build model in Tensorflow is not that easy as in Scikit-Learn but we have more control over this. Linear regression modeling has begun by defining the variables in this case, weights (W) and bias (b), weights respond to the slope and bias is intercept in this model. Having certain variables and a specific placeholder for the input data, it was necessary to transform the data depending on the regression exponent. For simple linear regression, the data were presented in the form of a matrix (data.shape,1) for quadratic regression (data.shape,2) and for cubic regression (data.shape,3), the function *modify_input()* was used for this. After defining the model as $y = XW + b$ calculations were started. The cost function was MSE which was optimized by GradientDescentOptimizer.

3.3.1 Citric Acid

For citric acid using the Tensorflow model the following model was obtained:

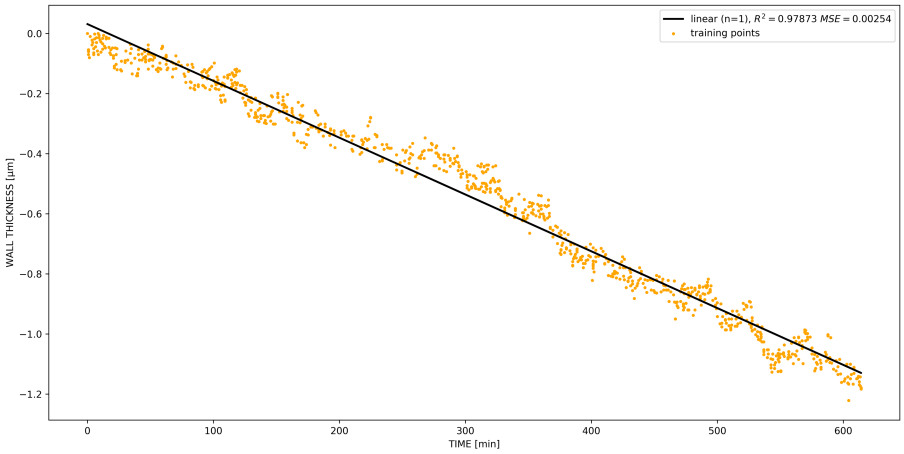

Figure 14: Linear regression

The weights and bias for Linear regression:

$$Weights : -1.16086$$
$$Bias : 0.0312$$

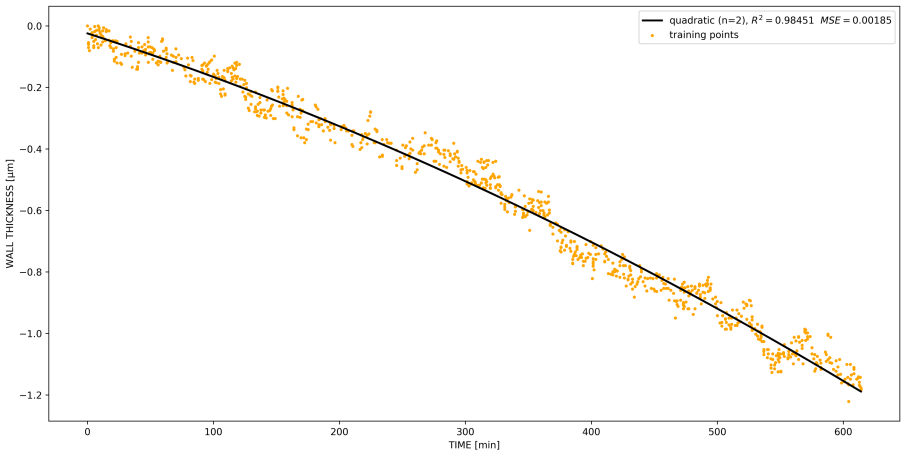

Figure 15: Quadratic regression

The weights and bias for Quadratic regression:

$$Weights : [-0.81216, -0.3527]$$
$$Bias : -0.0244$$

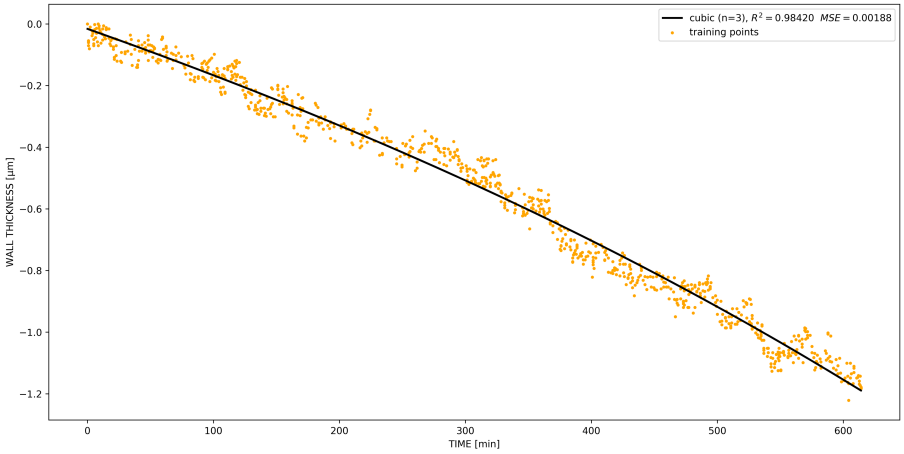

Figure 16: Cubic regression

The weights and bias for Cubic regression:

