POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Machine Learning Data Driven Approach for Predictive Maintenance of Process Units

Master Thesis in
Chemical Engineering

Author: **Francesco de Fusco**

Student ID: 10762320
Advisor: Prof. Flavio Manenti
Co-advisors: Andrea Galeazzi
Academic Year: 2021-2022

Abstract

This thesis work aims at developing a machine learning algorithm for a data driven predictive maintenance technique. Maintenance is a well known problem in industry that is responsible for economic and efficiency losses. The algorithm is meant to be integrated in the distributed control system of the Itelyum Regeneration plant in Pieve Fissiraga (LO). The development of such algorithm is carried out in Python programming language and developed with the free open source library Scikit Learn. The time series data is modeled through a Gaussian Process Regression and linear regression. The two approaches are than compared. The case study implemented describes the furnace of the thermal de-asphalting section of the plant. A key unit of the refinery plant for which maintenance and control must be ensured and optimize in order to ensure energy efficiency and safety and to avoid economic losses and downtime.

Keywords: chemical engineering, process unit, predictive maintenance, process furnace, machine learning, data driven, gaussian process, big data, industry 4.0

Abstract in lingua italiana

Questo lavoro di tesi mira a sviluppare un algoritmo di machine learning per un approccio data driven al problema della manutenzione predittiva. La manutenzione negli impianti è un problema ben noto in quanto essa è responsabile per ingenti perdite economiche e riduzione dell'efficienza delle unità. L'algoritmo è pensato per essere integrato con il sistema di controllo distribuito dell'impianto di rigenerazione Itelyum, situato a Pieve Fissiraga (LO). L'algoritmo è implementato attraverso il linguaggio di programmazione Python e sviluppato utilizzando Scikit Learn, una liberira gratuita e open source per il machine learning. I dati vengono modellati tramite metodi di regressione lineare e Gaussian Process e i due metodi vengono poi comparati. Il case study implementato riguarda la fornace nella sezione di de-asfaltazione termica dell'impianto. Un'unità chiave dell'impianto di raffinazione per la quale manutenzione e controllo devono essere assicurati in modo tale da garantire efficienza energetica, sicurezza ed evitare perdite economiche e tempi morti.

Parole chiave: ingegneria chimica, unità di processo, manutenzione predittiva, forno industriale, data driven, machine learning, gaussian process, big data, industria 4.0

Contents

1 | Introduction

1.1. Overview

The chemical industry has always been the joining ring between raw materials and consumers, allowing the development of the modern society as we know it. It is responsible for delivering goods, its a source of employment and it represents a playground for scientific improvements and applied research.

Chemical plants are often huge structures composed of hundreds of different units of large volumes for which safety and control must be ensured. On the other hand, the economic aspects shall not be neglected in order for the plant to remain competitive on the market. Last but not least, the environmental sustainability is something to account for, especially when dealing with chemicals.

The monitoring of operative conditions is a main aspect to ensure safety, optimal logistic, low environmental impact and in general high efficiency of the process. The monitoring is done by installing sophisticated control systems that elaborates the signals coming from the sensors. After many stages in the evolution of process control, the introduction of a distributed control system allowed an easy interconnection of plant controls. The data coming from the sensors are collected and gathered in a single interface for visualization. In the last decades, along with the interest in data analytic and with the development of affordable storage devices, chemical plants have began to stock the data for further elaboration. Indeed, data are informative of the process and by studying the history of the plant, it is possible to predict its future state to further increase its management and control.

These aspects shade light on a whole new way of managing a chemical plant by exploiting the full potential of the data through artificial intelligence and big data analytic techniques.

1.2. Research objective

Currently, Itelyum Regeneration Spa is embracing the era of big data starting from the installation of the Exaquantum monitoring system at its refinery plant in Pieve Fissiraga (LO). Nevertheless, the process of accessing, elaborating and showing the data is intricate and slow, making the system not exploitable at its full potential.

The aim of this thesis work is to develop an algorithm able to elaborate the data history coming from the furnace *PH-401B* operating in the thermal de-asphalting section of the plant. The work was mainly focused on the development of a ML model able to learn the structure of the time series explanatory of the process. The model is than used to return a long term forecast in order to predict the evolution of the process and plan for maintenance of the furnace. The algorithm should have the following features:

- Robustness. In order to deal with discrepancies of the model from the real data. These discrepancies are inevitable since data from the furnace are subjected to sudden changes due to process conditions.
- Easy to use. In order to overcome the distance between the data and the plant operators, the results should be obtained easily, the algorithm should have few input parameters.
- Insights. The algorithm should extract valuable information form the data otherwise not obtainable such as long term predictions and correlations between the data.
- Computationally affordable. The algorithm should run in no more than few hours in order to deliver new information every day.

The output of the algorithm is than showed in a dashboard to be used during daily operational meetings in order to have better insights on the unit. This system will help o better exploit the valuable information offered by the data in a fast and easy manner.

2 State of The Art

2.1. Background

The new era of Big Data (BD) is currently driving the secondary sector in what has been called "The Fourth Industrial Revolution" [18]. In particular, chemical industries are given an important opportunity for shifting towards smart manufacturing. Indeed, the large amount of data produced in a chemical plant merged with the use of software for Big Data Analysis (BDA) is the key to successful decision making and planning [6].

BD is an expression for massive data sets consisting of both structured and unstructured data that are particularly difficult to store, analyze and visualize [11]. Every chemical industry produces large amount of data acquired every second by the sensors that monitor variables such as pressure, temperature, density, level, flow, etc. Without mentioning all the data generated from finance, maintenance and communication during normal operations on the plant. Whereas previously these data were used instantly or partially stored, nowadays the industries have started to store the majority of the data thanks to the availability of more advanced storage devices [23]. This is surely the starting point to gain benefit from the massive amount of data generated over the years. Some of the analysis expected form BDA in chemical plants are:

- Descriptive: provide information on a historical process event that helps in understanding the nature of the process;
- Diagnostic: provide explanation on why a specific event occurred. Useful for hazard identification;
- Predictive: provides forecasting based on historical and current data. The forecasting can be short-term or long-term.

Under this perspective, it is clear that professional figures such chemical engineers should have competences in BDA in order to match the skills required by the so called Industry 4.0 [20]. Some of these skills are proficiency in programming languages such as MATLAB, Python, C++, MySQL and R, education in statistics and data analytic and knowledge

of software libraries for Machine Learning (ML) and Deep Learning [14]. With these competences, a process engineer would be able to perform BDA in order to:

1. Identify problems in product specifications such as if an off-spec product is an outlier or part of the process;
2. Identify relationship between several sensors by conducting multivariate analysis;
3. Identify the unit that can maximize the production;
4. Generate explanatory plots and graphs to represent the performance of a unit.

Such a professional figure could be of great importance in the management of a plant and the application of the mentioned tools of great effectiveness in operational meetings and decision making.

2.2. Predictive maintenance

The opportunities for chemical plants which will embrace the digital revolution are numerous. Among these, one of the most attractive is predictive maintenance. In contrast with other maintenance technique such as preventive or corrective, the aim of a predictive approach is the prediction of trends, behavior patterns and correlations by statistical or machine learning models for anticipating failure and ordinary operations and therefore reducing downtime [27].

The impact of maintenance is a well known problem in industry. In the field of the chemical industry, entire sections of the plants are periodically stopped for restoring units that undergo notorious aging phenomena such as coking, sintering or fouling. According to the 2018 Maintenance Survey by Plant Engineering, 35% of respondents said they spend more than 10% of their operating budgets on maintenance, and another 34% spend between 5% and 10%. Traditional maintenance strategies consist in regularly scheduled or occasional interventions. This means that when components wear down or break, the equipment is stopped until it is fixed. Abnormal event management is estimated to cost around 20 billion a year to the petrochemical industries, rating this as their number one problem to be solved [22]. On the other hand, predictive maintenance focuses on anticipating problems and therefore predict failures before they occur. Predictive maintenance is classified as [27]:

Physical based. The life cycle of the equipment is described by an analytical model whose inputs are data from the sensors measuring the conditions of the component.

Knowledge based. Uses expert systems approach or fuzzy logic when no physical model

is available.

Data driven. Development of statistical based models or artificial intelligence based models through machine learning.

The latest approach, that is of main interest for this thesis work, implies the analysis and elaboration of time series data that are descriptive of the unit life cycle. Through the employment of artificial intelligence based techniques, the time series is modeled in order to learn its structure and therefore forecast the future state of the unit.

Several physical based degradation models have been developed, such as the one dealing with sintering of the catalyst [12], fouling of heat exchangers [5] or coking of steam crackers [3]. Nevertheless, these models have been extrapolated form laboratory environments that are not fully explanatory of real world apparatus. The former issue is the main reason why these mechanistic models of degradation dynamics are rarely used in production environment. For this reason, the research has shifted to modelling time series data coming from process units with a non deterministic approach. Nevertheless, few advanced machine learning methods have been applied for the prediction of the life of chemical process equipment [4], leaving a gap in the literature. For example Wu et al. [25] have modeled fouling in heat exchangers through a classical statistical method such as the partial least squares regression; while Radhakrishnan et al. [16] and Aminian and Shahhosseini[2] have trained recurrent neural network also to predict an heat exchanger fouling process.

In recent years, Gaussian Process has received extensive attention in the world of machine learning due to its advantages in data modelling given by the Bayesian framework of this approach [26]. Indeed, these non parametric models are flexible enough to model highly complex data whilst prevent over fitting and also they aim at making predictions which quantify the uncertainty due to limitations in the quantity and quality of the data [7]. GPR was successfully applied in order to predict the quality of polypropylene at industrial level [8].

In order to train a GP model, it is necessary to have a previous knowledge about the shape of the dataset so to specify the covariance function with which the time series is modelled. Rasmussen and Williams, approached the structure modelling of complex kernels by observing the structure of the data and choosing the right combination of kernels according to the observations [17]. Nevertheless, this method can be tedious and time consuming especially if one does not have sufficient knowledge about the theory of GP. That is why Duvenaud et al. proposed an automatic kernel search procedure by the construction of a search tree [19].

3 | Methods

3.1. Time series

In mathematics, time series is a sequence of data points taken orderly in time:

$$\mathbf{y} = [y(t_1), y(t_2), ..., y(t_n)] \tag{3.1}$$

Nevertheless a time series reflects a continuous process, in practice data points are collected discretely and usually sampled uniformly. In Figure 3.1, it is possible to see a time series for which data were collected every $8h$ so that the vector of Equation 3.1 is composed by values taken at time $t_1 = 0$, $t_2 = 8$, $t_n = 8n$ hours, where n is the number of samples and y is a process variable such as CH_4 flow rate in this example.

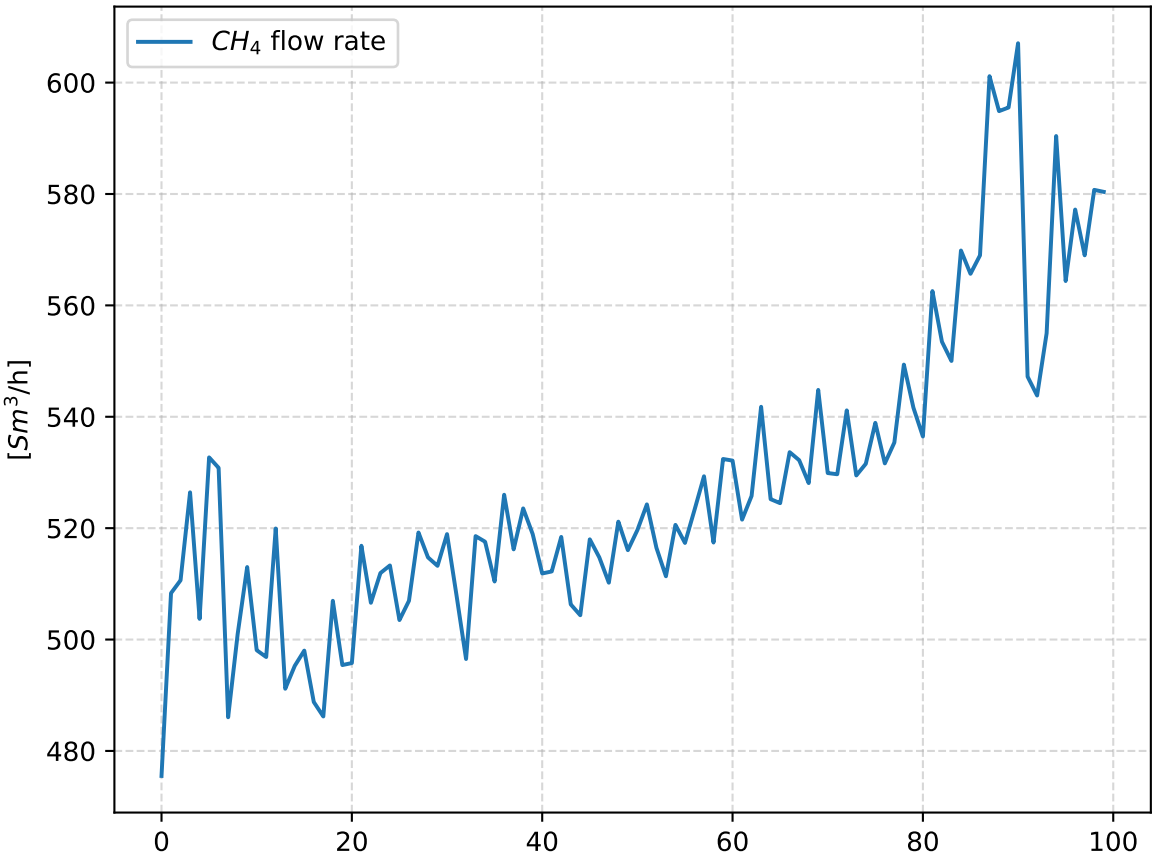


Figure 3.1: Example of a Time Series. The number of samples are reported on the x-axis, while the data points indicate the instantaneous methane consumption.

Some structure properties of a time series are reported hereafter.

Deterministic and Statistical Time Series. If the values of a time series are exactly determined by a mathematical function, the series is said to be deterministic. On the other hand, a non deterministic or statistical time series is one whose future values can only be described in terms of statistical distribution.

Stochastic process. Is a statistical phenomenon that evolves in time according to probabilistic laws. The time series to be analyzed may then be thought of as a particular realization, produced by the underlying probability mechanism, of the system under study [24]. In other words, we can study a time series as if it was the realization of a stochastic process where every observation is a random variable $[y(t_1), y(t_2), ..., y(t_n)]$ with probability distribution $P[y(t_1), y(t_2), ..., y(t_n)]$.

Mean. The mean of a time series can be defined as the expected value of the time series,

that is the level about which it fluctuates.

$$\mu = E[y_t] = \int_{-\infty}^{\infty} yP(y)dy \tag{3.2}$$

A time series can be stationary or non stationary depending on if the mean changes in time. In the second case, the time series will exhibit a mean shift.

Variance and Standard Deviation. The variance σ^2 is defined as

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mu)^2 \tag{3.3}$$

It is a measure of variability of the data and represents the average squared deviation from the mean. Since the variance is not expressed in the same units as the original data, for interpretation purposes the standard deviation σ is frequently used. It is obtained from the squared root of the variance.

Covariance. Measure of the statistical dependence between observations in a time series. It reflects the degree of correlation between data points at two different times. Given two observations at time s and t, the covariance is defined as

$$k(y_s, y_t) = E[(y_s - \mu_s)(y_t - \mu_t)] \tag{3.4}$$

3.1.1. Time series forecasting

A typical application of time series is structure modelling and forecasting. This is the process of understanding and mathematically define the structure of a time series. Once the structure of the time series is learned, it is possible to extrapolate future data for predictions. An example of time series forecasting is reported in figure 3.2.

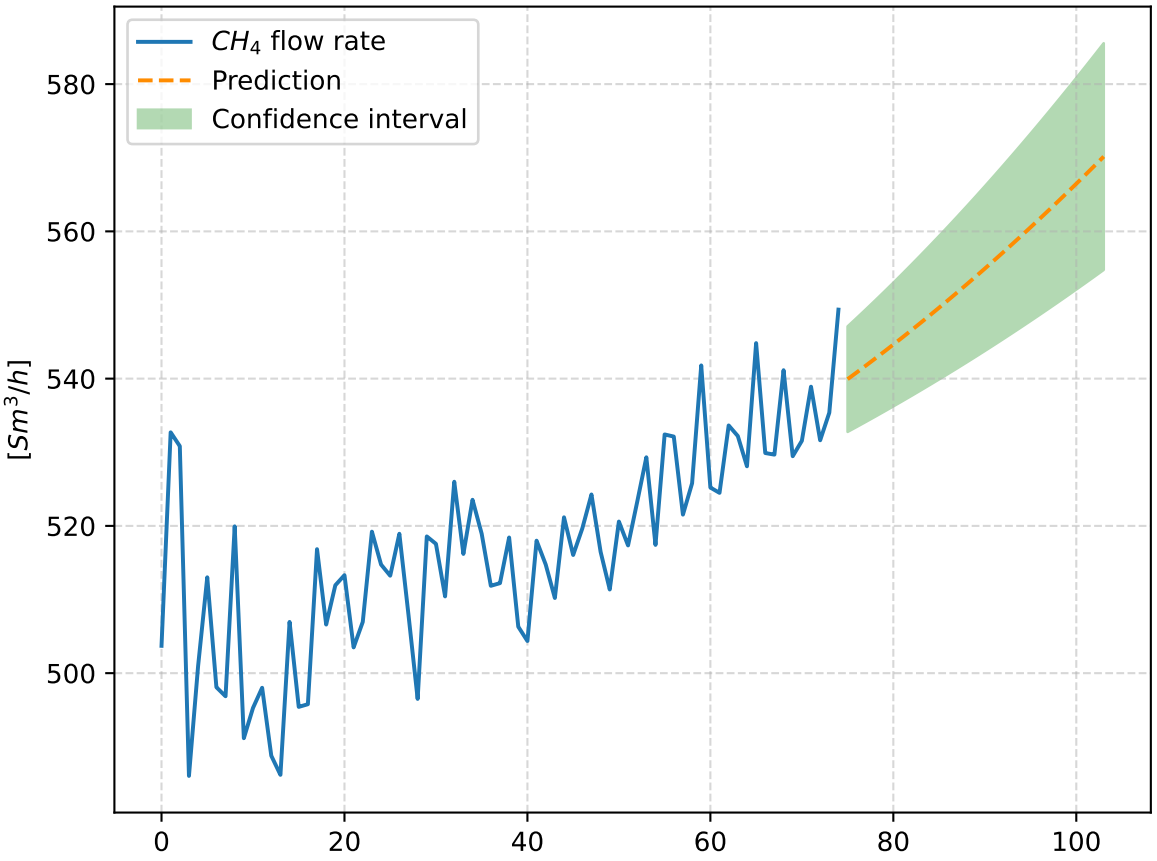


Figure 3.2: Example of a Time Series Forecasting. The prediction of the flow rate trend is represented by the orange curve. The confidence interval in green accounts for the uncertainty of the prediction.

The blue part of the time series is the observed data used to "learn" the structure of the model, while the orange part of the time series is the extrapolation of future data. The green region represents the interval within which the data will fall with 95% accuracy. It is possible to notice that the green region enlarges over time. This means that the prediction becomes more uncertain as the forecasting horizon becomes longer as one could expect.

The forecasting horizon is the ensemble of points in the future that we want to forecast and it can fall into three categories: short-term, medium-term and long-term. Where the former is closed to the present observations while the latter spans further away. The spans that classify the length of the horizon depend on the kind of process that one wants to describe. Long term forecasting is the most difficult. An effective way to obtain accurate long term predictions would be to have a a physical model describing the system. However, such a model is not always available and also its simulation is time consuming and requires appropriate computational power. For these reasons, statistical models are

favoured, with the drawback of loosing accuracy faster.

In order to correctly forecast a time series, one should be able to find some regularities in its structure. Stationarity is an example of regularity that is often assumed because of several advantageous mathematical properties such as the independence of the mean form time. On the other hand, for many time series the stationarity property does not hold and some structures like trend and seasonality are found. In figure 3.3 we can see an example of time series that exhibit an increasing trend and an irregular seasonality.