$$Weights : [-0.89067, -0.194658, -0.0885]$$
$$Bias : -0.01591$$

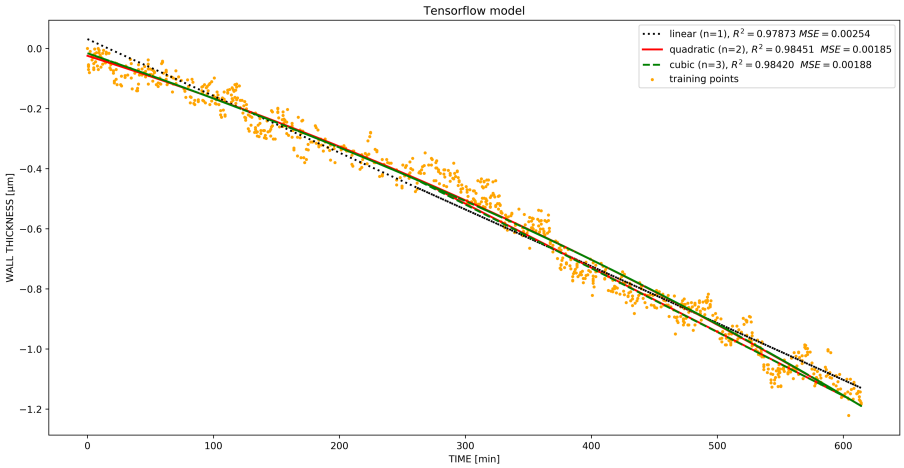

Figure 17: All linear regression curves

	Linear	Quadratic	Cubic
MSE	0.002536	0.001847	0.001972
R^2	0.978727	0.984512	0.98346

Table 3: Error of corrosion rate for citric acid using Tensorflow

3.3.2 Acetic acid

For acetic acid using the Tensorflow model the following model was obtained:

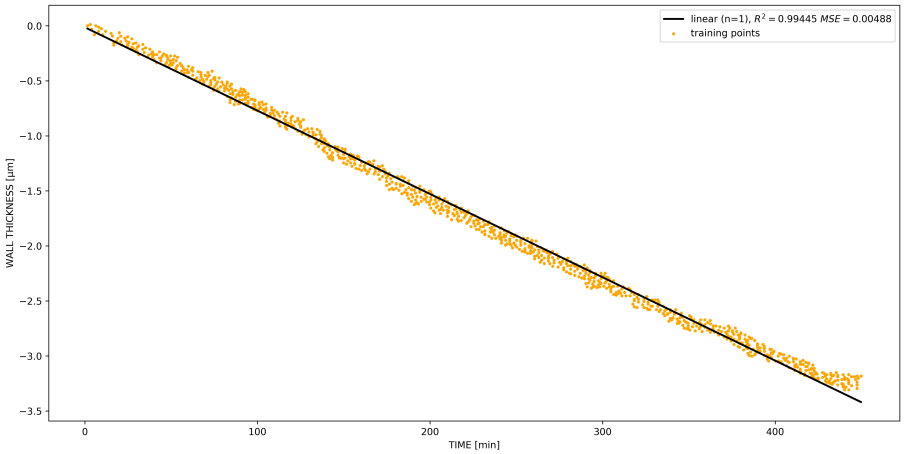

Figure 18: Linear regression

The weights and bias for Linear regression:

$$Weights : -3.40543$$
$$Bias : -0.01383$$

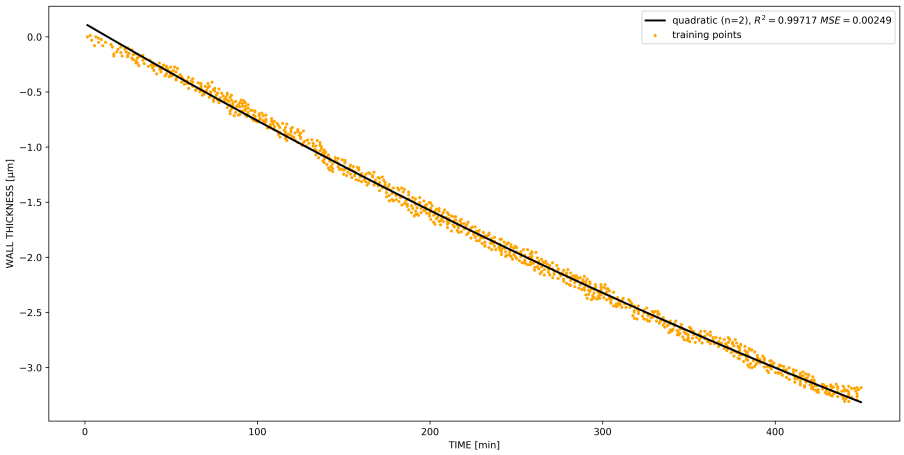

Figure 19: Quadratic regression

The weights and bias for Quadratic regression:

$$Weights : [-4.098285, 0.66455]$$
$$Bias : 0.1166$$

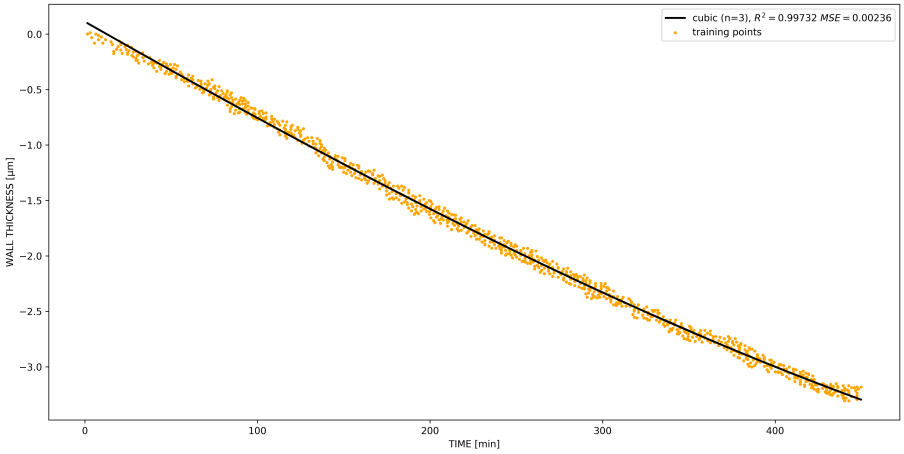

Figure 20: Cubic regression

The weights and bias for Cubic regression:

$$Weights : [-3.52975, -0.72753, 0.92288]$$
$$Bias : 0.06403$$

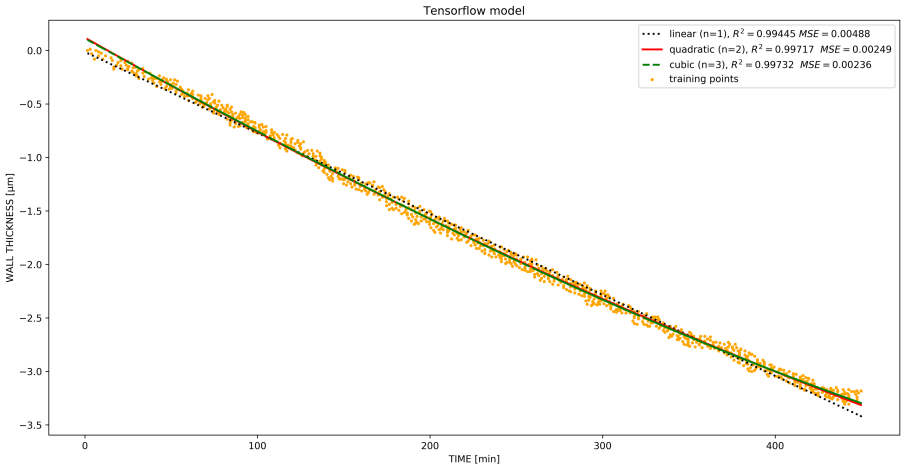

Figure 21: All linear regression curves

	Linear	Quadratic	Cubic
MSE	0.004881	0.002493	0.002225
R^2	0.994449	0.997164	0.997469

Table 4: Error of corrosion rate for acetic acid using Tensorflow

4 Conclusions

As mentioned at the beginning, machine learning is an extremely powerful prediction tool for linear as well as non-linear phenomena. For citric acid using the proposed model, the corrosion factor was -0.993 mm/year and for acetic acid it was -3.979 mm/year which is a result similar to that obtained by the authors of the article. For the linear regression shown, already the simple regression proved to be sufficiently accurate for the given data and the increase in the regression index does not significantly affect the prediction results.

References

[1] https://www.sciencedirect.com/science/article/pii/S1572665718300900

[2] https://apps.automeris.io/wpd/

[3] https://towardsdatascience.com/simple-and-multiple-linear-regression-in-python-c928425168f9

[4] Rodolfo B., *Building machine learning projects with tensorflow*, November 2016, Packt

[5] Hackeling Gavin., *Mastering Machine Learning with scikit-learn*, October 2014, Packt

[6] https://www.analyticsvidhya.com/blog/2018/03/introduction-regression-splines-python-codes/