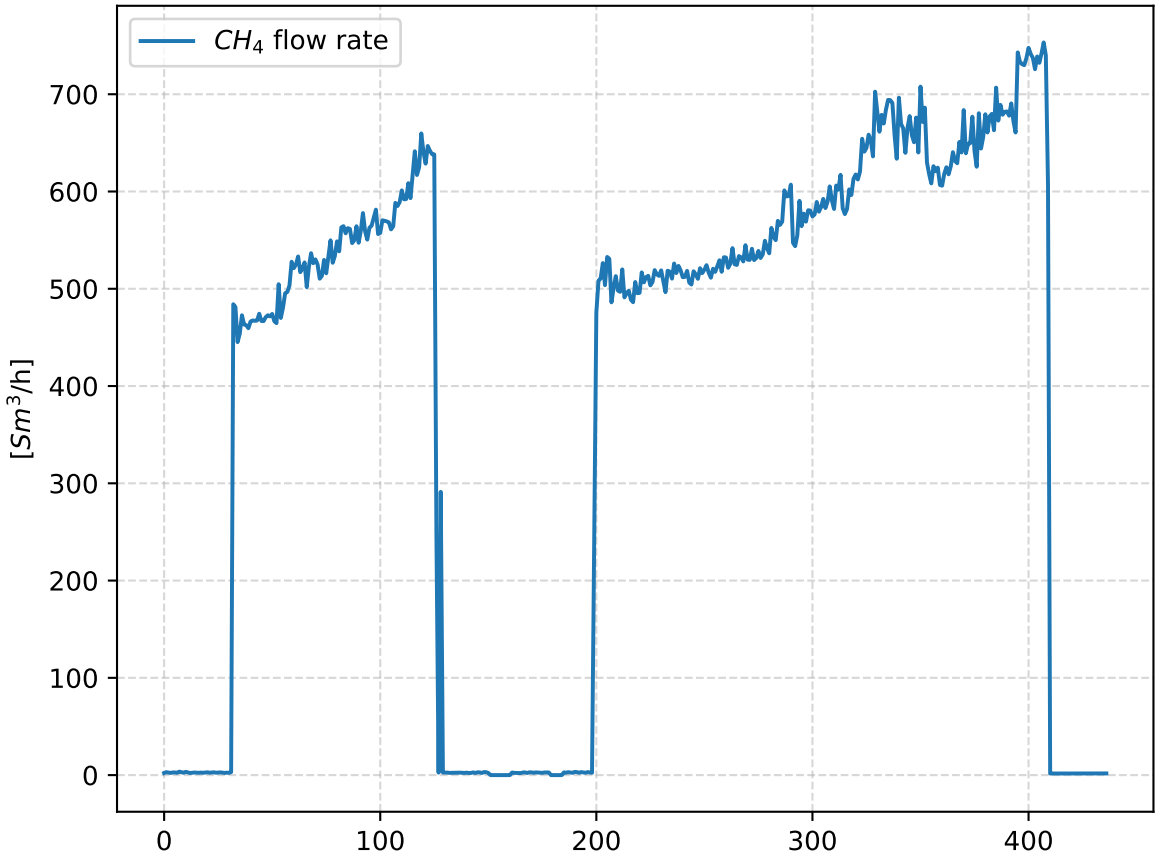


Figure 3.3: Example of a time series that exhibits patterns in its structure: the trend is given by the increasing value of the flow rate, while seasonality is given by the repeating patterns.

3.1.2. Modelling approach of time series

Given the difficulty to describe a system with a physical model, statistical models are used the most. There are different approaches used to explore the structure of a time series. The most common modelling approaches are Linear Regression, Auto Regressive

Integrated Moving Average model (ARIMA) and Exponential Weighted Moving Average (EWMA). On the other hand, due to the increasing interest in Artificial Intelligence (AI) and easier access to high computational power and open source algorithms, Artificial Neural Networks (ANN) and other ML models such as Gaussian Process Regression (GPR) are gaining momentum.

A general time series model is given by

$$\mathbf{y}(t) = f(\mathbf{X}(t); \boldsymbol{\theta}) + \boldsymbol{\epsilon}(t) \tag{3.5}$$

where $\mathbf{y}(t)$ is the vector of values observed in time. This latter is function of the regressors expressed by the design matrix $\mathbf{X}(t)$ and the vector of parameters $\boldsymbol{\theta}$. The model also takes into account the noise term $\boldsymbol{\epsilon}(t)$ modeled as Independent Identically Distributed (IID) with a Gaussian distribution with mean 0 and variance σ^2,

$$\epsilon \sim \mathcal{N}(0, \sigma^2). \tag{3.6}$$

The aim of a time series modelling task is to "learn" the optimal values of $\hat{\boldsymbol{\theta}}$ from the available data. In this way it will be possible to extrapolate future data at time $t = t + h$ by substituting $\hat{\boldsymbol{\theta}}$ in Equation (3.5). In this work, the structure modelling of the time series is performed by means of the linear regression and Gaussian process regression.

3.2. Linear regression

A linear regression model is a straightforward and effective model that is often able to describe adequately the relation between the data. An important class of linear regression is the polynomial regression, that is the one considered in this work. When referring to linear regression, polynomial is implied.

Due to the the simple mathematical formulation and the low computational power required for its learning, linear regression is used in many engineering and science applications [13]. Depending on the number of regressors implied, it is possible to distinguish simple and multiple linear regression models. The resulting forecasting model of the former is

$$y = \theta_0 + \sum_{i=1}^{n} \theta_i x_1^i + \epsilon \tag{3.7}$$

where n is the maximum order of the polynomial with which we want to describe our model.

In case the data result from the combination of more than one regressor, multiple linear regression is employed and the forecasting model becomes

$$y = \theta_0 + \sum_{i=1}^{n} \theta_i x_1^i + \sum_{i=1}^{n} \theta_j x_2^j + \sum_{i,j=1}^{n} \theta_{ij} x_1^i x_2^j \tag{3.8}$$

Such a model is able to capture more complex patterns since it s able to model possible interactions between two variables.

In general, any regression model that is linear in the parameters θ is a linear regression model regardless of the shape of the surface that it generates. While a multiple linear regression model can capture wiggles in the data, a simple linear regression will only be able to describe the mean of the time series and therefore capture its trend. This aspect makes the linear regression approach suitable for long term predictions.

3.2.1. Training of a linear regression model

With the expression "training of a linear regression model", it is intended the estimation of the values of the regressors θ_i. Let us consider the case of a simple linear regression problem where we have n pairs of observations $(t_1, y_1), ..., (t_n, y_n)$ to which we want to fit the model $y = \theta_0 + \theta_1 t + \epsilon$. The method used to estimate the parameters that best fit the observations is the least squares. It consist in the minimization of the sum of the squares of the deviation of the observations from the true regression line. This sum of the deviation is expressed as

$$L = \sum_{i=1}^{n} \epsilon_i^2 = \sum_{i=1}^{n} (y_i - \theta_0 - \theta_1 t_i)^2 \tag{3.9}$$

Therefore, the optimal values $\bar{\theta_0}$ and $\bar{\theta_1}$ of the parameters must satisfy

$$\frac{\partial L}{\partial \theta_0} = -2 \sum_{i=1}^{n} (y_i - t\bar{\theta_0} - \bar{\theta_1} t_i) = 0 \tag{3.10}$$

$$\frac{\partial L}{\partial \theta_1} = -2 \sum_{i=1}^{n} (y_i - t\bar{\theta_0} - \bar{\theta_1} t_i) t_i = 0 \tag{3.11}$$

Equation 3.10 is solvable analytically.

If we identify the values estimated by the model with

$$\bar{y_i} = \bar{\theta_0} + \bar{\theta_1} t_i, \tag{3.12}$$

than the quantity $e_i = y_i - \bar{y_i}$ is called residual and it describes the error in the fitting of the model.

Once the model is trained, it is possible to extrapolate unseen observations by substituting future values of t in Equation 3.12. Besides, it is useful to have a prediction interval on future observations y^* in order to take into account uncertainty in the prediction. The prediction interval is defined as

$$\bar{y_i^*} = \pm \delta \sigma_{err} \tag{3.13}$$

where σ_{err} is the standard deviation of the error and δ is set in accordance with the degree of confidence with which the interval is computed. For a 95% confidence interval, δ is set to 1.96, while for a 75% confidence interval $\delta = 1.15$.

3.3. Gaussian Process

The GP regression is a model based on few restrictive assumptions on the function f. A GP regression only assumes that the function f is smooth enough so that mean and covariance function exist. Furthermore, the non parametric nature of the model implies that the GP does no rely on any parametric model assumption, instead the GP is flexible with the capability to adapt the model complexity as more data arrive [17]. GPR finds its roots in the Bayesian modelling approach since a GP can be used as a prior probability distribution over functions in Bayesian inference.

In practice, given two data points, the GPR approach consists in giving a prior probability distribution to every possible function that could interpolate these points. Higher probabilities are given to functions that we consider to be more likely, for example because they have some properties with respect to other functions [17]. This reasoning is extended to every data point in the time series. The combination of the data with the prior distribution leads to the posterior distribution that is the result of the regression. Under this assumptions, the specification of the prior distribution is needed before fitting and the learning process is the problem of finding the proper prior and its properties.

In other words, we build a probabilistic model for the function itself. Indeed by considering a set of inputs $[t_1, t_2, ..., t_n]$, we assume that the function values $[y(t_1), y(t_2), ..., y(t_n)]$ are

distributed according to a multivariate Gaussian distribution with mean μ and covariance k. Therefore, given two data points $y(t_1)$ and $y(t_2)$ the GPR model is defined as

$$\begin{bmatrix} y(t_1) \\ y(t_2) \end{bmatrix} \sim \mathcal{N}\left[\begin{bmatrix} \mu(t_1) \\ \mu(t_2) \end{bmatrix} ; \begin{bmatrix} k(t_1,t_1) & k(t_1,t_2) \\ k(t_2,t_1) & k(t_2,t_2) \end{bmatrix} \right] \tag{3.14}$$

Through this technique, different kind of functions can be modelled by varying the mean and covariance function of the normal distribution.

The benefits of a GPR are the following:

- Flexibility. The high flexibility of GPR through kernel modification allows to model time series with a broad range of complexity.
- Probabilistic. Prediction of a GPR model is a distribution of function and not a punctual forecast. Therefore, the output of the GPR is the punctual datum (the mean of the distribution) plus its uncertainty (the variance of the distribution).
- Robustness. Presence of outliers and noise in the data set do not strongly affect the result of the GPR due to the probabilistic assumptions.

On the other hand, computational power can be an issue with GPR since it scales as $O(n^3)$ with the data. Since GPR is directly inferred from Bayesian modelling, this latter will be introduced in the next section.

3.3.1. Bayesian modelling

Bayesian probability is an interpretation of the concept of probability as a degree of belief in an event. The degree of belief is based on a prior knowledge about the event. A prior hypothesis is made without looking at the data and a prior probability is assigned to this hypothesis. This prior probability is than updated to a posterior probability in the light of new data acting as evidence. The Bayesian modelling is based on the Bayes' Theorem by the statistician Thomas Bayes.

Let us consider a linear regression model $\mathbf{y} = \boldsymbol{\theta} f(\mathbf{t})$. Following the notation used in section 3.1.2 we have that

$$\begin{aligned} &\mathbf{t} = [t_1, ..., t_n] \\ &f(\mathbf{t}) = \theta_0 + \theta_1 t_1 + ... + \theta_n t_n \\ &\boldsymbol{\theta} = [\theta_1, ..., \theta_n] \end{aligned}$$

where $\mathbf{t}$ is the vector of input and $\boldsymbol{\theta}$ is the vector of parameters we want to infer from the data. Than the inference is done through the Bayesian Theorem

$$p(\boldsymbol{\theta}|\mathbf{t},\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{t},\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{y}|\mathbf{t})} \tag{3.15}$$

where the denominator, a normalization factor, can be computed as

$$p(\mathbf{y}|\mathbf{t}) = \int p(\mathbf{y}|\mathbf{t},\boldsymbol{\theta})p(\boldsymbol{\theta})d(\boldsymbol{\theta}) \tag{3.16}$$

We can refer to each component of Equation 3.15 as

$$posterior = \frac{likelihood \times prior}{evidence} \tag{3.17}$$

where the term at the denominator is a normalization factor. If the normalization factor is disregarded, the Bayes' Theorem becomes

$$p(\boldsymbol{\theta}|\mathbf{t},\mathbf{y}) \propto p(\mathbf{y}|\mathbf{t},\boldsymbol{\theta})p(\boldsymbol{\theta}) \tag{3.18}$$

Assuming that the observations are independent, the likelihood can be written as a Gaussian distribution

$$\begin{aligned} p(\mathbf{y}|\mathbf{t},\boldsymbol{\theta}) &= \prod_{i=1}^{n} p(y_i|t_i,\boldsymbol{\theta}) = \prod_{i=1}^{n} \frac{1}{\sigma_n\sqrt{2\pi}} \exp\left(-\frac{(y_i - t_i^\top\boldsymbol{\theta})^2}{2\sigma_n^2}\right) = \\ &= \frac{1}{(2\pi\sigma_n^2)^{n/2}} \exp\left(-\frac{1}{2\sigma_n^2}|\mathbf{y} - \mathbf{t}^\top\boldsymbol{\theta}|^2\right) \end{aligned} \tag{3.19}$$

Also, we need to make an hypothesis about the prior over the parameters that expresses our beliefs about the parameters before any observation is taken. We assume that the prior is Gaussian distributed with zero mean and known covariance matrix K

$$p(\boldsymbol{\theta}) \sim \mathcal{N}(0, K) \tag{3.20}$$

The choice and role of the prior will be better discussed in section 3.3.3.

By substituting Equations 3.19 and 3.20 in Equation 3.18, we obtain

$$p(\boldsymbol{\theta}|\mathbf{t},\mathbf{y}) \propto \exp\left(-\frac{1}{2}(\boldsymbol{\theta}-\bar{\boldsymbol{\theta}})^\top(\frac{1}{\sigma_n^2}\mathbf{t}\mathbf{t}^\top + K^{-1})(\boldsymbol{\theta}-\bar{\boldsymbol{\theta}})\right) \tag{3.21}$$

where $\bar{\boldsymbol{\theta}} = \sigma_n^{-2}(\sigma_n^{-2}\mathbf{t}\mathbf{t}^\top + K^{-1})^{-1}\mathbf{t}\mathbf{y}$. It is possible to recognize that the form of the posterior distribution is a Gaussian with mean $\bar{\boldsymbol{\theta}}$ and covariance matrix A^{-1}

$$p(\boldsymbol{\theta}|\mathbf{t},\mathbf{y}) \propto \mathcal{N}(\bar{\boldsymbol{\theta}}, A^{-1}) \tag{3.22}$$

where $A = \sigma_n^{-2}\mathbf{t}\mathbf{t}^\top + K^{-1}$.

In order to make predictions all the possible parameter values are averaged and weighted by their posterior probability. Thus, it is possible to predict the unseen data y^* as

$$\begin{aligned} p(\mathbf{y}^*|\mathbf{t}^*,\mathbf{y}) &= \int p(\mathbf{y}^*|\mathbf{t}^*,\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{t},\mathbf{y})d\boldsymbol{\theta} = \\ &= \mathcal{N}\left(1\frac{1}{\sigma_n^2}\mathbf{t}^{*\top}A^{-1}\mathbf{t}\mathbf{y}, \mathbf{t}^{*\top}A^{-1}\mathbf{t}^*\right) \end{aligned} \tag{3.23}$$

It is possible to see how prediction made using the Bayesian linear regression differs from the classical linear regression. Indeed in the former, the average of all possible values of the parameters based on their probability of existence are taken into account.

3.3.2. Gaussian Process Regression

In section 3.3.1, the function f used to fit the data was obtained through the inference of the parameters $\boldsymbol{\theta}$. However, the Gaussian Process Regression employs a different but equivalent way to obtain the same result overcoming the limits of a parametric model. Indeed, the GPR considers inference directly in the function space by describing the distribution over functions.

A GP is defined completely by its mean function $m(\mathbf{t})$ and covariance function $k(\mathbf{t},\mathbf{t}')$ so that it can be written as

$$f(\mathbf{t}) \sim \mathcal{GP}(m(\mathbf{t}), k(\mathbf{t},\mathbf{t}')) \tag{3.24}$$

In realistic modelling situations we have to deal with noisy observations. The practice used to model noise in time series was already discussed in section 3.1.2. For what concerns GPR, the noise term must be taken into account when specifying the prior. Therefore, assuming that σ^2 is the variance of the noise, the prior of the noisy observations becomes

$$k(\mathbf{y}) = K(t, t') + \sigma^2 I \tag{3.25}$$

where $K(t, t')$ is the covariance matrix and I is the identity matrix. By following the same procedure seen in section 3.3.1 we can write the distribution of the prior as

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}^* \end{bmatrix} \sim \mathcal{N}\left(m(\mathbf{t}); \begin{bmatrix} k(t,t) + \sigma^2 I & k(t,t^*) \\ k(t^*,t) & k(t^*,t^*) \end{bmatrix} \right) \tag{3.26}$$

and the predictive equation as

$$p(\mathbf{f}^*|\mathbf{t}, \mathbf{y}, \mathbf{t}^*) \sim \mathcal{N}(\bar{\mathbf{f}}^*, k(\mathbf{f}^*)) \tag{3.27}$$

and the prediction interval can be computed as $m(t) = \pm\sigma_{err}(t)$.

In order to grasp the concept, it may be useful to give a visual example of the GPR. Given a data set $\mathcal{D} = [(t_1, y_1), ..., (t_{10}, y_{10})]$ of ten observations, imagine we want to find the functions that better interpolates the observations. At first we specify the prior, a group of functions that is more likely to interpolate the observations due to its properties. The combination of the prior and the data leads to the posterior distribution over the functions. If more data points are added, the parameters adjust itself to pass through the points and the uncertainty decreases. This situation is illustrated in figure 3.4.

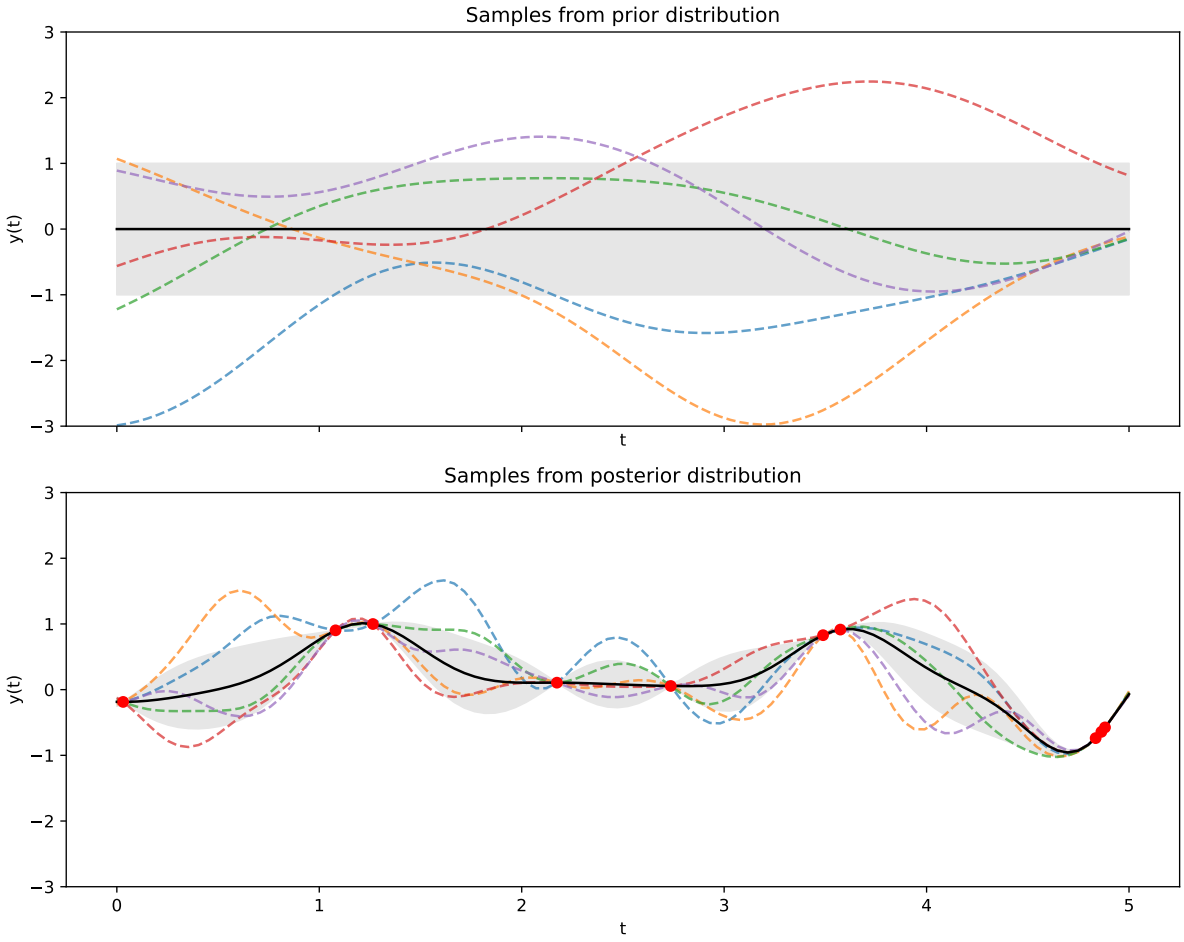


Figure 3.4: Illustrative example of prior and posterior distribution. In the first panel it is shown the sample drawn from the prior distribution while in the second panel the sample after that the data points are added. The dashed lines are the samples from the distributions of functions while the solid line is the mean prediction. The shaded region is twice the standard deviation at each point and represents the uncertainty of the model. It is possible to see how the uncertainty reduces near the observations

In the first panel we can see the sample drawn from the prior distribution while in the second panel is the sample after that the data points are added. The dashed lines are the samples from the distributions of functions while the solid line is the mean prediction. The shaded region is twice the standard deviation at each point and represents the uncertainty of the model. It is possible to see how the uncertainty reduces near the observations.

Since the GP is not a parametric model, one does not have to worry if it is possible for the model to fit the data. While this would be an issue when trying to fit a linear model on non linear data. On the other hand, the specification of the prior is important since it

represents the properties of the functions considered for inference. These properties are induced by the covariance function $k(t, t')$. Once the covariance function is specified it is possible to change some properties of the function to better fit the data. Therefore, the learning step in GPR is the problem of finding the suitable covariance function and its parameters. The learning process will be discussed in details in section...

3.3.3. Covariance function

In the previous section we have seen the importance in the GPR to specify a prior distribution over functions. That is a prior belief over the kind of functions that we expect to observe before seeing any data. This prior belief is specified by defining the covariance function $k(t, t')$ also known as kernel. In time series, two points with inputs t and t' that are close, are likely to return values $f(t)$ and $f(t')$ that are similar. Under the GP view, is the kernel function that expresses this similarity. In general, not all the functions with input variable t and t' are valid kernels. Indeed, a valid covariance function must have a positive semi-definite covariance matrix. This section will give an overview on the most common kernels and their properties and how to combine them in order to model complex patterns. The term kernel and covariance function will be used interchangeably.

Linear Kernel. The linear (LIN) kernel, also known as the Dot Product kernel, it is the simplest kind of covariance function and it is defined as

$$k(t, t') = \sigma^2 + t \cdot t' \tag{3.28}$$

The variance σ^2 determines the intercept of the function as illustrated in figure 3.5. If the variance is null, we call it homogeneous linear kernel. If only the linear kernel is used in a GPR, than it is like doing a Bayesian linear regression as the one in section 3.3.1. That is why the linear kernel is often used to build more complex covariance functions as we will see in section... . The linear kernel is non-stationary since it depends on the absolute location of the inputs $t \cdot t'$ and not on their relative position $|t - t'|$.

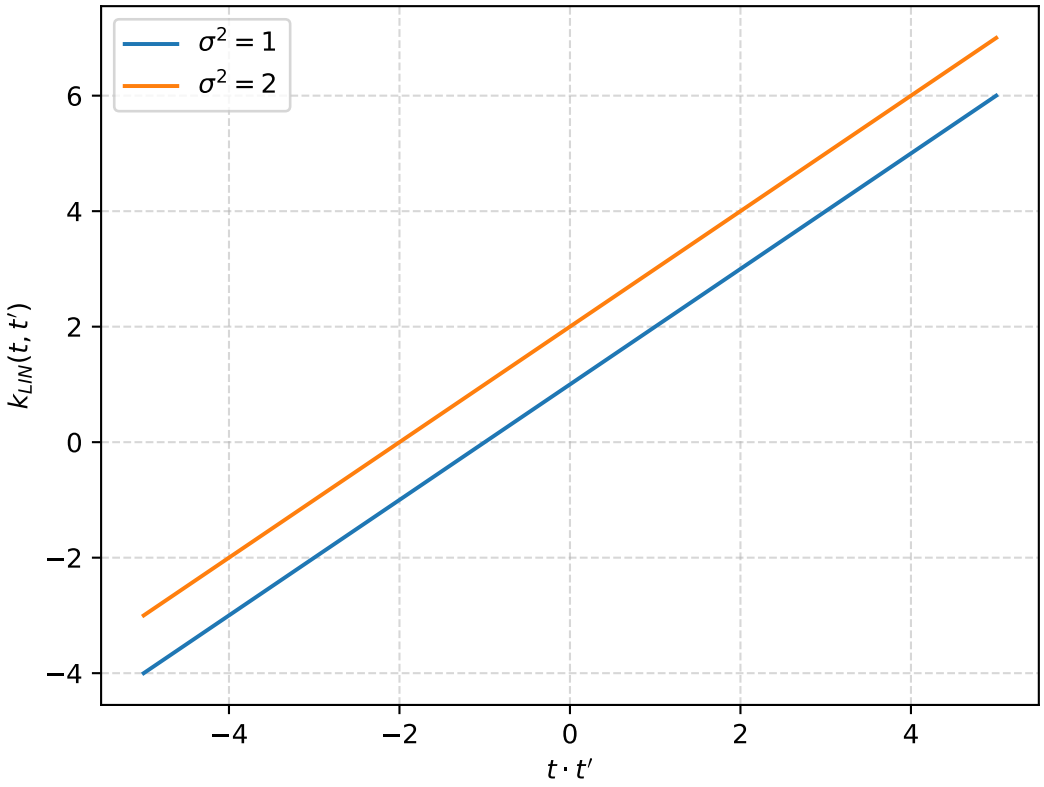


Figure 3.5: Linear kernel on varying of σ^2.

Radial Basis Function. This covariance function is also known as the Radial Basis Function (RBF) kernel. The RBF kernel is widely used in GPR since it can be integrated against many functions thanks to its properties. It is a non stationary kernel and it is defined as

$$k(t, t') = \sigma^2 \exp\left(-\frac{(t - t')^2}{2\lambda^2}\right) \tag{3.29}$$

The length scale parameter λ determines the length of the wiggles in the function. On the other hand, the variance σ^2 controls the average distance of the function from its mean. A small length scale causes a rapid change in the wiggles of the function making it quickly decaying to zero when forecasting data points outside the data set. Conversely, large values of λ make the function smoother due to the slower change in wiggles. RBF functions with high values of λ can predict on longer horizons.

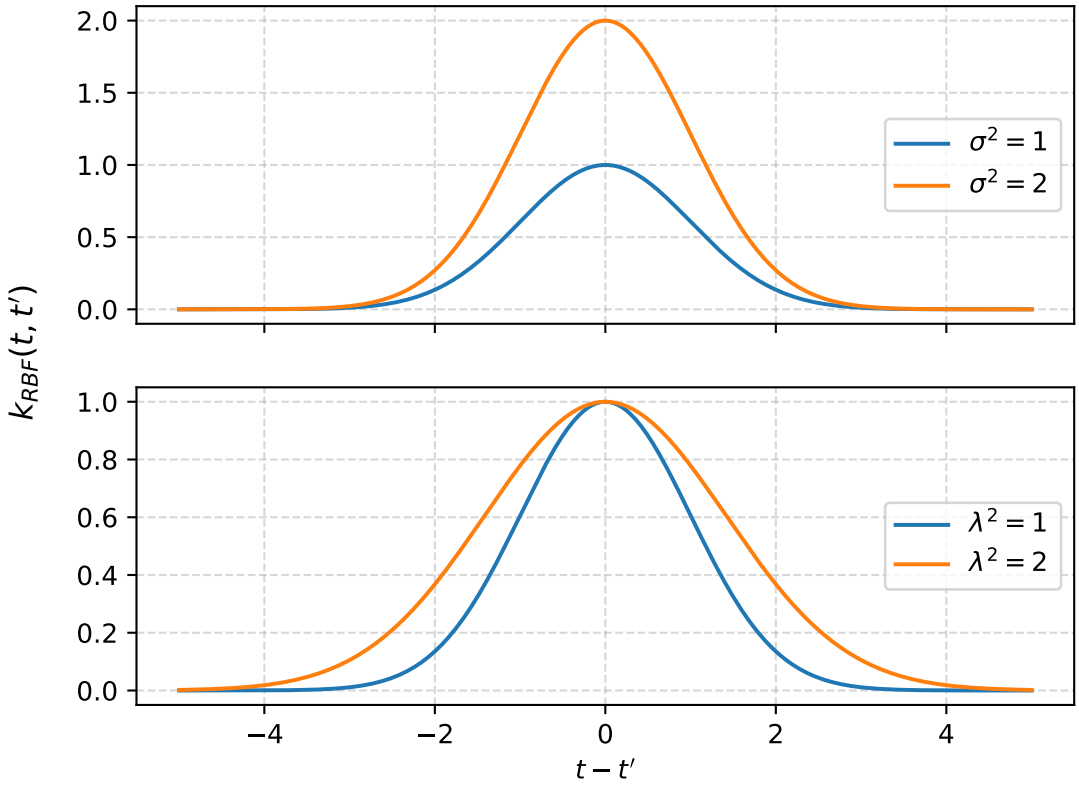


Figure 3.6: RBF kernel on varying of σ and λ. In the first figure the value of λ is fixed at $\lambda = 1$, while in the second figure the value of σ is fixed at $\sigma = 1$.

Periodic Kernel. The Periodic (PER) kernel is defined as

$$k(t, t') = \sigma^2 \exp\left(-\frac{sin^2(\frac{\pi|t-t'|}{p})}{2\lambda^2}\right) \tag{3.30}$$

This kernel is able to model a function that has a periodic structure. The terms λ and σ^2 have the same effect on the periodic kernel as the one on the RBF kernel. The term p is the period of the function. Larger values of p determines longer distances between repetitions of the function.

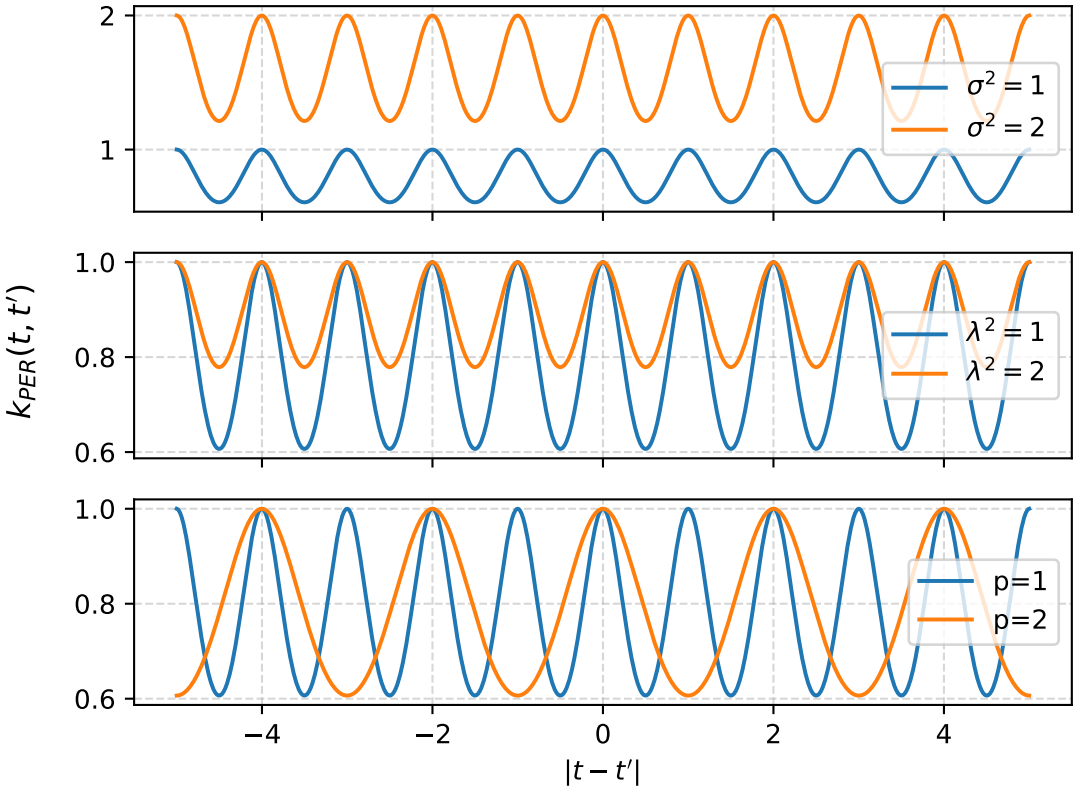


Figure 3.7: PER kernel on varying of σ, λ and p. When not specified, the values of the parameters are fixed to 1.

Rational Quadratic Kernel. The Rational Quadratic kernel (RQ) is defined as

$$k(t, t') = \sigma^2 \left(1 + \frac{(t-t')^2}{2\alpha\lambda^2}\right)^{-\alpha} \tag{3.31}$$

This kernel is the equivalent of adding together many RBF kernels with different length scales. This allows to model functions that vary smoothly across many length scale. In addition to the λ and σ^2 parameters, the RQ kernel presents the parameter α called power of the kernel. This latter defines how quick the change between length scales is. For $\alpha \to +\infty$, the RQ kernel tends to the RBF kernel.

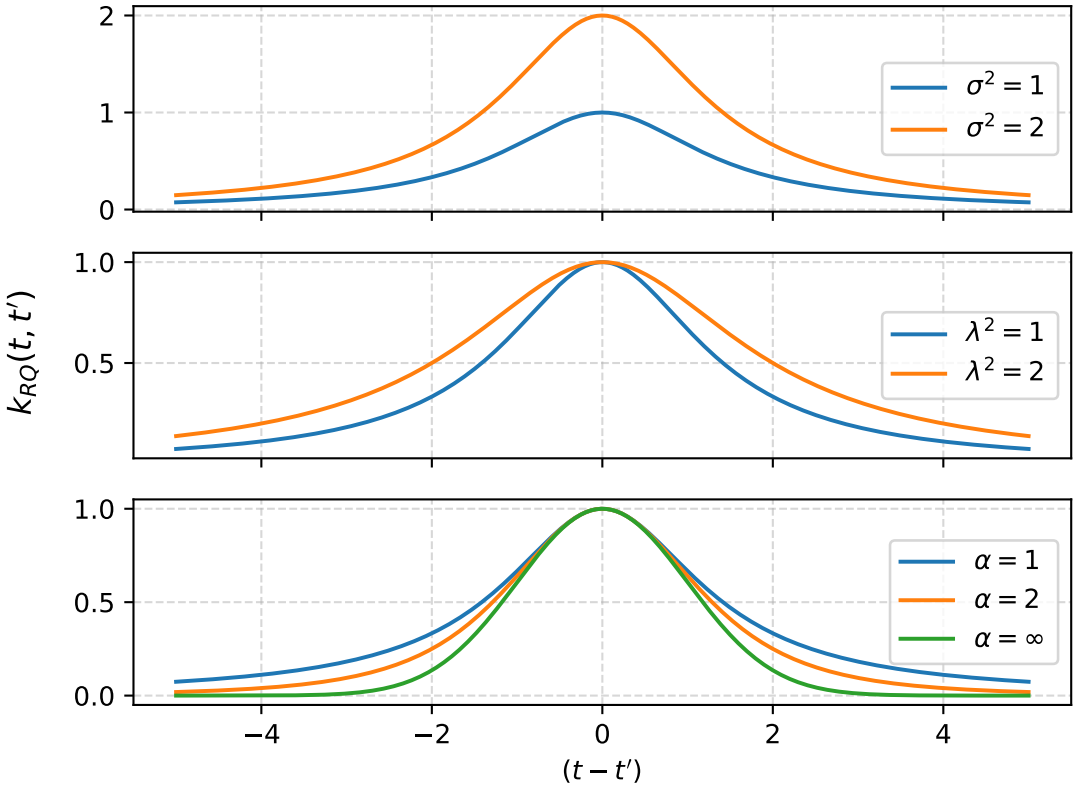


Figure 3.8: PER kernel on varying of σ, λ and α. When not specified, the values of the parameters are fixed to 1. It is possible to observe the equivalence between the RBF kernel and the PER kernel when $\alpha \to +\infty$: the green function is the same as the blue function in the upper panel of figure 3.6.

Composition of Complex Kernels. There could be situations in which a more complex covariance function is needed. For example, when the time series exhibits a trend or a parabolic structure. Since the definition of a completely new covariance function is not a trivial problem, a solution is to combine different base kernels in order to obtain a complex kernel. Considering that the resulting covariance matrix must be positive semi-definite as stated in section 3.3.3, the operations used to build complex kernels are addition and multiplication. With the former, the characteristic of each added kernel is retained. On the other hand, the properties of the kernels are fused together with the multiplication. In figure 3.9 it is possible to see the effects of the operations on RBF, LIN and PER kernels.

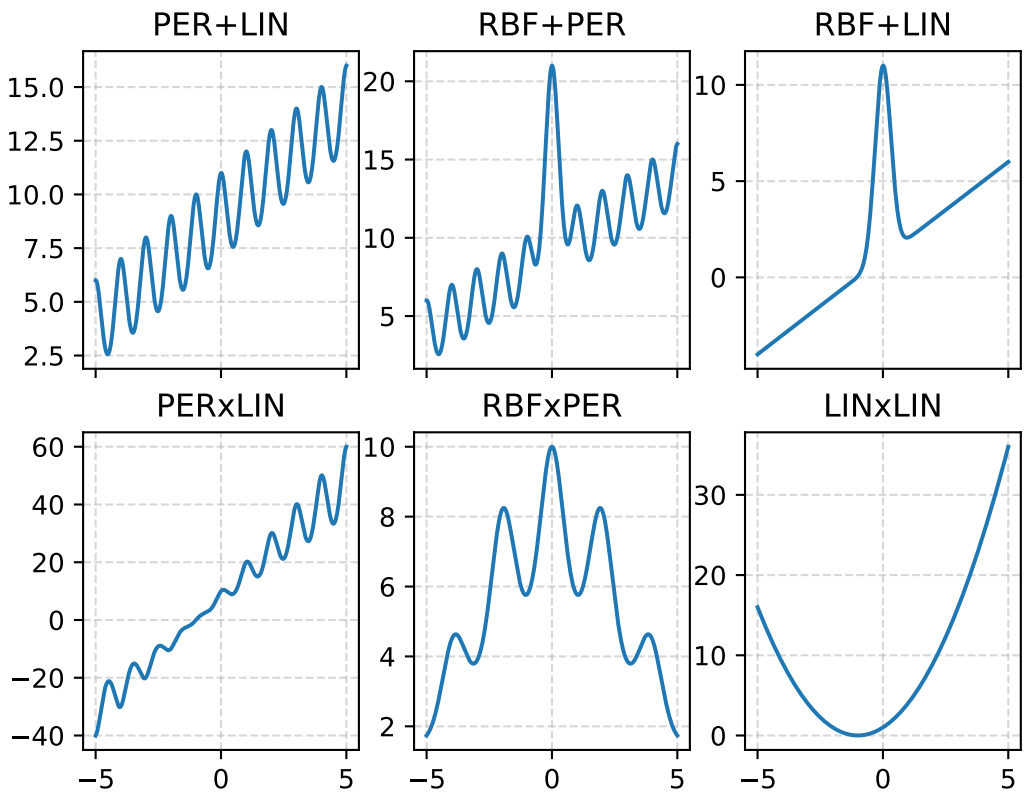


Figure 3.9: Composition of complex kernels starting from base kernels. The multiplication of n linear kernels returns the effect of a polynomial of degree n. he combination of a PER and RBF kernel is able to model patterns in which the characteristic length scale varies along the axis.

From the figure above, it is possible to see the effects of the different kernels. Of great interest is the effect of the linear kernel through which the modelling of non linear trend is possible. Indeed, the multiplication of two linear kernels results in a curvilinear structure. On the other hand, it is evident that the combination of a PER and RBF kernel is able to model patterns in which the characteristic length scale varies along the axis.

3.3.4. Training of a GP model: hyperparameters selection

In the previous section, we have seen that the shape of the covariance functions depend on properties such as the length scale λ, the variance σ^2 or the period p. We will refer to these properties as hyperparameters. Given a complex covariance function composed by the addition and multiplication of two kernels k_1 and k_2, the ensemble of hyperparameters

is represented by the vector $\boldsymbol{\theta} = (\sigma_1, \lambda_1, \sigma_2, ...)$.

Since the hyperparameters are responsible for shaping the covariance function so that it fits the data the best, the training of a GP model implies searching for the best family of hyperparameters. Actually, the best hyperparameters could be selected manually if one has sufficient knowledge of the time series to model. Nevertheless, this manual approach is impractical in most cases, especially when dealing with real data from a chemical plant. That is why a systematic approach is used. This latter consists in the maximisation of the evidence as defined in Equation 3.16, also called marginal likelihood. Since the marginal likelihood usually results in small numbers, the logarithm is applied so to avoid numerical instabilities. The integral can be computed analytically and the logarithm of the result is the so called Log Marginal Likelihood (LML):

$$log\left(p(\mathbf{y}|\mathbf{t}, \boldsymbol{\theta})\right) = -\frac{1}{2}\mathbf{y}^\top K_y^{-1}\mathbf{y} - \frac{1}{2}log|K_y| - \frac{n}{2}log(2\pi) \tag{3.32}$$

Equation 3.32 can be written as function of the vector of hyperparameters by making explicit K_y, the covariance matrix of the target. $\frac{1}{2}log|K_y|$ is a complexity penalty term while $\frac{n}{2}log(2\pi)$ is a normalization constant. In order to obtain the best set of hyperparameters, the maximum of the partial derivative of the log marginal likelihood with respect to $\boldsymbol{\theta}$ is computed.

As an example, consider a function $f(t) = t \cdot sin(t)$ from which a synthetic data set is created by adding a noise term that is Gaussian distributed. Now, let us try to fit a RBF kernel with three different length scales and compute the log marginal likelihood. The results are reported in the table below:

	$\lambda^2 = 0.1$	$\lambda^2 = 1$	$\lambda^2 = 25$
LML	-104.08	-93.90	-287.68

Table 3.1: Log marginal likelihood of RBF kernel with three different length scales

From table 3.1 we can see that small and big length scales are penalized by the log marginal likelihood. By looking at figure 3.10, it is clear that a length scale that is too small tends to over fit the data. On the contrary, a big length scale is not able to properly go with the model. Since the log marginal likelihood penalizes the complex models, the GPR is less prone to over fitting.

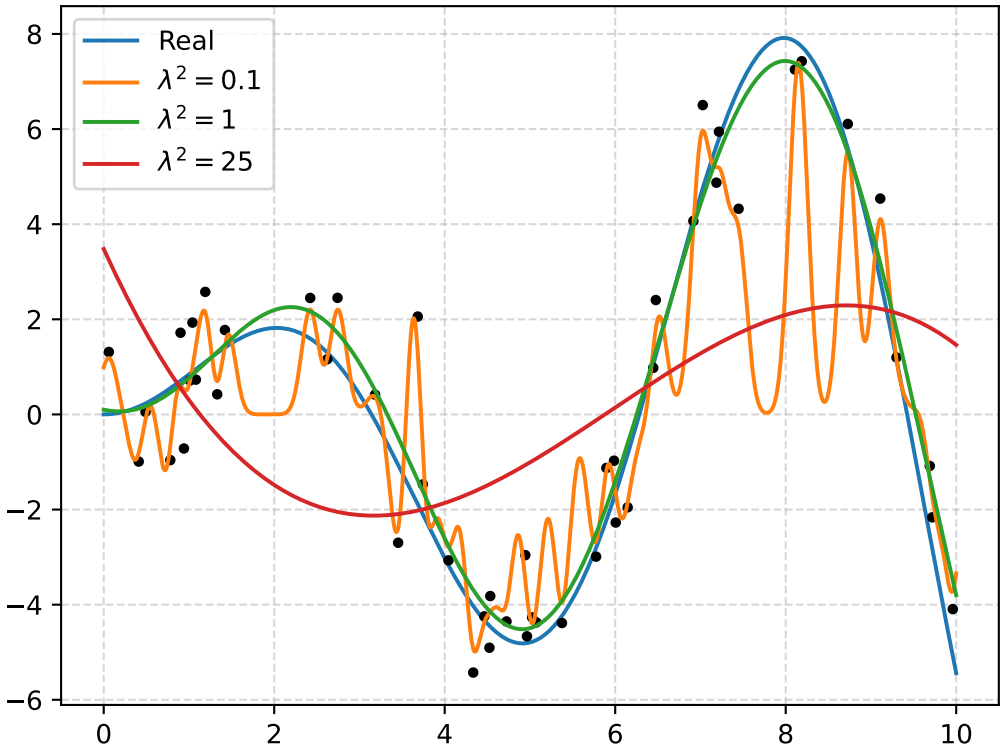


Figure 3.10: Fitting a RBF kernel with different length scale on synthetic data set. The LML penalises the the model that overfits the data.

From a computational point of view, the bottleneck in the calculation of the derivative is given by the inversion of the covariance matrix in the term K_y^{-1}. Actually during the computation, the matrix is not inverted due to the excessive computational cost. Whilst, the Cholesky decomposition is applied. Nevertheless, the cost of this operation is still $\mathcal{O}(n^3)$, making it the computational cost of a GPR model.

3.4. Model selection: search strategies

In the previous chapters, it was seen that in order to predict the evolution of a time series, it is necessary to learn its structure from previous observations during the structure modelling step. In this thesis work, we are interested in finding the linear model and GP model that best capture the structure of the time series.

In the field of search strategy, it is possible to distinguish between different approaches. Among these, there are the greedy search strategy and the exhaustive search strategy.

Once the domain of possible models is set, one could apply an exhaustive search, with which all eh possible options are explored. The downside of this approach is the computational power required to do such operation, on the other hand it is more likely to find the best result. Conversely if the greedy search is applied, a path composed of locally optimum solutions is followed, thus making the search less effective but computational affordable. An illustration of the greedy search is reported in Figure 5.8.

For what concerns the linear model, the problem is to find the degree of the polynomial able to best fit the data. In order to complete such task, an exhaustive search strategy can be used to explore all the possible degrees of the polynomial up to a desired order. In this case, the exhaustive search is applicable since the computational power required to compute polynomial regression is very low.

For what concerns the GPR, it is useful to build an automatic method for the selection of the best kernel combination. The search is done among all the possible combinations of base covariance functions. By applying this method, the domain of possible solutions assumes the shape of a decision tree, as reported in Figure 3.11. In the case in figure, only three kernels are used to build the tree.

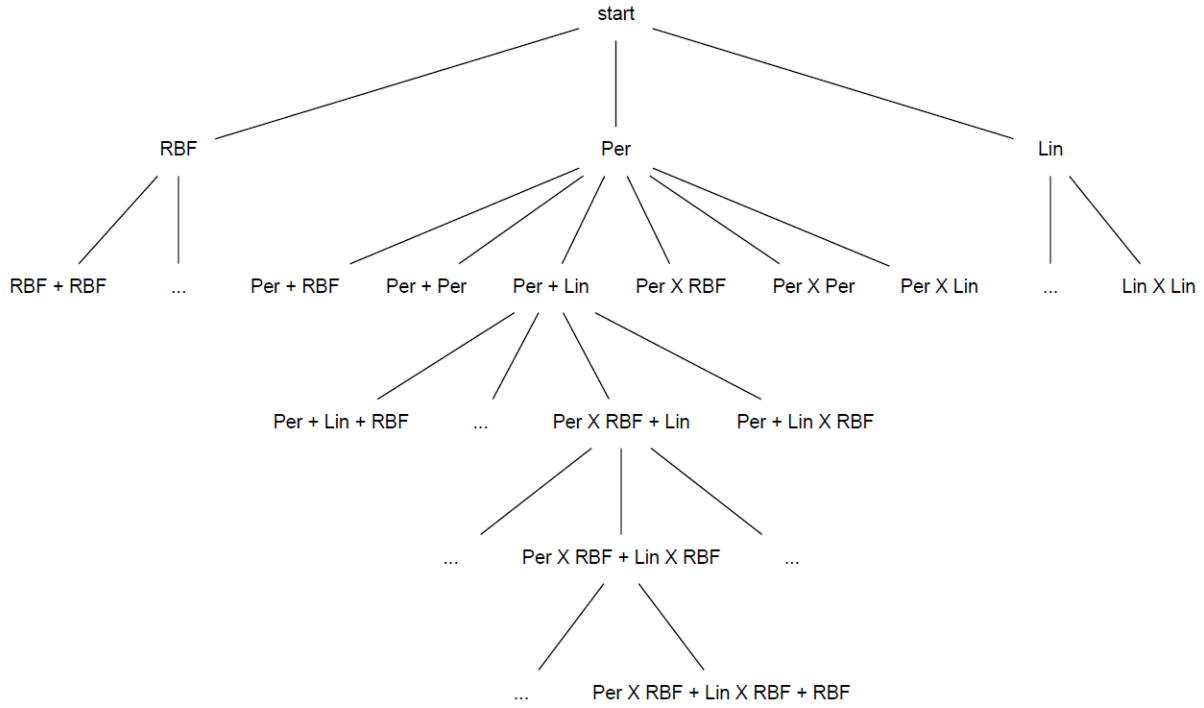


Figure 3.11: Search scheme for optimal kernel combination. The search domain assumes the shape of a tree. In the case in figure, the RBF, PER and LIN kernels are used to build the tree. The complexity of the tree and therefore scales up exponentially with the depth and the number of base kernels. The depth of the tree in figure is 5. Figure reference [19]

We can see that the complexity of the tree scales quickly with his depth, that is the level of the ramifications. Since the computational power required for GPR is $\mathcal{O}(n^3)$, an exhaustive search is impractical. In general, there will be $n \times (2n)^{d-1}$ number of nodes, where n is the number of base kernel and d is the maximum level or depth of the tree. By this calculation, a three-level, four-level, and five-level tree will have 108, 648, and 3888 number of nodes. We can see that the number of nodes grows exponentially with the number of search level. The level or depth of the tree depends on the complexity of the time series.

3.5. Cross validation

The cross validation (CV) technique is used to avoid the problem of the over fitting in the training set. In CV, the training data is split into several folds and for each fold a training and test set is made. The model is trained on the training set of each fold and than the predictive accuracy is evaluated on the test set of the fold. The most known CV technique is the k-fold cross validation. Nevertheless, CV should be treated differently when dealing with time series due to the dependencies between the data. Indeed in this case, data should be sampled orderly in time. The chose scheme was a time series split as reported in figure 3.12.

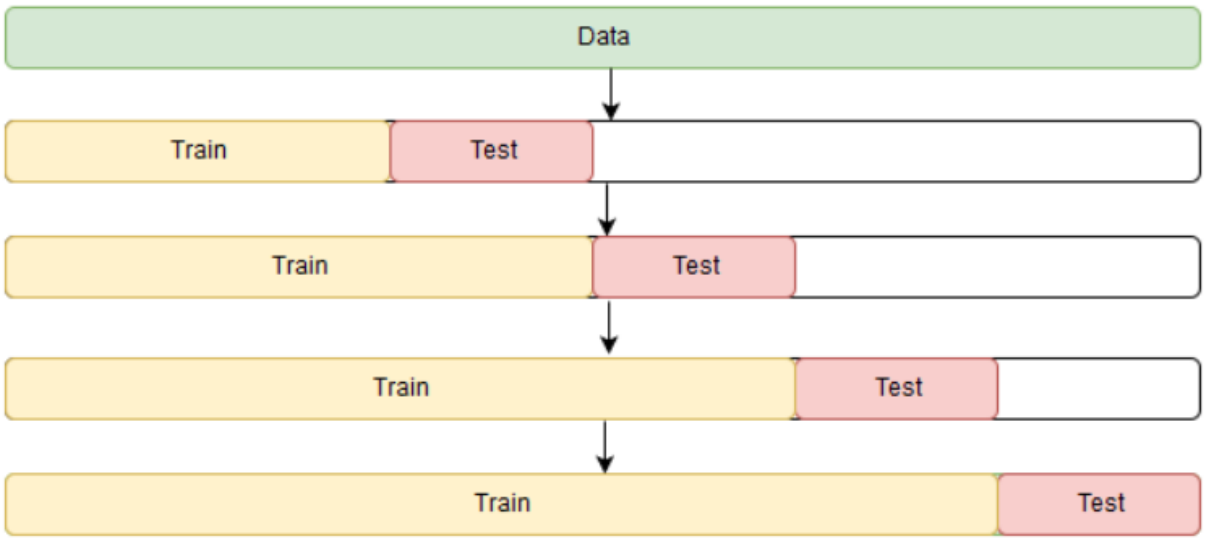


Figure 3.12: Time series split cross validation. In this case, the entire dataset reported in green, is split in 4 folds. The size of the train set changes at every fold. It is also possible to leave a gap between the train and test set.

In the example in figure, the data set is divided into 4 folds, the training set differs

from the previous one due to the addition of new data points while the size of the test set remains unchanged while shifting towards the horizon. The model is trained on the training set and is tested on the test set for four times. In the end, the total error of the model is computed as the mean of the error of the k folds. In the case of figure 3.12, the total error of the model is $\frac{1}{4}\sum_{k=1}^{4} e_k$ where e_k is the error of the model trained on the $k-th$ fold. It is also possible to leave h steps between the train and test set in order to train the model in forecasting $h-steps$ ahead. On the other hand, the downside of the CV is that it requires more computational power for the training of the model since it has to be trained k times. Also, a portion of the data can not be used for training the model because of the fold structure.

3.6. Evaluation metric

In order to evaluate the accuracy of a model with respect to another, an evaluation metric is needed since it explains the performance of the model. In this work the Mean Absolute Error (MAE) is used as evaluation metric. MAE is the average of differences between predictions $\bar{y_i}$ and actual observations y_i:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \bar{y_i}| \tag{3.33}$$

The metric ranges between 0 and $+\infty$ and is indifferent to the direction of the errors. It is a negatively oriented score, meaning that lower values are indicative of a better model. MAE does not penalize large errors, thus making it more desirable when working with noisy dataset.

In the CV technique, the final MAE is given by averaging the MAE calculated on every fold:

$$MAE_{CV} = \frac{1}{k}\sum_{i=1}^{k} MAE_i \tag{3.34}$$

3.7. Dummy variable

The aim of a dummy variable is to account for a particular event that happened in a point in time. Such variable assumes a value that is either 1 or 0. Given a time series with regressor $t = (t_1, ..., T, ..., t_n)$, let us consider an event happening at time $t = T$. Than a

dummy variable can be defined in two ways as

$$d = \begin{cases} 1 & t \geq T \\ 0 & t < T \end{cases} \tag{3.35}$$

$$d = \begin{cases} 1 & t = T \\ 0 & t \neq T \end{cases} \tag{3.36}$$

The definition of the dummy variable shown in Equation 3.35 is used when the event still affects the process after it happened. For example, when the event causes a significant mean shift in the time series for $t \geq T$. On the other hand, the definition given by Equation 3.36 is used for a point-wise evaluation of the event. Dummy variables should be included in the estimation of predictive models as regressors, otherwise the evaluation metrics could return false estimates about the model.

3.8. Ensemble forecast

An ensemble forecast is a machine learning technique able to return models with higher robustness. In the ensemble forecast, the train set is sampled in different ways and different models are trained on different samples. The final model will be a mean of the different models obtained.

4 | Tools

In order to conduct this thesis work, a set of computational tools was used. Python programming language was used to implement ML algorithm and carry out the experiments, while Microsoft Excel was used as an interface between the DCS and Python. The free, open source environment used to write the algorithms in Python is Spyder. The experiments were conducted on a laptop ThinkPad P53s with 24 GB of RAM and processor Intel CORE i7-8565U CPU @ 1.80 GHz with 4 cores .

4.1. Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed [21].

These characteristics made the Python programming language one of the most used worldwide. This success allowed the development of a numerous free, open source libraries for predictive analysis, ML and much more. A library is a collection of classes and functions that users import into Python programs. In this work, the libraries form Scikit Learn where used for ML an predictive data analysis.

Other libraries used and that are worth of mention are Pandas to format and organize the data coming from excel, NumPy that provides Python with a contiguous numeric array datatype and Matplotlib for creating static, animated, and interactive visualizations.

4.2. Scikit Learn

Scikit Learn (SKL) provides an open source machine learning library for the Python programming language. The ambition of the project is to provide efficient and well-

established machine learning tools within a programming environment that is accessible to non-machine learning experts and reusable in various scientific areas. The library has been designed to tie in with the set of numeric and scientific packages centered around the NumPy and SciPy libraries [15].

`gaussian_process`, `model_selection`, `preprocessing` and `linear_model` are the most important modules called from the SKL library. Once the module is imported, it is possible to call the functions belonging to the model. The process is done by executing the following line of code: `from library.module import function`. After the functions execute, the methods of the functions can be called with the line of code `function.method` for further actions. Hereafter, it is reported an overview on the modules, functions and methods used and their application.

4.2.1. gaussian_process

Is the module responsible for dealing with GP methods in order to solve regression and classification problems. The functions called from the modules are:

`kernel`. To call the base kernels from the list of kernels available in the SKL library. The base kernels can be used as they or to build more complex kernels by addition and multiplication.

`GaussianProcessRegressors`. To perform the GP regression. The function requires different parameters for the computation.

- `kernel`: the kernel specified as prior to be used in the regression.
- `n_restarts_optimizer`: since the maximization of the log marginal likelihood has different solution, the optimizer can be started several times.
- `alpha`: this value is added to the diagonal of the kernel matrix during fitting. This can prevent a potential numerical issue during fitting by ensuring that the calculated values form a positive definite matrix. Since it can be interpreted as the variance of additional Gaussian measurement noise on the training observations, it is computed as $alpha = \sigma^2_{noise}$.

After the algorithm solves the optimization problem, the `fit` method is called to fit the regression model. Also, the `predict` method is used to return the prediction on train and future data and the prediction interval. These methods requires the vector of training data and the vector of regressors as input.

4.2.2. model_selection

Is the module used to deal with CV techniques. The function imported from this method was `TimeSeriesSplit` which creates k-folds as explained in section 3.5. This function requires the following parameters:

- `n_splits`: the number of folds in which the time series is split.
- `gap`: the number of data points that divide the train and test set.

4.2.3. preprocessing and linear_model

The `preprocessing` module is used for scaling, centering, normalization and binarization methods while the `linear_model` module implements a variety of linear models.

The function imported from the `preprocessing` module is the `PolynomialFeatures` and it is used to build the polynomial combinations. This function generates a new feature matrix consisting of all polynomial combinations of the features with degree less than or equal to the degree specified with the parameter `degree`.

`LinearRegression` is the function imported from the module `linear_model`. No parameters are needed for initialization, instead the methods `fit` and `predict` are called for fitting and prediction as in the `GaussianProcessRegression`.

5 | Data

This chapter reports an overview on the data used for the implementation of the ML algorithms. The data were made available by the Itelyum regeneration plant in Pieve Fissiraga (LO) that has an historian of several years collected thanks to the information management system installed. The data are accessible through an Excel add-in and from there are imported to Python for visualization and elaboration.

5.1. Big Data use case: Itelyum Regeneration

Itelyum Regeneration Spa is a European leader in the regeneration of exhausted lubricant oils and a great example of circular economy. Exhausted lubricant oil is a dangerous waste, therefore its collection must happen trough a qualified and authorized company. On the basis of accurate physical-chemical analysis, the exhausted oil is sent to the re-refinery plant to undergo the regeneration process. The regenerated lubricant oil is sent to the market to close the life cycle.

The re-refining process consists in three steps [9]:

1. *Preflash*: the used oil is heated up to 140 $°C$ and distilled in a vacuum column. This operation allows the separation of light compounds such as light hydrocarbons and water from the oil. The light hydrocarbons are than used for energy supply to the plant.
2. *Thermal de-Asphalting*: the dehydrated oil is heated up and distilled at 360 $°C$ in a vacuum column. the three cuts are sent to storage ready for the catalytic process.
3. *Hydrofinishing*: the oil and the hydrogen are heated up to 300 $°C$ and sent to a catalytic reactor. The catalyst favours the reaction of hydrogen with unsaturated compounds, sulphur and nitrogen.

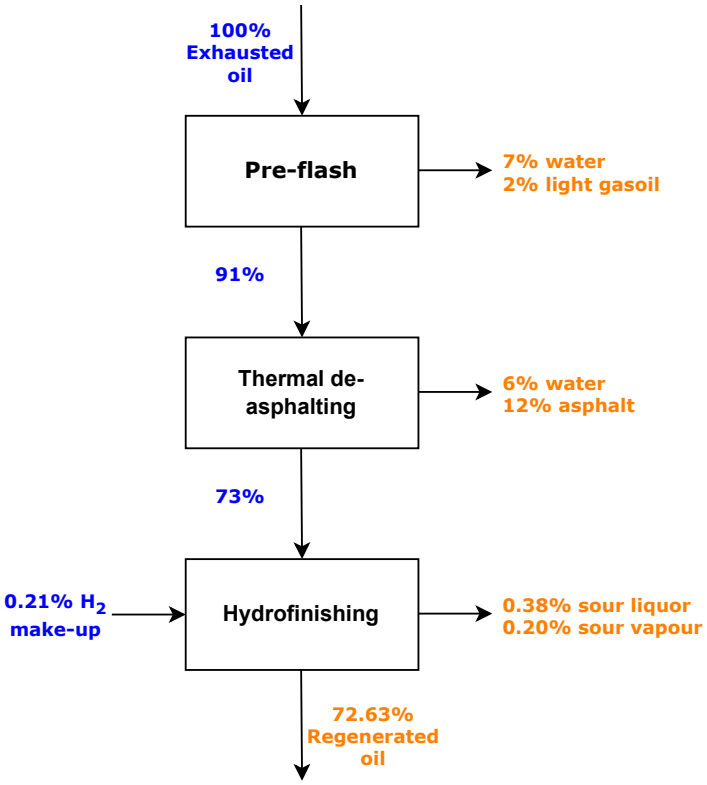


Figure 5.1: Block flow diagram of the Itelyum Regeneration process. 72.63% of the exhausted oil is recovered through a regeneration process consisting of three steps: pre-flash. thermal de.asphalting and hydrofinishing.

Starting from 100 Kg of waste oil, the process allows to recover:

- 65 Kg of new oil.
- 22 Kg of bitumen and gasoil.
- 8 Kg of purified water.
- 5 Kg of waste to be treated by thirds.

The plant processes about 200.000 Kg of exhausted oil in one year, therefore reinserting about 130.000 Kg of regenerated product in the lubricant life cycle.

5.1.1. Exaquantum

The Itelyum regeneration plant located in Pieve Fissiraga (LO), is currently embracing the digital revolution. Since 2018, the plant has been equipped with the Exaquantum system by Yokogawa. Exaquantum is one of the most comprehensive Plant Information

Management Systems (PIMS) available for process industries. Exaquantum can acquire data from all facets of a process and transform that data into easily usable, high-value, widely distributed information [1].

This technology employs a client/server architecture based on Microsoft Windows operating system. The Exaquantum PIMS acts as a server by integrating data coming from other sources and from the plant management system such as Distributed Control System (DCS), Programmable Logic Controllers (PLCs), Process Control System (PCS), etc. The client can than access the server by a local computer located at the plant or via web and employ the interface by Exaquantum for real time data visualization, administration tools, etc. Furthermore, the data are stored in a long term archive so that it is possible to access historical trends and events. In this way, data becomes an integral part of the set of tools used by the plant engineers in vital decision-making processes.

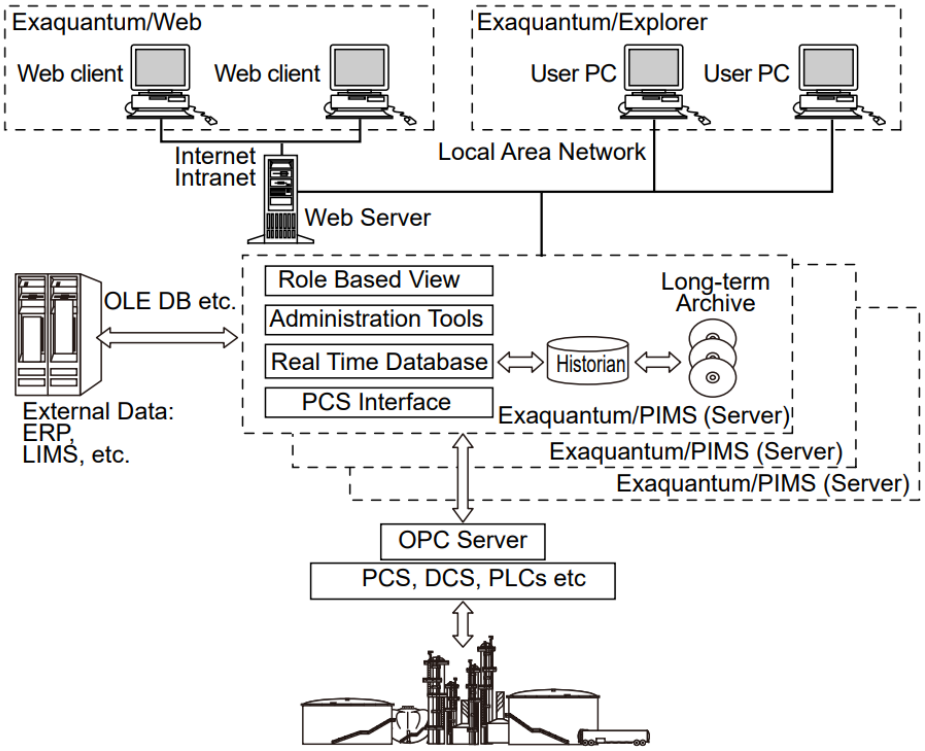


Figure 5.2: Overview of Exaquantum PIMS by Yokogawa. Data are collected from the control systems of the plant and gathered in servers that are accessible by the client through a local or web server.

In addition to the data coming from the Exaquantum PIMS, further information on special routines at the plant was taken directly from an Excel file that is filled daily by the operators of the plant in order to keep track of the process. In this file, the daily

average of some critical variables is computed and stored along with information about the day in which maintenance is carried out. Nevertheless, this practice implies a huge loss of information about the data. The excel file was mainly used to identify the days in which special interventions were done.

5.2. Dataset

A furnace is a key unit in a chemical plant, since a proper heating allows for a correct execution of the process in order to obtain the desired specifications of the output product. Furthermore, it is a highly energy-intensive unit, contributing in large part to the operational expenditures (OPEX) of the plant. In particular, a furnace that processes mineral oils is prone to fouling phenomena due to the formation of carbon coke that deposits on the tube's walls. This causes the restriction of the tube section and sudden spikes in pressure drops. Therefore from time to time, an intervention of maintenance is needed in order to unclog the tubes. This is done by a process called blowing, during which a current of high pressure steam is fed to the tubes in order to wash the coke away. Also, the skin temperature of the tubes should be monitored in order to ensure integrity and a correct heat exchange with the oil.

For the reasons stated above and in accordance with the manager of the plant Eng. Francesco Gallo, the study was focused on the furnace *PH-401B* in the thermal deasphalting section of the plant. The furnace can be divided into three sections: lower, upper and middle. Methane is fed to the lower section, making it the hottest part of the furnace. For what concerns the tube skin temperatures, the measure of one tube per section was taken, so that every measure is indicative for all the tubes belonging to that section. When furnace *PH-401B* is stopped for maintenance, the oil feed is sent to the by-pass furnace *PI-401A* in order to carry out the maintenance without stopping the entire plant.

The list of variables analyzed and their description is reported in Table 5.1, while a qualitative scheme of the furnace is reported in Figure 5.3. When importing the data to Excel from the Exaquantum add-in, other than the variable it is possible to specify the period of time and the time step Δt between the data. The plots in this section were obtained in the period ranging from 11/06/2021 to 11/12/2021 with $\Delta h = 8h$ or $\Delta h = 10min$. Dummy variables were obtained form an excel sheet provided by teh operators of the plant.

Label	Description	SI unit
FI-4091	Methane feed to furnace	Sm^3/h
PI-4301	Oil pressure at furnace inlet	bar
PI-4304	Oil pressure at furnace outlet	mmH_2O
SK-4072	Tube skin temperature in furnace lower section	$°C$
SK-4070	Tube skin temperature in furnace upper section	$°C$
SK-4067	Tube skin temperature in furnace middle section	$°C$
d_b	Blowing	–
d_{pm}	General plant maintenance	–
d_{BA}	Furnace switch	–

Table 5.1: Description of analyzed variables: methane feed to furnace, pressure at inlet and outlet of the tubes, tube skin temperatures for lower, middle and upper section and dummy variables describing special interventions.

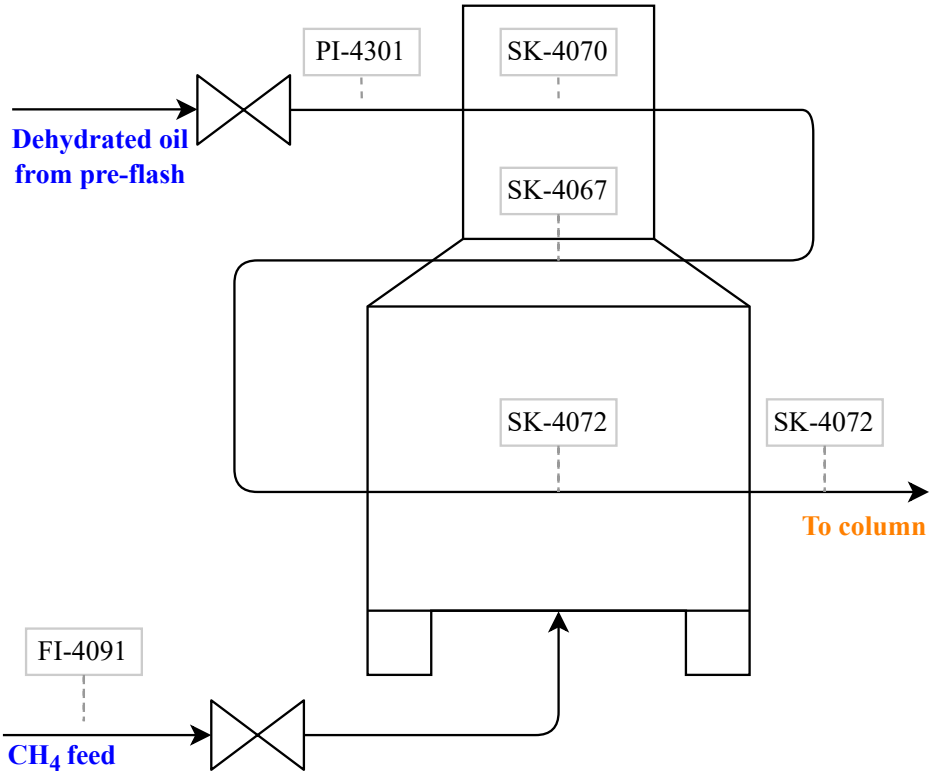


Figure 5.3: Qualitative scheme of furnace *PH-401B*: the oil feed goes through different sections of the furnace to ensure a proper heating. The methane feed is placed at the bottom, making the lower part the hottest.

In order to take into account the total pressure drop inside the tubes, the pressure difference $\Delta P = |PI4301\text{-}PI4304|$ is considered. Nevertheless, being $PI\text{-}4304 \simeq 0$ bar, the total pressure drop is indicated as *PI-4301*.

In Figure 5.4, the time series plots of the dataset are reported for a period of six months. The time interval ranges from 11/06/2021 to 11/12/2021. Starting from the first plot, we have the time series data of variable *FI-4091*, that is the CH_4 feed to the furnace. The second plot represents the pressure drop *PI-4301* inside the tubes that process the mineral oil. In the third plot, it is reported the time series of the tube skin temperatures *SK-4067*, *SK-4070*, *SK-4072* for the three sections of the furnace. The tube skin temperature of the lower section is represented by the green curve, while the orange curve describes the upper section. The last plot shows the dummy variables for general maintenance of the plant: blowing, switch from furnace *PH-401B* to *PH-401A* and general plant maintenance indicated as d_b, d_{BA} and d_{pm} respectively.

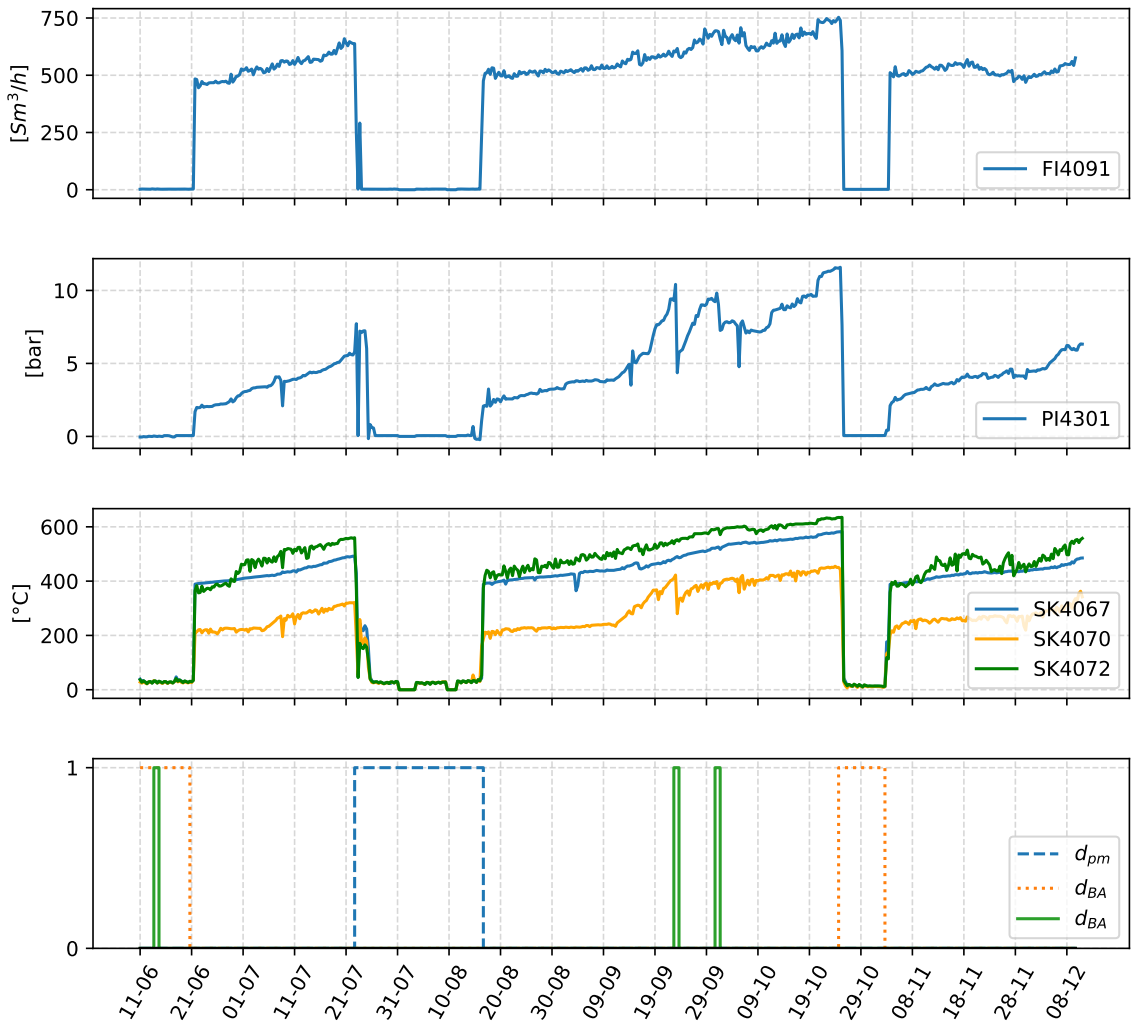


Figure 5.4: Overview of variables coming from furnace *PH-401B*. From above, time series of methane feed, pressure drop, tube skin temperatures in lower, middle and upper section and dummy variables indicating blowing, general plant maintenance and switch from furnace *PH-401B* to furnace *PH-401A*. The time series share the same x-axis. Three different cycles, interrupted by maintenance operations, are clearly visible. $\Delta t = 8h$

First of all, it is clear that there is an irregular seasonality in the structure of the time series. The furnace goes through different life cycles whose periods are defined by the maintenance operations. Three of these cycles are shown in the plots. Also, the time series are non-stationary since they exhibit a mean shift. This is due to the fouling process. Indeed, given that the coke deposits inside the tubes with time, it reduces the heat exchange with the charge, so that a higher skin temperature is gradually needed to

ensure a proper heating. The target temperature is reached by feeding more methane to the furnace and this explains the mean shift of *FI-4091*. Also, the effects of the coke deposition on pressure drops are clearly visible by looking at *PI-4301*, since it causes spikes in pressure due to the reduction of the tube section. Additionally, the effects of maintenance intervention are recognizable. The furnace was stopped from 23/07 to 18/08 for a general maintenance of the plant during which the production was interrupted. Whereas, in the periods 11/06 - 21/06 and 24/10 - 3/11 the furnace *PH-401B* was stopped for maintenance without shutting the plant down thanks to the presence of the back-up furnace *PH-401A*.

By looking at Figure 5.5 and Figure 5.6 it is possible to have a better insight on the effects of the blowing on the variables. In the figures, the blowing of the day 22/09 and 01/10 is directly highlighted in red over the variable of interest. The upper plot in Figure 5.5 clearly shows the beneficial effects of the blowing on the pressure drop. The threshold value over which the blowing is needed is around 12 bar. Conversely, by looking at the upper plot, it can be seen that the methane feed is drastically reduced during the blowing phase. The mean of variable *FI-4091* is not affected by blowing as it is for *PI-4301*.

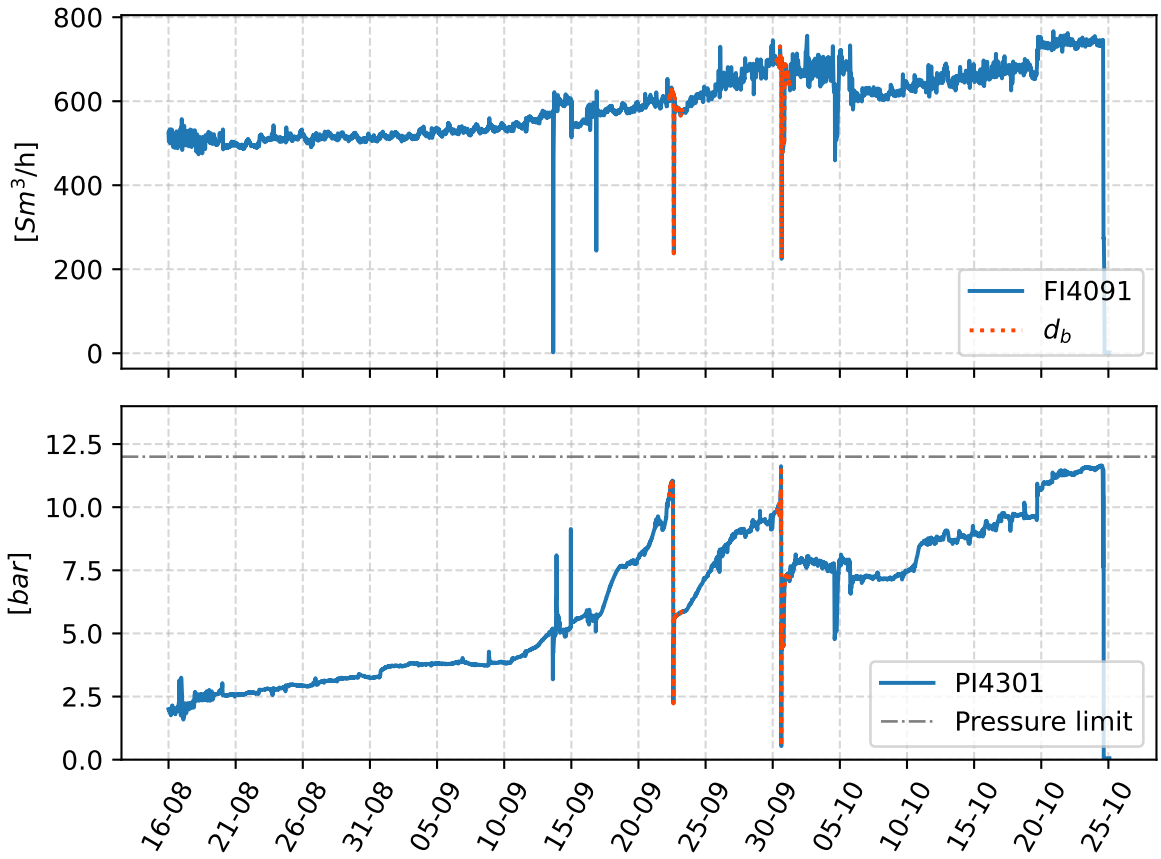


Figure 5.5: Starting from above, effects of blowing on methane feed and pressure drop in furnace *PH-401B*. A significant mean shift after the maintenance only happens for the pressure drop since the tubes are unclogged. The grey dashed line indicates the pressure around which the blowing is conducted that is 10 bar. The time series share the same x-axis. $\Delta t = 10min$

Figure 5.6 shows that the blowing procedure also affects the skin temperature of the tubes. Starting form the top, it is possible to reason that the mean of the time series representing the lower and middle sections of the furnace is not affected by blowing. On the other hand, the tube skin temperature of the upper section *SK-4070* shows a significant mean shift only after the intervention of 22/09.

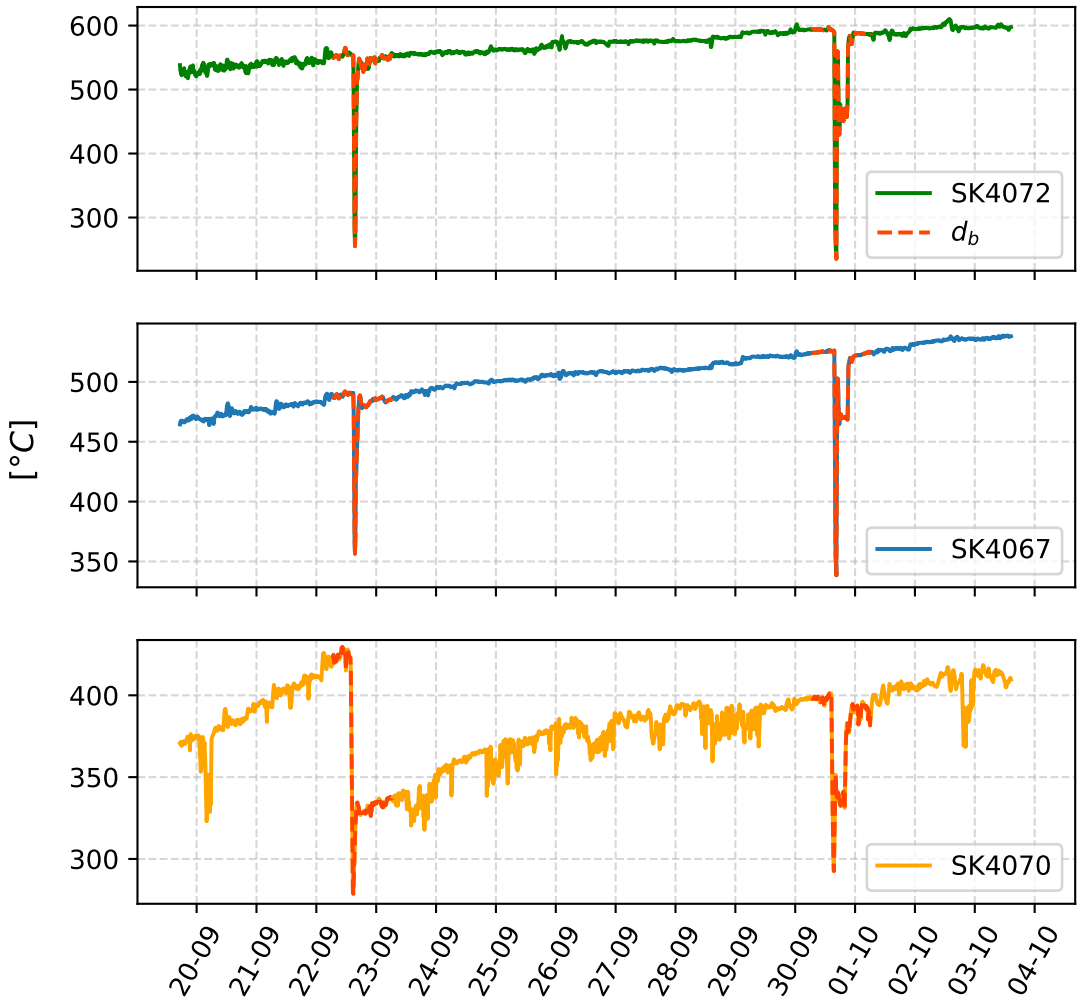


Figure 5.6: Starting from above, effects of blowing on tube skin temperature in the lower, middle and upper section of furnace *PH-401B*. The red part highlights the days in which blowing is carried out. A significant mean shift is visible only in the upper section after the intervention in day 22/09. The time series share the same x-axis. $\Delta t = 10min$

Nevertheless, the dummy variables indicated in the excel file provided by the operators of the plant do not explain all the abnormal fluctuations of the data. For example, this can be seen in Figure 5.7 where a close-up on the methane feed in the period ranging from 16/08 to 25/10 is shown with a $\Delta t = 100min$.

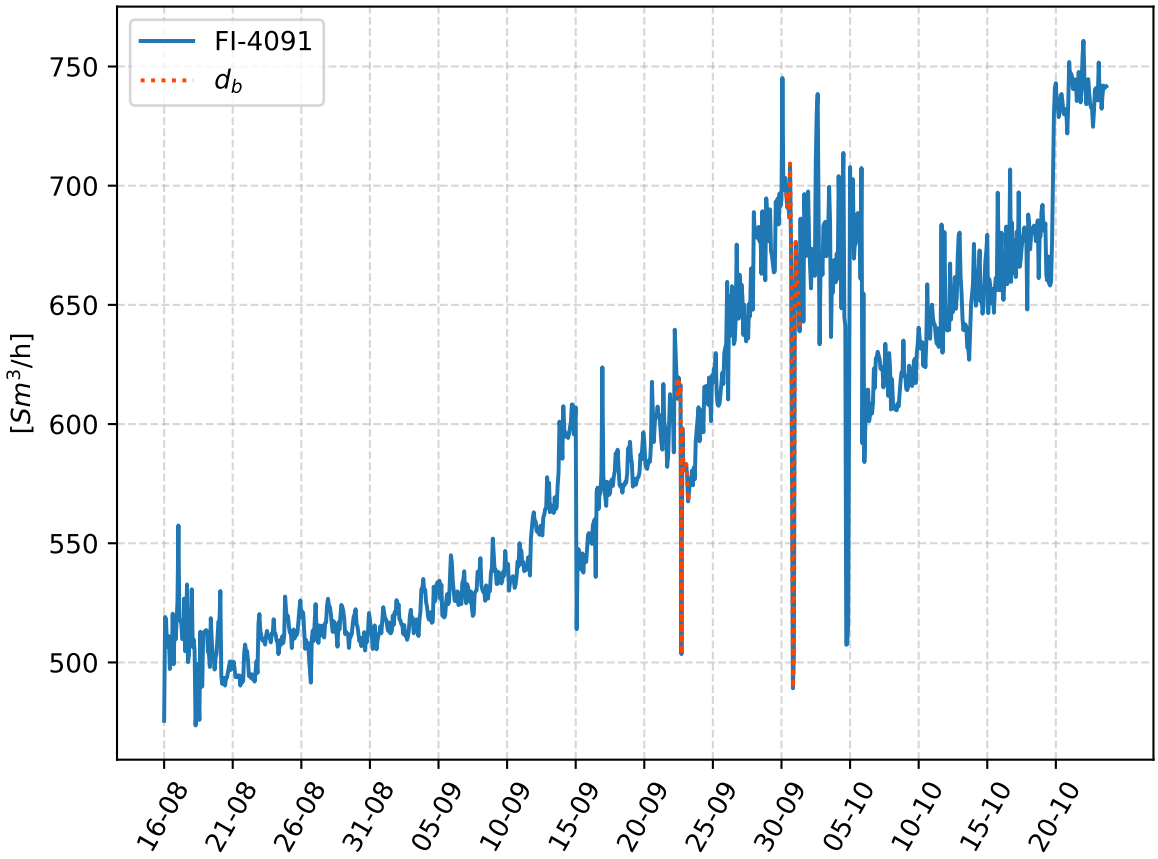


Figure 5.7: The time series presents many irregularities. Whilst the effects of blowing are highlighted in red, no assignable cause is available for other irregularities such as the jump at 15/09 and the mean shift after 5/10.

The one highlighted in red are the blowing operations as explained previously. Nevertheless, a jump can be seen in the day 15/09 and after the 05/10. Also, a significant mean shift is detectable from day 20/10 on. Since there are no assignable causes for these irregularities at least by integrating the history of events from the sources available, it is not possible to assign a dummy variable.

It is clear that the structure of the time series presents irregular patterns characterized by trends, sudden mean shifts, irregular seasonality and outliers. On top of that, there are significant fluctuations and noise. Due to these anomalies, the development of a predictive model is a tedious work especially for a long term forecasting. According to Hyndman and Athanasopoulos, the key to a successful long term forecast is to model and extrapolate the underlying pattern that exists in the past data and ignore the random fluctuations that occur [10].

5.3. Problem statement

The aim of this work is the development of an algorithm for a data driven approach to predictive maintenance of the furnace *PH-401B* by monitoring and predicting the trend of the methane feed, tube pressure and tube skin temperature.

The monitoring system consists in data that continuously arrive from the Exaquantum PIMS, are stored in a CSV file and are fed to the algorithm. The algorithm elaborates the data, find the best model able to describe the time series and returns the long-term prediction. Therefore, every time there is availability of new data, for example after one day of plant operations, the model is refitted in order to give new predictions.

The system is rebooted every time the furnace is shut down for maintenance. Therefore, at time $t = 0$ the furnace is turned on after maintenance, while at time $t = T$ the furnace is turned off for maintenance. Therefore, the period of one life cycle is given by the array $\mathbf{t} = (t_0, ..., T)$. The choice to reboot the system at every cycle is given by the high irregularities present in the data that make it too difficult for the model to learn the structure of the time series.

5.4. Experiment setup

In the experiment, GPR and Polynomial Regression (PR) are compared along with three different learning approaches to see which model is better at extrapolating the patterns in the time series for long term forecasting.

Three different learning approaches were compared on the dataset of the second cycle of the methane feed to furnace *FI-4091*, reported in Figure 5.5 by simulating the real application of the algorithm. After choosing a refitting period Δt_r, the models were trained with a gradually increasing number of data points in order to return the prediction on a prediction range h. Afterwards, the score of the model obtained with one of the three learning approaches was computed on the prediction. The scores were than used to compare the performances of the three learning approaches to choose the most suitable.

The learning approaches are reported in Table 5.2. CV implies a 60-20-20 train-validation-test split, while the training without CV implies a 80-20 train-test split. Finally, the ensemble forecast approach returns the mean of two models, one trained on even days and one trained on odd days.

Model name	Description
PRCV	Polynomial regression with CV
PRnCV	Polynomial regression without CV
PRE	Polynomial regression with ensemble method and no CV
GPRCV	GPR with CV
GPRnCV	GPR without CV
GPRE	GPR with ensemble method and no CV

Table 5.2: Description of the three different learning approaches applied to PR and GPR.

During the experiments, the refitting period is set to $\Delta t_r = 1day$, while the time step between the data points is set to $\Delta t = 8h$, making three new data points per day. The value of Δt was arbitrary chosen by seeing that the performance of the model would decrease with too many data points. This is due to the extreme noise in the data. The score used for comparison is the MAE.

In order to search for the best degree of the polynomial and the best combination of base kernels to fit the data, two different search strategies were used for PR and GPR. Also, the dummy variables were implemented as regressors in the PR only, since the GP is able to automatically model the irregularities caused by the maintenance intervention.

5.4.1. Polynomial model

Since the time series follow an increasing linear or quadratic trend, it was chosen to start with the linear regression model. The learning phase focuses on finding the adequate degree of the polynomial and than compute its vector of parameters θ by minimizing the sum of squares as explained in Section 3.2. In this phase, an exhaustive search strategy is adopted: the highest allowable degree of the polynomial is set to $d_{max} = 5$ and all the possible models of order up to d_{max} are evaluated. After the computation, every model is tested on the test set and its MAE is computed. This operation must be done k-times in the case of a CV with k folds. According to the value of the metric, the best model is selected and it is trained on the totality of the train and test set. The prediction interval of the model is than evaluated with Equation **??** and than the forecast is computed along with the prediction interval. Finally the prediction is computed along with the confidence interval as seen in Section 3.2.

5.4.2. GP model

GP model was chosen for its versatility and robustness against the outliers and its ability to model wiggles and sudden mean shifts in the data. The learning phase focuses on finding the best combination of base kernels from the decision described in Section 3.4 tree and the optimal hyperparameters of the covariance function by minimizing the LML as described in Section 3.3.

In order to search for the optimal combination of base kernels, it was developed a hybrid approach between the method used by Rasmussen and Williams [17] and the one used by Duvenaud et al. [19], described in Section 2.2. The kernels that could best model the structure of the time series are chosen among the base kernels by looking at the data and are than used to build the search tree in Figure 3.11. In this way, less combinations are explored since a previous knowledge about the time series allows to discard from the start the kernels that for sure are not able to model the structure. By doing so, the model will not be fully automated but the options to search for will be less. In order to search for the optimal combination of kernels, a greedy search is applied. The depth of the tree is set to $depth = 5$ in order not to have too complex kernels. Furthermore, the addition of two linear kernels is avoided since it results in another linear kernel.

By looking at the structure of the time series, the chosen kernel for the structure modelling are RBF, LIN and RQ. Indeed, the first is able to model the trend of the time series by addition and multiplication with itself or with the RBF and RQ. On the other hand, the RBF and RQ kernel are responsible for modelling the small fluctuations and the smooth fluctuations of the data respectively. The combinations of these three kernels define the decision tree in Figure 5.8 with the RQ kernel in place of the PER kernel.

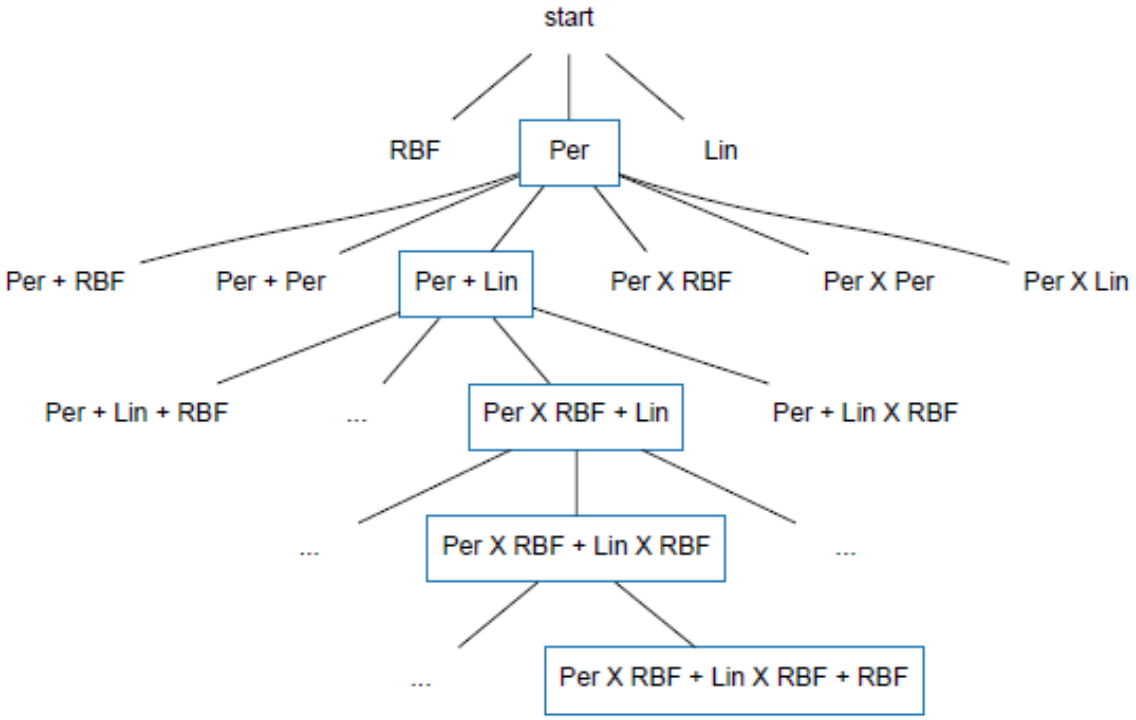


Figure 5.8: Example of greedy search strategy with the RBF, LIN and PER kernel. A path of local optima is followed, thus this approach does not assure the best result possible, but computational power and time required for the search are largely diminished. The tree implemented in this work uses the RQ kernel in place of the PER kernel. Figure reference [19].

Every model evaluated form the search tree is trained on the test set and tested on the train set to compute the error metric. As in the PR, the model with the lowest metric is chosen as the best model and is trained on the totality of the test and train set to obtain the prediction. Of course, in the case of CV, the search tree must be computed for every fold, making it computationally very expensive.

A stated in Section 4.2, the `GaussianProcessRegressors` function is the one used to perform the GPR. The value of the parameters given as input to the function are the following: `GaussianProcessRegressors(kernel=k, n_restart_optimizer=50, alpha=`σ^2_{noise}`)`. Where `k` is the kernel computed from the decision tree. The number of restart of the optimizer is set to 50 in order to have a better search of the maxima of the objective function. The value σ^2_{noise} is the variance of the noise, where the noise is computed by taking the first difference of the training set.

6 | Results and discussions

In this chapter, it is reported an overview on the results obtained during the training and testing of the models. The two approaches, PR and GPR, are compared and discussed along with the three different ML approaches implied to train the models. Based on the results, the best approach to implement the online monitoring system is chosen along with the best model.

6.1. Learning approach comparison

In the following experiments, the three learning approaches are compared both on the PR and GPR. The refitting period is set to $\Delta t_r = 1day$, that is the model is refitted with new data every day. Since the time step between the data is set to $\Delta t = 8h$, three new data points are used to refit the model every day. The long term prediction and its MAE are computed for a prediction range of $h = 10days$.

In Figure 6.1, the results for the PR are reported. The upper plot shows the total MAE of every model divided by the number of times the model has been refitted. The MAE is shown both for the train and test sets. The second plot shows, the point wise MAE on the test set for every iteration. While the last plot is for comparing the trend of the MAE with the time series of interest; that is the methane feed to the furnace as explained in Section 5.2.

By looking at the first plot, it would be possible to conclude that the best approach is the PRCV. Nevertheless, by looking at the point wise MAE, it can be seen that the PRCV performance are the lowest for the majority of the times. PRnCV and PRE mostly perform better than PRCV since the orange and green lines are under the blue line for most of the iterations. The exceptions are after major disturbances in the data such as after the 16/09 and after the 7/10. Also at start, the PRnCV and PRE seems to be uncertain.

The values of the MAE on the train set are not close to zero meaning that the model does not over-fit the data on the train set. This is a sign that the model is able to generalize

well on the test set.

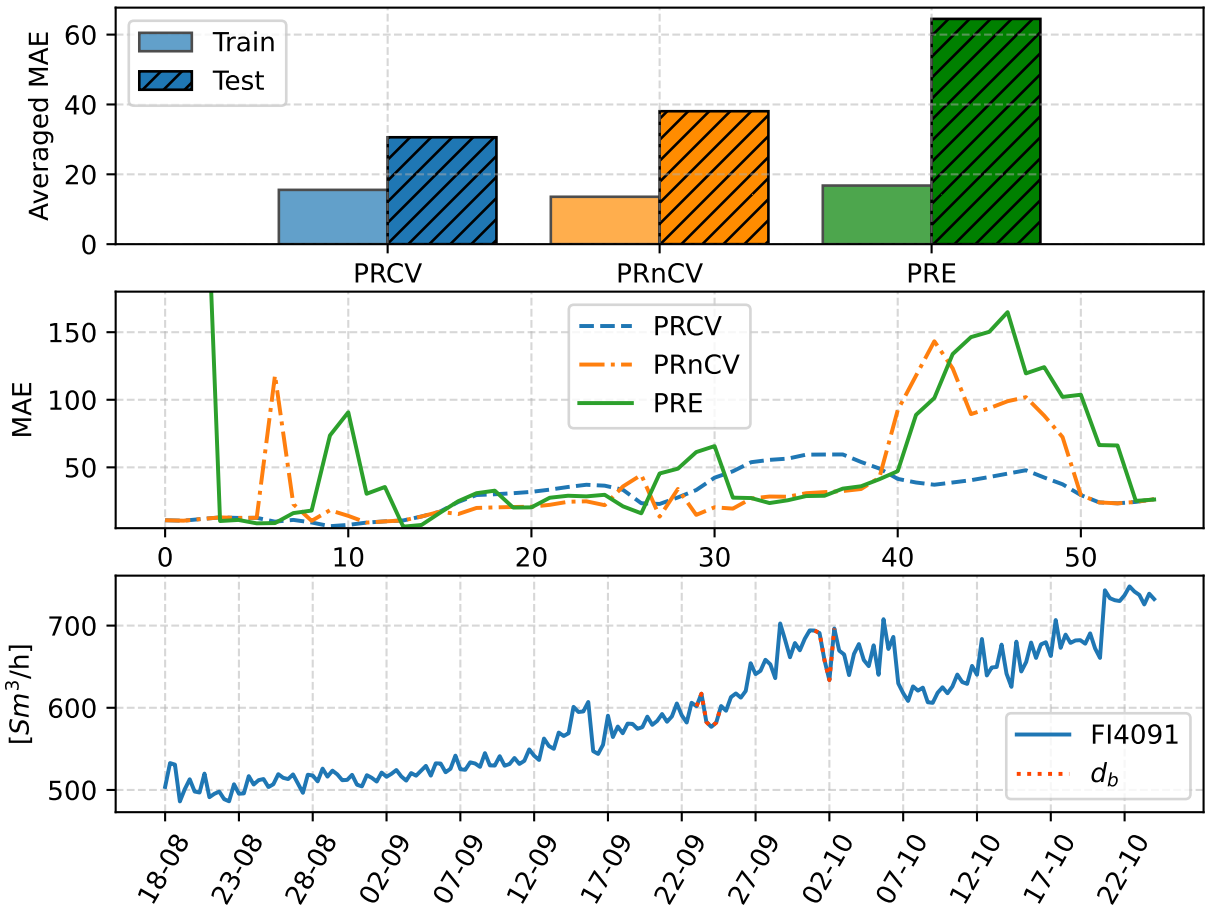


Figure 6.1: Comparison of the performances of PRCV, PRnCV and PRE. From above, are reported the average MAE on train and test set, the MAE for every day of refitting and the methane feed to funace for comparison. From the first plot, the PRCV seems the best choice. Nevertheless, by looking at the second plot, it is possible to see that PRnCV and PRE perform better the majority of times. $\Delta t_r = 1day$, $\Delta t = 8h$, $h = 10days$.

By looking at the degree of polynomials chosen by the algorithm, it was seen that the PRCV always chooses the linear model. This results in a lower total MAE, but the model can not properly go with the time series that has a clear quadratic trend. The linear model will always be under the mean of the curve when tying to predict. On the other hand, PRnCV and PRE choose different orders for the model at every iteration, the large MAEs are caused since a model of order grater than one rears up very fast when extrapolating therefore they are more influenced by disturbances. Also it is possible to see that blowing does not affect the MAE considerably.

In the case of the GPR, the results plotted in Figure 6.1 show that the GPRnCV is the

best method to choose among the tree. It has a lower average MAE and also the point-wise MAE has lower values and less fluctuations. Also in this case, the blowing does not affect the performance of the model that on the contrary seems to be influenced more by the fluctuations generated by unknown causes in day 16/09 and 07/10. Furthermore, the computational time required to train a GPR with an ensemble method or with CV is much higher than the time required to train the GPRnCV.

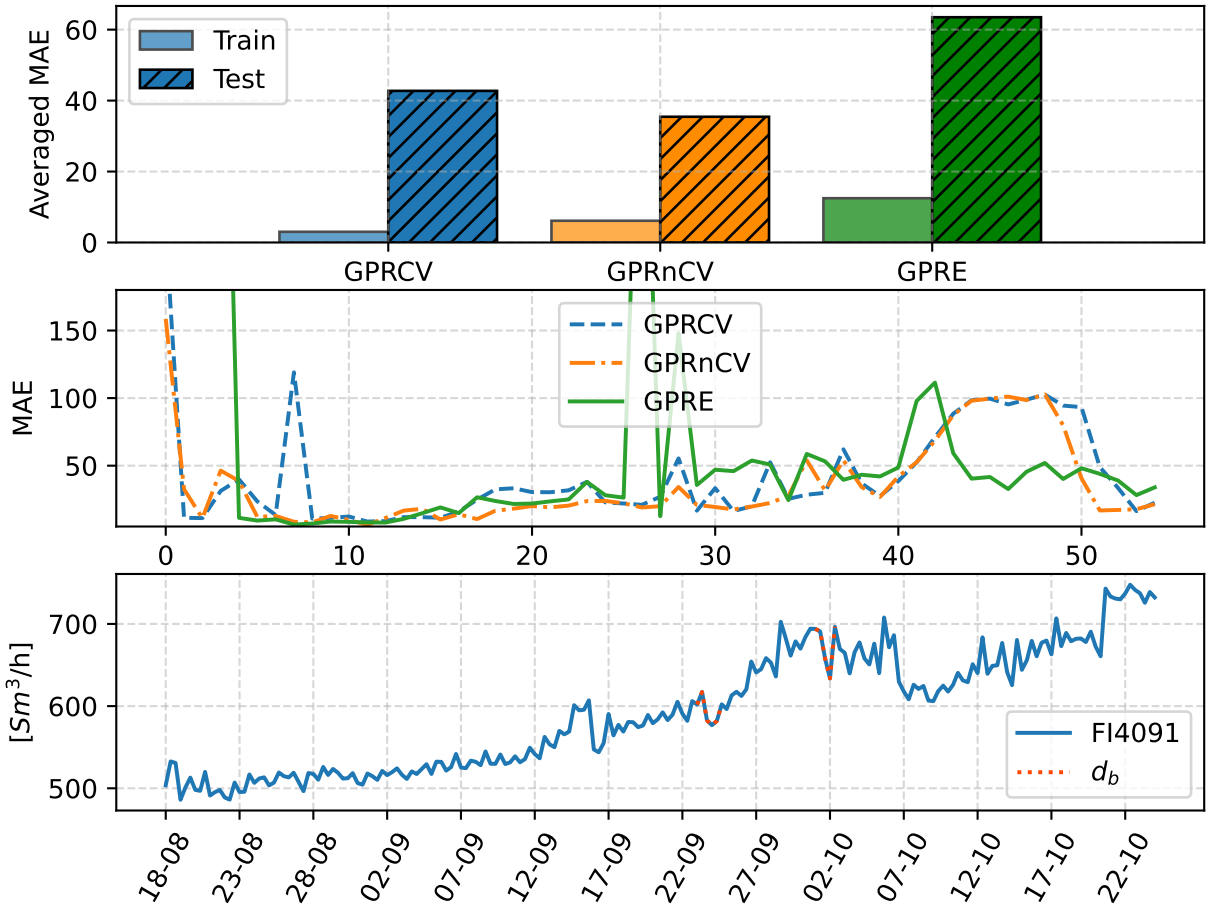


Figure 6.2: Comparison of the performances of GPRCV, GPRnCV and GPRE. From above, are reported the average MAE on train and test set, the MAE for every day of refitting and the methane feed to furnace for comparison. From the first plot, the GPRnCV seems the best choice. This is confirmed by the second plot since the GPRnCV is always lower than GPRCV and GPRE. $\Delta t_r = 1day$, $\Delta t = 8h$, $h = 10days$.

For what concerns the train and test errors, the models seem to over-fit on the train set more than the linear regression model. This is caused by the higher flexibility of the GPR that is able to model also the wiggles in the data.

In conclusion, the best approach for the learning phase is with a simple train-test technique

with a 80-20 split. The approaches with CV and ensemble method seems to give worst results. The former tends to always choose a linear model, thus keeping the error low but actually unable to extrapolate on future data. The second presents high fluctuations in the test error. The problem with CV and ensemble approaches is caused by a lack of data in the training set. Indeed, the models start to converge for the last iterations that are the one in which more data are used for the training of the model.

Under this perspective, the final results will be extrapolated with the GPRnCV and PRnCV methods only.

6.2. Application of best learning approach

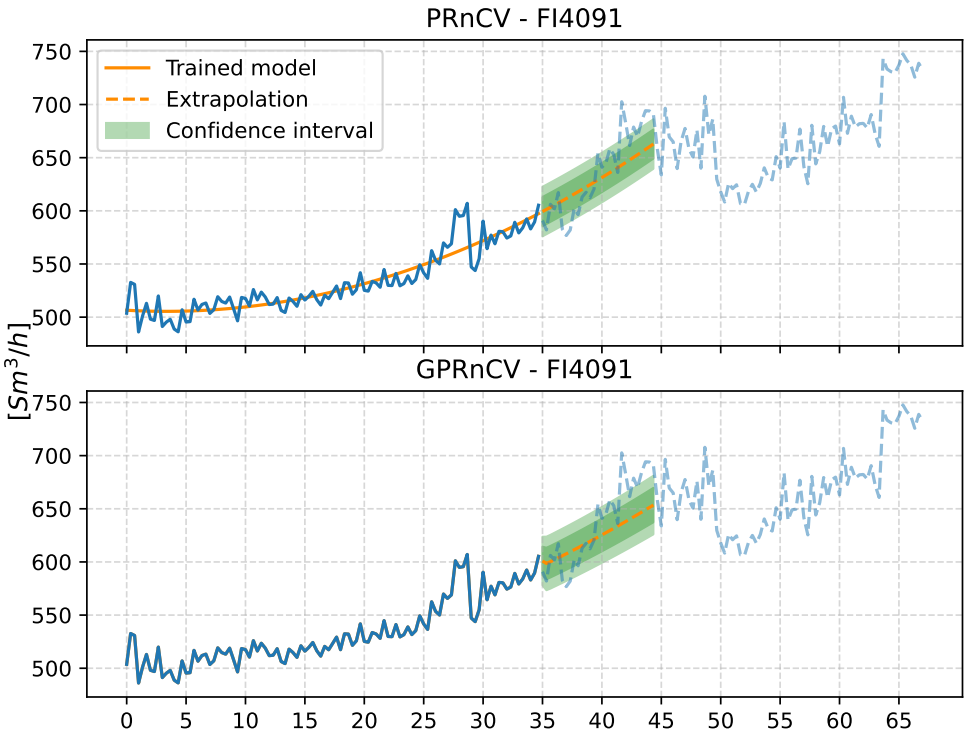
In the previous section, the time series of the methane feed to furnace *FI-4091* was used to choose the best approach for the learning phase of the model. After simulating a daily refitting on the second cycle, the GPRnCV and PRnCV were chosen as best approaches. In this section, the two approaches are explored more in details for the methane feed to furnace *FI-4091* and are generalized to the time series of the pressure drop *PI-4301* and tube skin temperature of the middle section *SK-4067*. The results were obtained for a refitting period $\Delta t_r = 1day$, time step between data points $\Delta t = 8h$ and prediction range of $h = 10days$ for *FI-4091* and *SK-4067* and $h = 5days$ for *PI-4301*.

6.2.1. Methane feed: *FI-4091*

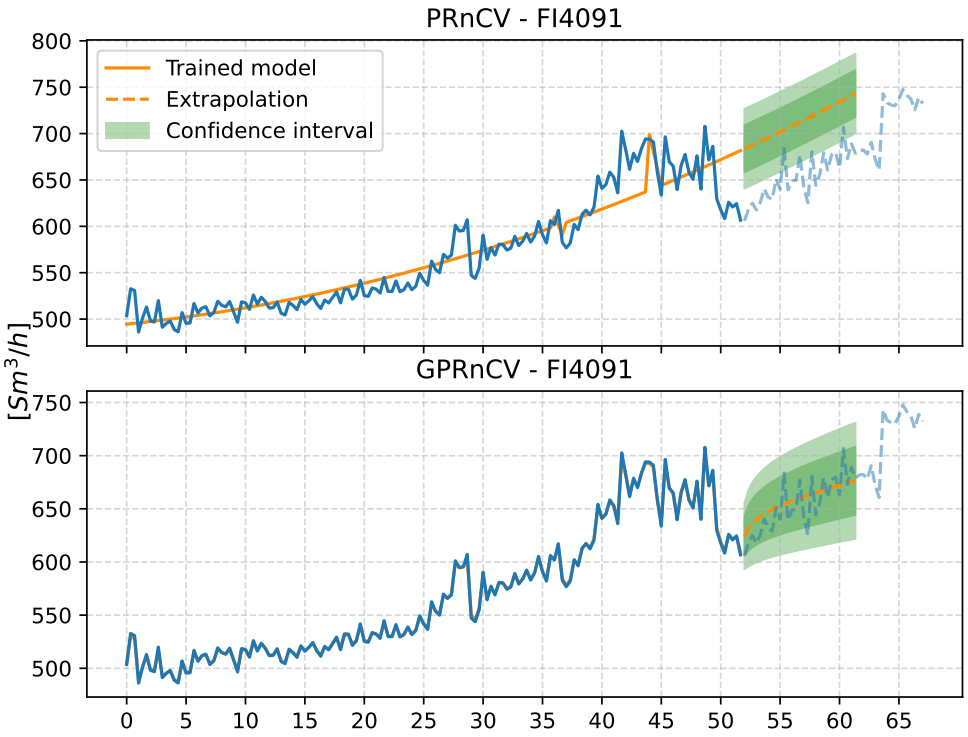
In Figure 6.3, the predictions obtained by the models GPRnCV and PRnCV on *FI-4091* are compared. Figure 6.3a, shows the prediction obtained 35 days after the startup, while Figure 6.3b has been obtained 52 days after the startup. Until day 42, the time series follows a quadratic trend that is correctly identified by both the polynomial and the GPR, as it is possible to see in Figure 6.3a. Indeed, in this period the best kernel selected form the decision tree is always composed by a combination of LIN and RBF kernels. As seen in Section 3.3, the multiplication of n LIN kernels results in a structure equivalent to that of a polynomial of degree n. Thus, the quadratic trend is learned by the LIN kernel, while the RBF models the small wiggles in the data

The power of the GPR can be grasped form the results obtained in Figure 6.3b. Being non-parametric, the GPR is able to better follow the structure of the data, while the polynomial adjust itself in order to reduce the error of the fit without truly learning the structure of the time series. The spikes in the trained model of the PRnCV are due to the use of dummy variables as regressors, indicating the effects of blowing in that point.

Table 6.1 reports the metrics obtained by the two predictions for both the GPRnCV and PRnCV models. The R^2 of GPRnCV is always higher than the one of PRnCV since the former is able to overfit the data thus explaining the majority of variability.



(a) Prediction obtained 35 days after the startup of the furnace.



(b) Prediction obtained 52 days after the startup of the furnace.

Figure 6.3: Comparison of predictions by GPRnCV and PRnCV on methane feed to furnace *FI-4091*. Form Figure 6.3a, the GPR model chooses a combination of LIN and RBF kernels. Form Figure 6.3b, being non-parametric the GPR is able to fit the structure of the time series. The shaded areas indicate 75% and 95% confidence intervals. $\Delta t = 8h$, $h = 10days$.

Days after startup	Model	MAE test	MAE train	R^2 train
35	GPRnCV	20.476	0.022	96.53 %
	PRnCV	19.787	8.867	84.61%
52	GPRnCV	14.398	1.223	99.93%
	PRnCV	59.726	15.314	86.97%

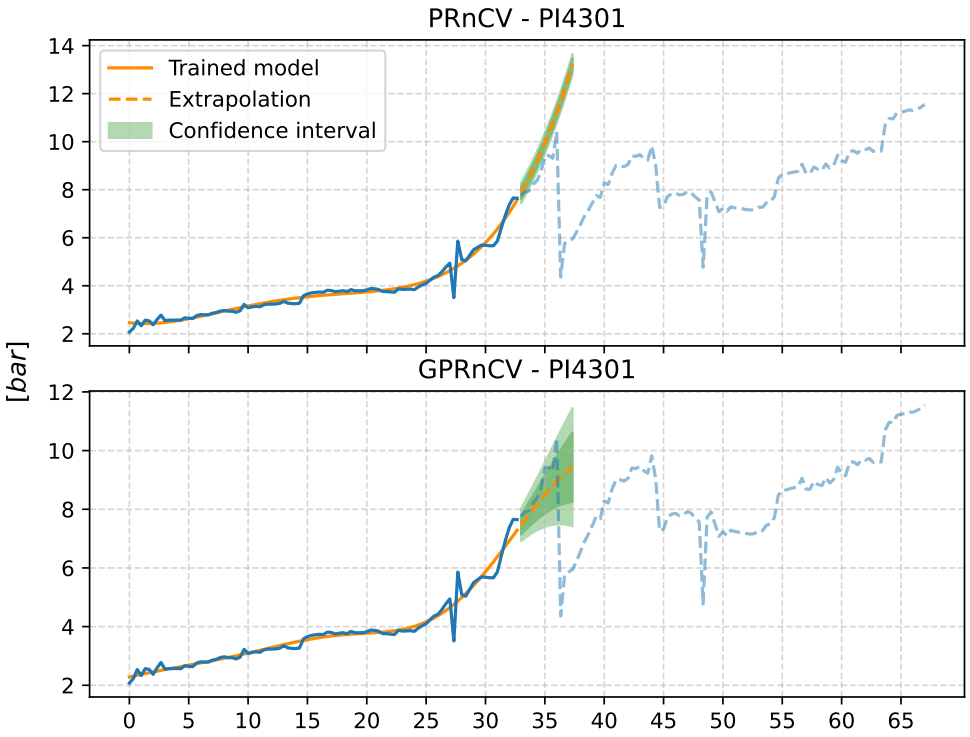
Table 6.1: Metrics from prediction on methane feed *FI-4091*. The value of R^2 shows that the GPRnCV is able to explain almost all the variability of the train set thanks to its flexibility but still being able to generalize well on the test set.

6.2.2. Pressure drop: *PI-4301*

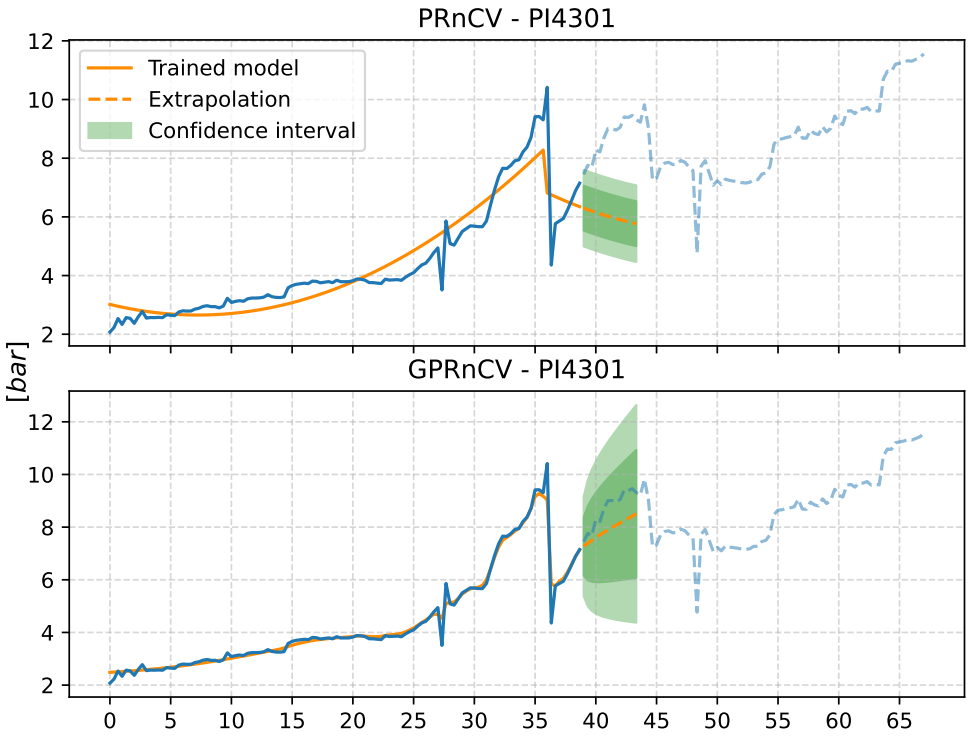
Since the pressure drop exhibits a structure with a shorter characteristic length scale as described in Section 5.2, in order to predict the peaks in pressure, a prediction range of $h = 5days$ was adopted.

In this case, the limitations of the parametric model resulting form PR are clearly visible in Figure 6.4, that compares the predictions obtained 33 and 39 days after the startup of the furnace. Day 39 was chosen in order to compare the models on their prediction abilities after the sudden change in pressure caused by blowing. As shown in Figure 6.4b, the PR is trying to minimize the sum of squares by tracking a line that goes across the data. On the other hand, the flexibility of the GPR is able to fit the the sudden decrease caused by blowing on day 36 and its subsequent increase. This is done thanks to the RQ and RBF kernels that are able to model the wiggles in the structure of the time series as discussed in Section 3.3. On the other hand, the first part of the time series is well modelled by both PRnCV and GPRnCV as shown in Figure 6.4a. The former returns a polynomial of fourth order that is reproduced by a combination of LIN and RBF kernels as in the case of *FI-4091*.

In Table 6.2 the metrics resulting form the predictions returned by the models for the two cases are reported. The results show that both the models are able to describe the majority of the variability on the train set for both the the cases. Nevertheless, by comparing the values of MAE on the test set, the GPRnCV shows a better performance on prediction.



(a) Prediction obtained 33 days after the startup of the furnace.



(b) Prediction obtained 39 days after the startup of the furnace.

Figure 6.4: Comparison of predictions by GPRnCV and PRnCV on pressure drops *PI-4301*. Form Figure 6.4a, the GPR model chooses a combination of linear kernels, thus actually performing a linear regression. Form Figure 6.4b, being non-parametric the GPR is able to fit the structure of the time series thus giving a better prediction. The shaded areas indicate 75% and 95% confidence intervals. $\Delta t = 8h$, $h = 4days$.

Days after startup	Model	MAE test	MAE train	R^2 train
33	GPRnCV	0.770	0.108	98.34 %
	PRnCV	2.677	0.490	87.18%
39	GPRnCV	1.445	0.128	96.53%
	PRnCV	2.473	0.144	96.52%

Table 6.2: Metrics from prediction on pressure drop *PI-4301*. Both the models are able to capture the majority of the variability on the train set for both cases. However, from the MAE on test the GPRnCV is able to better predict the trend of the curve.

6.2.3. Tube skin temperature: *SK-4067*

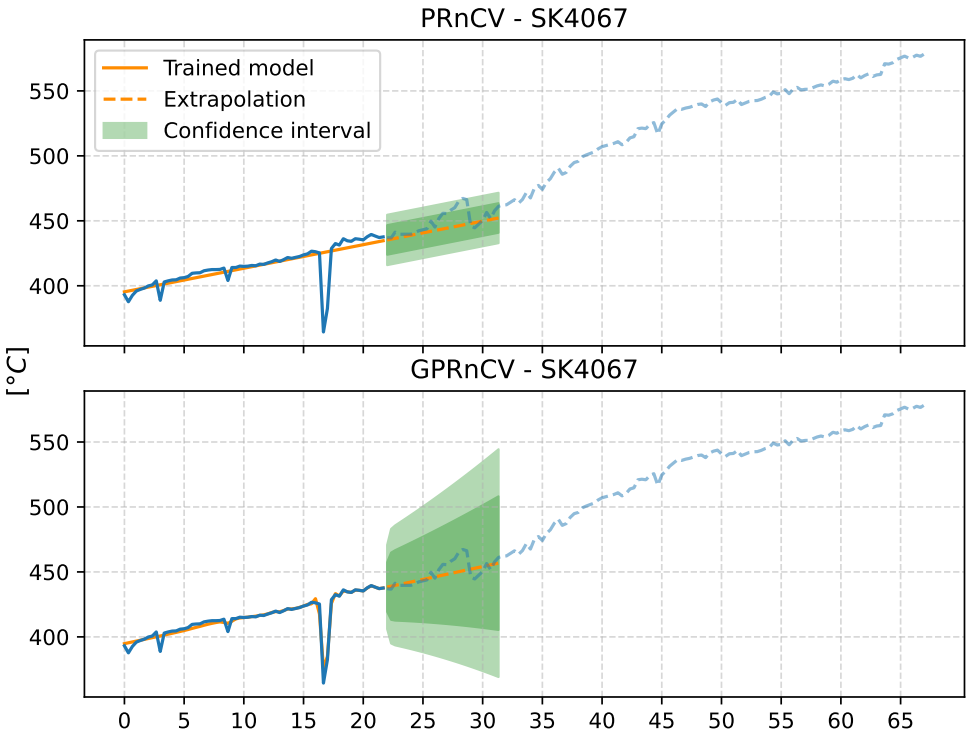
Figure 6.5, shows the comparison obtained obtained by means of GPRnCV and PRnCV models on *SK-4067* for a prediction range $h = 10days$. Figure 6.5a and 6.5b show the predictions obtained 22 and 53 days after the startup respectively.

Figure 6.5a shows the predictions obtained 22 days after the startup of the furnace. In this case, bot the GPRnCV and PRnCV are able to properly predict the trend of the time series; however, only the GPRnCV fits the unexpected decrease in temperature for which no assignable cause is available, thus resulting in a wider confidence interval. On the other hand, Figure 6.5b shows the ability f the GPRnCV in being more responsive to the smooth changes in the curve thanks to the RQ kernel as discussed in Section 3.3.

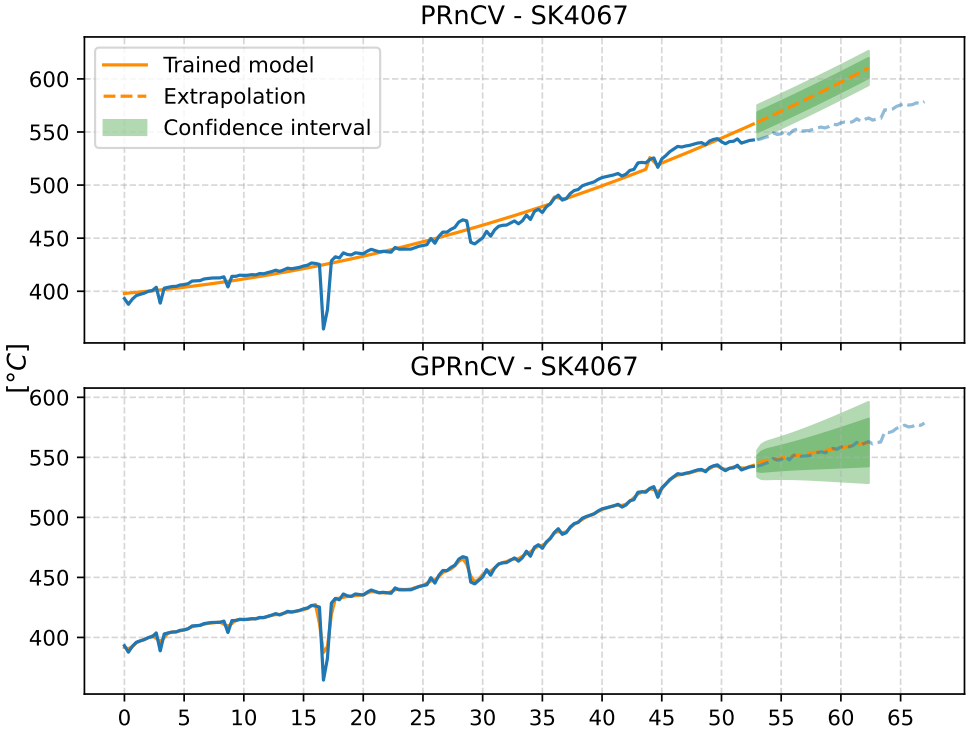
In Table 6.3, the metrics of the predictions obtained for the tube skin temperature are reported. The value of the R^2 for the PRnCv model is strongly affected by the sudden decrease in temperature on day 17 due to an unknown cause. Also in this case by looking at the values of MAE on the test set, it is possible to conclude that the prediction ability of the GPRnCV is higher than the one of the PRnCV model.

Days after startup	Model	MAE test	MAE train	R^2 train
22	GPRnCV	4.975	1.692	96.35%
	PRnCV	6.289	4.252	57.58%
53	GPRnCV	1.210	1.309	99.65%
	PRnCV	5.329	30.875	97.07%

Table 6.3: Metrics from prediction on tube skin temperature *SK-4067*. The value of the R^2 for the PRnCv model is strongly affected by the sudden decrease in temperature on day 17 due to an unknown cause. The values of MAE on the test set show that the prediction ability of the GPRnCV is higher than the one of the PRnCV model.



(a) Prediction obtained 22 days after the startup of the furnace.



(b) Prediction obtained 53 days after the startup of the furnace.

Figure 6.5: Comparison of predictions by GPRnCV and PRnCV on pressure drops *PI-4301*. Form Figure 6.5a, the GPR model chooses a combination of linear kernels, thus actually performing a linear regression. Form Figure 6.5b, being non-parametric the GPR is able to fit the structure of the time series thus giving a better prediction. The shaded areas indicate 75% and 95% confidence intervals. $\Delta t = 8h$, $h = 10days$.

6.3. Limit of the model

In the previous section, it was seen that the model is able to adequately extrapolate the trend of the time series in a prediction range $h = 10days$ for the methane feed *FI-4091* and tube skin temperature *SK-4067*, while for the pressure drop it can adequately extrapolate on a prediction range of $h = 5days$. Nevertheless, the model shows its limitations when the prediction range becomes too long.

In order to test the limits of the model, an experiment involving a long term prediction was carried out. For this purpose, both GPRnCV and PRnCV model are trained daily on the available data, while the prediction is carried out up to the day at which the furnace is turned off. Therefore day by day, the available data increase while the forecasting horizon decreases therefore, it is expected that the more data arrive, the higher the convergence of the model. The performance of the long term predictions are once again compared by means of the MAE.

Figure 6.6 shows the MAE obtained daily from the long term prediction on the variable of interest: *FI-4091, PI-4301* and *SK-4067*. On the x-axis, the days after the startup of the plant are reported. The target indicates the value of the MAE to which the models converge, while the shaded region indicates the period for which the models do not converge to the target.

From the first plot, it is possible to see that both the GPRnCV and PRnCV model stabilize around a value of $MAE = 30Sm^3/h$ after 52 days following the startup of the furnace, that is 15 days before the shutdown. Therefore, both the models are able to predict the finale state of the methane feed 15 days in advance with a mean average error of about $30bar$. The high instability of the models before day 52 is given by the strong irregularities in the time series tat does not seem to follow a regular path.

For what concerns the pressure drop *PI-4301*, both GPRnCV and PRnCV converge to a low value of the MAE that is around $MAE = 1bar$. GPRnCV is able to return a good prediction 38 days in advance while PRnCV 45 days in advance. Also, the linear model seems to return a slightly better prediction.

At last, the tube skin temperature before shutdown is adequately predicted 17 days before by the GPRnCV and 10 days before by the PRnCV with a value of the MAE lower than $10°C$.

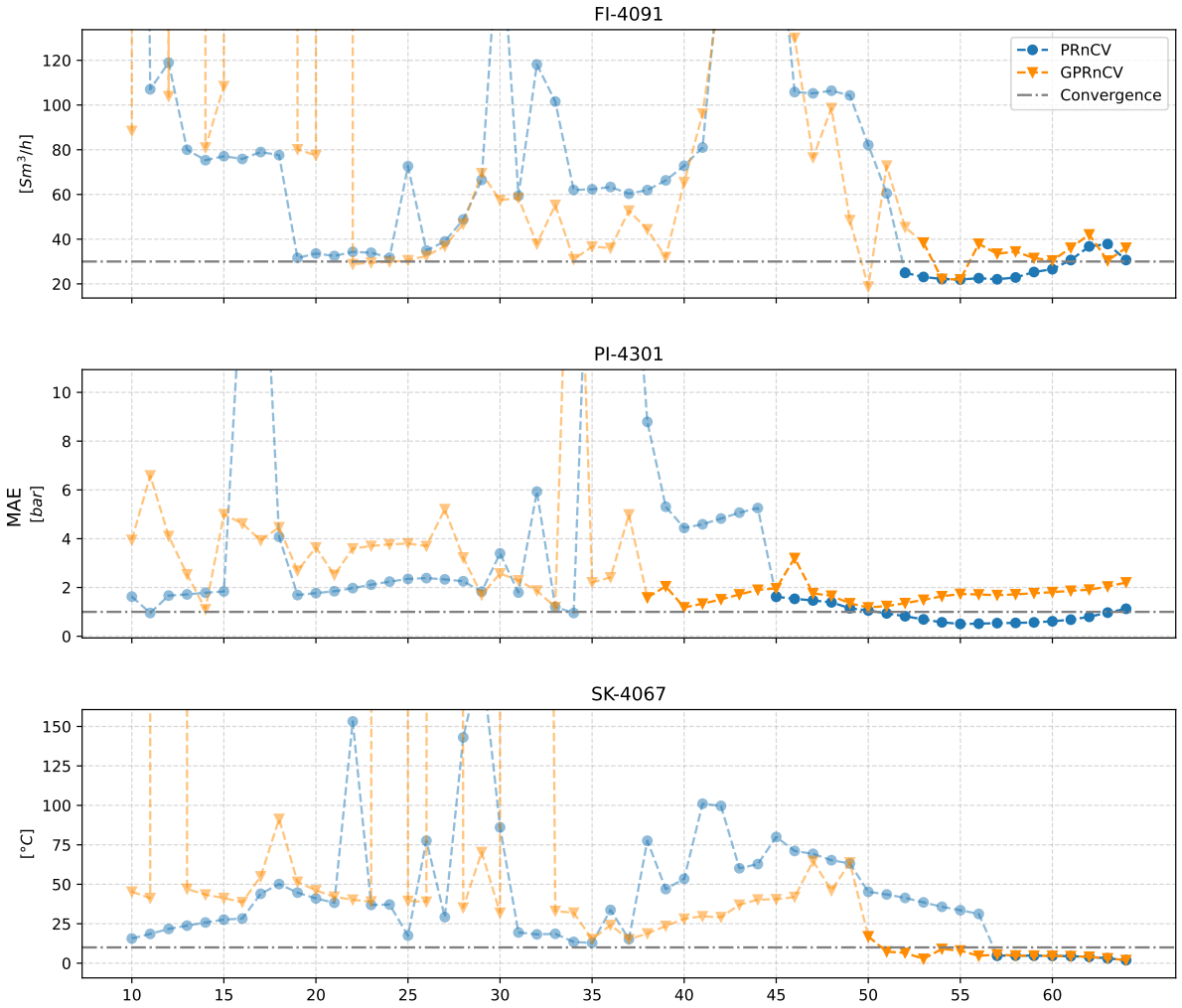


Figure 6.6: MAE resulting from daily long term predictions. The models are trained on the available data every day and extrapolate up to day 67 at which the furnace is shut down. On the x-axis, the days after the startup of the plant are reported. The target indicates the value of the MAE to which the models converge, while the shaded region indicates the period for which the models do not converge to the target. GPRnCV converges faster than PRnCV for *PI-4301* and *SK-4067*. The value of the MAE oscillates and the convergence is not close to zero due to the fluctuations present in the time series of interest.

6.4. Final result

In this chapter, the evaluation of Gaussian Process Regression and Polynomial Regression was discussed along with the best learning approach to train the model: Cross Validation,

ensemble and simple train-test split. All the evaluations were carried out by comparing the value of the MAE on the prediction.

The first part of the experiment found out that the best learning approach is the simple train-test split. This is because the CV and ensemble methods use less points to train the models given the way in which the data are split. Therefore, the Gaussian Process Regression and Polynomial Regression trained with no CV nor ensemble method were chosen as candidate models. The performance of the two models were than studied on the three time series of interest simulating a daily refitting and a prediction range of ten days for the methane feed to furnace *FI-4091* and tube skin temperature *SK-4067* and 4 days for the pressure drop *PI-4091*. It was seen that the ability of the GPR in modelling the wiggles in the time series was very useful especially for the prediction of the pressure drop that exhibits sudden mean shifts that are not modelled by the PR. This is useful since the GPR does not necessarily need dummy variables to account for sudden shifts in the data. In the end, the performance of the model on predictions of more than one months were tested. This revealed the inability of the GPRnCV and PRnCV in forecasting on such ranges mainly because of the strong irregularities present in the time series.

In conclusion, the GPR is the best candidate for the prediction of such kind of time series data. Indeed thanks to its non-parametric nature, it is able to better learn the structure of time series data coming from such a chemical process; that are characterized by sudden mean shifts and peaks for which no assignable cause is available and therefore difficult to model with a parametric model such as the PR. Also, the GPR is perfectly able to simulate a PR by choosing complex covariance functions composed of LIN kernels only.

7 | Conclusion and future developments

In conclusion, this work provides a starting point to gain benefit from the massive amount of data generated over the years by the Itelyum Regeneration plant located in Pieve Fissiraga (LO). Thanks to the Exaquantum PIMS by Yokogawa, it is possible to access the historian of data collected from the DCS of the plant in order to carry out BDA analysis for the optimal control and management of the process units.

The study aimed at developing a data driven approach to predictive maintenance of the process furnace located in the thermal de-asphalting section of the plant. This was done thanks to time series modelling of methane feed, pressure drop and tube skin temperatures in order to extrapolate the future state of the variables. By comparing PR and GPR, it was found out that the latter is most suitable for modelling the time series coming from the process furnace since they show high irregularities and no well defined structure that is well captured by a non-parametric and flexible model as the GPR. Also, the approach used to train the GPR model, that is a greedy search tree, is able to generalize well on different time series structure.

For future studies, the ability of the model to forecast on longer horizons should be improved in order to obtain a monthly prediction already 10-20 days after the startup of the furnace. This could be done by better tuning the hyperparameters of the kernels or by training the model on a larger amount of data. That is, also consider the data of past life cycle of the furnace when training the model. Here, the challenge is to extrapolate only relevant information for the prediction of the trend among the noisy observations and sudden mean shifts. Furthermore, it would be useful to generalize the approach to other process units. Also of grate interest, would be to find correlations between variables located in different parts of the plant to understand how the changes in the process conditions affect the plant.

Bibliography

[1] Plant Information Management System (Exaquantum) | Yokogawa Electric Corporation. https://www.yokogawa.com/solutions/solutions/asset-operations-and-optimization/data-historian/plant-information-management-system/#Details.

[2] J. Aminian and S. Shahhosseini. Evaluation of ANN modeling for prediction of crude oil fouling behavior. *Applied Thermal Engineering*, 28(7):668–674, May 2008. ISSN 1359-4311. doi: 10.1016/j.applthermaleng.2007.06.022.

[3] M. Berreni and M. Wang. Modelling and dynamic optimization of thermal cracking of propane for ethylene manufacturing. *Computers & Chemical Engineering*, 35(12): 2876–2885, Dec. 2011. ISSN 0098-1354. doi: 10.1016/j.compchemeng.2011.05.010.

[4] M. Bogojeski, S. Sauer, F. Horn, and K.-R. Müller. Forecasting industrial aging processes with machine learning methods. *Computers & Chemical Engineering*, 144: 107123, Jan. 2021. ISSN 0098-1354. doi: 10.1016/j.compchemeng.2020.107123.

[5] F. Brahim, W. Augustin, and M. Bohnet. Numerical simulation of the fouling process. *International Journal of Thermal Sciences*, 42(3):323–334, Mar. 2003. ISSN 1290-0729. doi: 10.1016/S1290-0729(02)00021-2.

[6] U. Edosio. Big data paradigm-analysis, application, and challenges. In *13th Research Seminar Series Workshop*, volume 6, page 7, 2014.

[7] R. Frigola. *Bayesian Time Series Learning with Gaussian Processes*. Thesis, University of Cambridge, Aug. 2015.

[8] Z. Ge, T. Chen, and Z. Song. Quality prediction for polypropylene production process based on CLGPR model. *Control Engineering Practice*, 19(5):423–432, May 2011. ISSN 0967-0661. doi: 10.1016/j.conengprac.2011.01.002.

[9] F. D. Giovanna. Lubricants Recycling – A Case Study: How Italy Managed to Become an Excellence and an Example for the Other EU's Member States. In B. Bilitewski, R. M. Darbra, and D. Barceló, editors, *Global Risk-Based Management of Chemical Additives I: Production, Usage and Environmental Occurrence*,

The Handbook of Environmental Chemistry, pages 225–251. Springer, Berlin, Heidelberg, 2012. ISBN 978-3-642-24876-4. doi: 10.1007/698_2011_100.

[10] R. J. Hyndman and G. Athanasopoulos. *Forecasting: Principles and Practice (2nd Ed)*. Otexts.

[11] M. S. Lavasani, N. R. Ardali, R. Sotudeh-Gharebagh, R. Zarghami, J. Abonyi, and N. Mostoufi. Big data analytics opportunities for applications in process engineering. *Reviews in Chemical Engineering*, Dec. 2021. ISSN 2191-0235. doi: 10.1515/revce-2020-0054.

[12] L. Li, P. N. Plessow, M. Rieger, S. Sauer, R. S. Sánchez-Carrera, A. Schaefer, and F. Abild-Pedersen. Modeling the Migration of Platinum Nanoparticles on Surfaces Using a Kinetic Monte Carlo Approach. *The Journal of Physical Chemistry C*, 121 (8):4261–4269, Mar. 2017. ISSN 1932-7447. doi: 10.1021/acs.jpcc.6b11549.

[13] D. C. Montgomery and G. C. Runger. *Applied Statistics and Probability for Engineers*. John Wiley & Sons, Mar. 2010. ISBN 978-0-470-05304-1.

[14] T. Nguyen, R. G. Gosine, and P. Warrian. A systematic review of big data analytics for oil and gas industry 4.0. *IEEE access : practical innovations, open solutions*, 8: 61183–61201, 2020. doi: 10.1109/ACCESS.2020.2979678.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[16] V. R. Radhakrishnan, M. Ramasamy, H. Zabiri, V. Do Thanh, N. M. Tahir, H. Mukhtar, M. R. Hamdi, and N. Ramli. Heat exchanger fouling model and preventive maintenance scheduling tool. *Applied Thermal Engineering*, 27(17):2791–2802, Dec. 2007. ISSN 1359-4311. doi: 10.1016/j.applthermaleng.2007.02.009.

[17] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, Mass., 3. print edition, 2008. ISBN 978-0-262-18253-9.

[18] K. Schwab. The Fourth Industrial Revolution: What it means, how to respond. page 7, Jan. 2016.

[19] B. A. Swastanto. Gaussian Process Regression for Long-Term Time Series Forecasting. 2016.

[20] I. A. Udugama, C. Bayer, S. Baroutian, K. V. Gernaey, W. Yu, and B. R. Young. Digitalisation in chemical engineering: Industrial needs, academic best practice, and curriculum limitations. *Education for Chemical Engineers*, 39:94–107, Apr. 2022. ISSN 1749-7728. doi: 10.1016/j.ece.2022.03.003.

[21] G. Van Rossum and F. L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009. ISBN 1-4414-1269-7.

[22] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri. A review of process fault detection and diagnosis: Part I: Quantitative model-based methods. *Computers & Chemical Engineering*, 27(3):293–311, Mar. 2003. ISSN 0098-1354. doi: 10.1016/S0098-1354(02)00160-6.

[23] K. Villalobos, V. J. Ramírez-Durán, B. Diez, J. M. Blanco, A. Goñi, and A. Illarramendi. A three level hierarchical architecture for an efficient storage of industry 4.0 data. *Computers in Industry*, 121:103257, Oct. 2020. ISSN 0166-3615. doi: 10.1016/j.compind.2020.103257.

[24] G. T. Wilson. Time Series Analysis: Forecasting and Control, 5th Edition, by George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel and Greta M. Ljung, 2015. Published by John Wiley and Sons Inc., Hoboken, New Jersey, pp. 712. ISBN: 978-1-118-67502-1. *Journal of Time Series Analysis*, 37(5):709–711, 2016. ISSN 1467-9892. doi: 10.1111/jtsa.12194.

[25] O. Wu, A. E. F. Bouaswaig, S. M. Schneider, F. M. Leira, L. Imsland, and M. Roth. Data-driven degradation model for batch processes: A case study on heat exchanger fouling. In A. Friedl, J. J. Klemeš, S. Radl, P. S. Varbanov, and T. Wallek, editors, *Computer Aided Chemical Engineering*, volume 43 of *28 European Symposium on Computer Aided Process Engineering*, pages 139–144. Elsevier, Jan. 2018. doi: 10.1016/B978-0-444-64235-6.50026-7.

[26] W. Yan, H. Qiu, and Y. Xue. Gaussian process for long-term time-series forecasting. In *2009 International Joint Conference on Neural Networks*, pages 3420–3427, June 2009. doi: 10.1109/IJCNN.2009.5178729.

[27] T. Zonta, C. A. da Costa, R. da Rosa Righi, M. J. de Lima, E. S. da Trindade, and G. P. Li. Predictive maintenance in the Industry 4.0: A systematic literature review. *Computers & Industrial Engineering*, 150:106889, Dec. 2020. ISSN 0360-8352. doi: 10.1016/j.cie.2020.106889.

List of Figures

List of Tables

List of Symbols

h	prediction interval
Δt_r	refitting period
Δt	time step between data points
PH-401B	process furnace
FI-4091	methane feed to furnace $[Sm^3/h]$
PI-4301	pressure drop inside the tubes of the furnace $[bar]$
SK-4067	tube skin temperature in the middle section of the furnace $[°C]$
MAE	Mean Average Error
LIN	Linear kernel
RBF	Radial Basis Function kernel
RQ	Rational Quadratic kernel
PER	Perodic kernel
GP	Gaussian Process
GPR	Gaussian Process Regression
PR	Polynomila Regression
CV	Cross Validation
GPRnCV	Gaussian Process Regression without Cross Validation
GPRCV	Gaussian Process Regression with Cross Validation
GPRE	Gaussian Process Regression with Ensemble method
DCS	Distribute Control System
PIMS	Plant Information Management System
ML	Machine Learning
BD	Big Data